



Universidad Carlos III de Madrid

Escuela Politécnica Superior

Departamento de Informática

**Adaptaciones de Unidades de Aprendizaje
en Tiempo de Ejecución**

TESIS DOCTORAL

Telmo Agustín Zarraonandia Ayo

Licenciado en Informática

Leganés, Madrid, 2007



Universidad Carlos III de Madrid

Escuela Politécnica Superior

Departamento de Informática

TESIS DOCTORAL

**Adaptaciones de Unidades de Aprendizaje
en Tiempo de Ejecución**

Autor: Telmo Agustín Zarraonandia Ayo

Directores: Dra. Camino Fernández LLamas
Dr. Juan Manuel Dodero Beardo

Leganés, Madrid, 2007

Tribunal nombrado por el Mgfco. y Excmo. Sr. Rector de la Universidad Carlos III de Madrid el día de de 2007:

Presidente:

Vocales:

Secretario:

Realizado el acto de la defensa y lectura de la Tesis el día de de 2007 en

Calificación:

EL PRESIDENTE

LOS VOCALES

EL SECRETARIO

A mis aitas

Resumen

La presente tesis doctoral se enmarca dentro del conjunto de trabajos relacionados con procesos de aprendizaje asistido por ordenador en los cuales es necesaria la figura de un tutor o instructor que supervise el proceso. Si bien, a día de hoy, en algunos tipos de procesos esta figura puede ser reemplazada por algún tipo de agente inteligente o algún otro mecanismo, en otros casos sigue siendo aún necesario y conveniente que el control y supervisión del proceso esté en manos de un tutor real.

El objetivo general de esta tesis es incrementar las posibilidades de adaptación de los diseños de procesos de enseñanza supervisados por instructor especificados mediante lenguajes de modelado educativo (EML). Dicho objetivo surgió al tratar de implementar procesos de estas características y detectar las limitaciones en los sistemas existentes. Con tal objeto, se planteó el desarrollo de una extensión al modelo de información de un EML que permitiese a los instructores de procesos de aprendizaje de las características antes mencionadas, la aplicación de adaptaciones análogas a las que llevarían a cabo en un entorno de enseñanza tradicional.

Con este propósito, y utilizando un proceso iterativo de desarrollo y validación, se construyó, partiendo de la especificación de un EML ya existente, **un modelo de adaptación** que permitiese su extensión y posibilitase la descripción de modificaciones tanto sobre la definición de los diseños de procesos desarrollados empleando dicha especificación, como sobre su comportamiento en ejecución. Dado que para detectar la necesidad de adaptar y conocer el tipo y características de la adaptación requerida, es necesario que el instructor pueda evaluar y monitorizar el desarrollo del proceso, así como conocer el grado de éxito alcanzado por las adaptaciones introducidas, el modelo de adaptación fue ampliado para incluir los elementos necesarios para proporcionar soporte al desarrollo de estas funciones. Con el fin de organizar el desarrollo de todas estas tareas se definió **un método de diseño iterativo de procesos de aprendizaje** que permite capturar dentro del propio diseño la experiencia obtenida de sus ejecuciones previas. Finalmente, se diseñó **un mecanismo para la implantación del modelo** que permite mantener la lógica de las adaptaciones separada de la información original

de la unidad de aprendizaje de tal forma que puedan definirse y aplicarse en tiempo de ejecución adaptaciones no previstas con anterioridad al comienzo de la misma.

La factibilidad técnica de la solución ha sido probada mediante el desarrollo de un caso de estudio consistente en la aplicación del modelo y mecanismo de adaptación a un caso concreto de EML, el IMS Learning Design. Dada su flexibilidad para permitir implementar distintos enfoques pedagógicos y a su amplia aceptación, esta especificación ha sido la escogida como guía para el desarrollo y validación de la solución. La adecuación del método propuesto al desarrollo de procesos de aprendizaje reales, y la capacidad expresiva del modelo para permitir la descripción de las distintas adaptaciones del diseño del proceso, han sido constatadas en tres escenarios de aprendizaje distintos, a lo largo de cuyo desarrollo fue posible traducir las adaptaciones que se introdujeron sobre la descripción del proceso implementado en la plataforma utilizada en un entorno real, a adaptaciones sobre una versión del proceso en forma de UoL.

Abstract

The frame of this work is the computer supported learning process supervised by human tutors. Although nowadays the instructor figure can be replaced by intelligent agents or other mechanisms in some occasions, other times it is still necessary and convenient to rely on the control and supervision of a real tutor.

This work aims at increasing the adaptation capabilities of learning process designs supervised by an instructor and specified by means of an educational modelling language (EML). The support provided to the instructor of the above mentioned characteristic process revealed to be insufficient and to overcome this limitations an extension of an existing EML information model was arranged. It should allow instructors to introduce adaptations analogous to the ones they perform on traditional learning environment.

For this purpose and taking as a base an existing EML specification, an **adaptation model** was produced following an iterative process of development and validation. The model should allow the instructor to describe the variations of the learning process design as well as to modify its runtime behaviour. On the other hand an instructor should be able to evaluate and monitor the process progress. This allows to perceive the necessity to adapt the process and to establish the characteristics of the required adaptation. Furthermore, she also should be able to measure the introduced adaptation success. For those reasons the adaptation model was extended with the necessary elements to provide support for these tasks. A **method for the iterative design of learning process** was produced with the aim to organize the performance of these functions and to capture the experience gained from prior executions in the process definition. Finally, a **mechanism for the model implementation** was designed. This mechanism maintains the adaptation information and the original unit of learning definition separated, and permits the definition and application of unforeseen adaptations during execution time.

The technical feasibility of the solution has been proved through a case study consisting on the application of the model and adaptation mechanism to the most outstanding EML proposal: the IMS Learning Design specification. As this is the most widely ac-

cepted EML, it also has been chosen as a guide for the development and validation of the proposed solution. The suitability of the presented method for the development of real learning processes has been stated by its application on three different scenarios. All the adaptations applied on the real process implementation has been translated into adaptations of UoL versions of the processes. This way, the expressivity of the adaptation model has also been shown.

Índice general

1. Introducción	1
1.1. Antecedentes	2
1.2. Objetivos	3
1.3. Método de Trabajo	6
1.3.1. Planificación general	6
1.4. Estructura del Documento	8
2. Estado de la cuestión	9
2.1. Modelos de educación on-line y tipos de interacción	10
2.1.1. Tipos de procesos según la interacción alumno instructor	12
2.1.2. Necesidad e importancia de la figura del instructor	14
2.1.3. Tareas del instructor	16
2.2. Adaptaciones en procesos educativos supervisados por instructor	18
2.3. Características de las adaptaciones	19
2.4. Enfoques de diseño de sistemas computacionales de soporte al proceso enseñanza	22
2.4.1. Sistemas de Educación Adaptativos e Inteligentes en Web	22
2.4.2. Sistemas de gestión del aprendizaje (LMS)	25
2.5. Lenguajes de Modelado Educativo	32
2.5.1. PALO	34
2.5.2. IMS Learning Design	35
2.6. Tecnologías relacionadas	37
2.6.1. aLFanet	37
2.6.2. LOCO-Cite	40
2.6.3. Resumen del capítulo	40

3. Planteamiento del problema	43
3.1. Soporte insuficiente para el desarrollo de las funciones del tutor	44
3.2. Lógica de adaptación encapsulada dentro de la UoL	46
3.3. Falta de un método que organice la introducción y evaluación de las adaptaciones	48
3.4. Relación entre problemáticas y objetivos de la tesis	49
4. Adaptación, observación y evaluación de procesos de aprendizaje	51
4.1. Adaptaciones del proceso de aprendizaje	52
4.1.1. Definición de <i>Adaptation poke</i>	54
4.1.2. Tipos de adaptaciones	55
4.1.3. Clasificación de los <i>adaptation pokes</i>	55
4.1.4. Validación de la UoL adaptada	58
4.2. Recuperación de información sobre la ejecución del proceso de aprendizaje	61
4.2.1. Progress Watcher	61
4.3. Evaluación del éxito del proceso de aprendizaje	62
4.4. Modelo conceptual	64
4.4.1. Modelo conceptual estático	64
4.4.2. Modelo conceptual dinámico	65
4.4.3. Elementos del modelo conceptual	66
4.5. Modelo de información	70
4.5.1. Elementos relacionados con los procesos de evaluación	70
4.5.2. Elementos relacionados con los procesos de adaptación	79
4.5.3. Elementos relacionados con los procesos de observación	83
4.5.4. Elementos comunes a todos los procesos	85
4.6. Método para el diseño iterativo de procesos de aprendizaje	95
4.6.1. Fases del método de diseño iterativo de procesos de aprendizaje	96
4.6.2. Implementación del método	98
4.7. Arquitectura de extensión de un motor de ejecución de EMLs	99
4.7.1. Implementación mediante Programación Orientada a Aspectos .	102
4.7.2. Validación de la adaptación mediante ontología	103
5. Evaluación	105
5.1. Desarrollo de un caso de estudio: IMS LD	106
5.1.1. Selección de características adaptables	106

5.1.2.	Selección de características observables	107
5.1.3.	Extensión de un motor de ejecución	110
5.1.4.	Validación del proceso adaptado	110
5.2.	Evaluación por casos de uso	113
5.2.1.	Estructura de los casos de uso	115
5.2.2.	Caso 1: Sustitución de recursos	115
5.2.3.	Caso 2: Recuperación de información sobre el curso	118
5.2.4.	Caso 3: Introducción de nuevos elementos	120
5.2.5.	Caso 4: Sincronización entre instancias de distintos participantes de un mismo proceso	123
5.2.6.	Caso 5: Repetición del proceso o partes del proceso	127
5.2.7.	Caso 6: Programación de las actividades de un curso	130
5.2.8.	Caso 7: Variación de características de elementos del curso	133
5.2.9.	Caso 8: Creación de versiones	136
5.3.	Aplicación a escenarios de aprendizaje reales	137
5.3.1.	ESA6: Curso de formación continua en el empleo	138
5.3.2.	Máster: Curso de educación superior de postgrado	151
5.3.3.	EAO: Curso de educación superior de grado	171
5.3.4.	Resumen de la aplicación a escenarios de aprendizaje reales	183
5.4.	Evaluación experta	189
5.5.	Resumen	193
6.	Conclusiones	197
6.1.	Conclusiones	197
6.1.1.	Conclusiones sobre el modelo	198
6.1.2.	Conclusiones sobre el método de diseño iterativo	199
6.1.3.	Conclusiones sobre el mecanismo de implementación	199
6.1.4.	Conclusiones adicionales	200
6.2.	Aportaciones	200
6.3.	Carencias y limitaciones	202
6.4.	Líneas de trabajo futuras	204
6.4.1.	Extensiones al trabajo realizado	204
6.4.2.	Perspectivas de trabajos relacionados	205

A. Traducción a XML Schema	207
A.1. XML y XML Schema	207
A.2. Descripción del modelo de información	208
A.2.1. Elementos relacionados con los procesos de evaluación	208
A.2.2. Elementos relacionados con los procesos de adaptación	224
A.2.3. Elementos relacionados con los procesos de observación	231
A.2.4. Elementos comunes a todos los procesos	234
B. Notación de tablas y diagramas	255
B.1. Formato de los diagramas	255
B.2. Formato de las tablas	256
B.3. Tipos de datos	257
C. Cuestionarios de evaluación	259
C.1.	259
D. Resultados de la evaluación	265
D.1.	265
Bibliografía	281

Índice de figuras

4.1. Adaptaciones de un proceso de aprendizaje	53
4.2. Modelo conceptual estático	64
4.3. Modelo conceptual dinámico	65
4.4. Diagrama del elemento Evaluation-Profile	70
4.5. Diagrama del elemento Proc-Component	71
4.6. Diagrama elemento Plain-Component	73
4.7. Diagrama del elemento Learning-Objective	76
4.8. Diagrama del elemento Evaluation	77
4.9. Diagrama del elemento Activation-Condition	78
4.10. Diagrama elemento Poke	80
4.11. Diagrama del elemento Peek	83
4.12. Diagrama del elemento Peek-Ref	85
4.13. Diagrama del elemento Expression	86
4.14. Diagrama del elemento Condition	87
4.15. Diagrama del elemento MathExpression	88
4.16. Diagrama del elemento Term	88
4.17. Diagrama del elemento relExpression	90
4.18. Diagrama del elemento LogicExpression	91
4.19. Diagrama del elemento BasicExpression	92
4.20. Diagrama del elemento DateExpression	94
4.21. Fases del proceso de diseño iterativo de una unidad de aprendizaje	96
4.22. Estructura del motor de ejecución	100
4.23. Diagrama de secuencia de la aplicación de un <i>adaptation poke</i>	101
5.1. Nueva operación en el patrón visitor para implementar la validación del estado en ejecución	113
5.2. Imagen de la versión del proceso implementada en Moodle	140

5.3. Imagen de la replica del proceso implementada mediante UoL	140
5.4. Captura de pantalla de la descripción de varias actividades para asignaturas del “Master de Ingeniería de la Web”	151
5.5. Imagen de la versión del proceso implementada en Moodle	172
5.6. Análisis sobre las características del modelo propuesto	190
5.7. Análisis sobre los requisitos de la adaptaciones del diseño	191
5.8. Análisis sobre las características del método propuesto	191
5.9. Análisis sobre las características del mecanismo de implementación propuesto	192
5.10. Análisis sobre los requisitos de la adaptaciones del diseño por los participantes de un proceso educativo	193
A.1. Elemento Evaluations-Profile	208
A.2. Elemento Proc-Components	212
A.3. Elemento Plain-Components	214
A.4. Elemento Learning-Objectives	217
A.5. Elemento Learning-Objective	220
A.6. Elemento Evaluation	221
A.7. Elemento Activation-Condition	223
A.8. Elemento Poke	225
A.9. Elemento Actions	228
A.10.Elemento Peek	231
A.11.Elemento Peek-Command	233
A.12.Elemento Expression	235
A.13.Elemento Condition	239
A.14.Elemento MathExpression	240
A.15.Elemento TermExpression	243
A.16.Elemento RelExpression	245
A.17.Elemento LogicExpression	246
A.18.Elemento BasicExpression	248
A.19.Elemento Peek-Ref	252
A.20.Elemento DateExpression	252

Índice de tablas

2.1. Tecnologías aplicables según el rol del tutor en el proceso de enseñanza	14
3.1. Relación entre problemáticas y objetivos de la tesis	50
4.1. Características de los pokes de retoque	56
4.2. Características de los pokes de reajuste	56
4.3. Características de los pokes correctivos	57
4.4. Características de los pokes de variación	57
4.5. Tabla de datos del elemento Evaluations-Profile	71
4.6. Tabla de datos del elemento Proc-Component	72
4.7. Tabla de datos del elemento Plain-Component	73
4.8. Tabla de datos del elemento Internal-Component	74
4.9. Tabla de datos del elemento External-Component	75
4.10. Tabla de datos del elemento Tutor-Component	75
4.11. Tabla de datos del elemento Learning-Objective	77
4.12. Tabla de datos del elemento Evaluation	78
4.13. Tabla de datos del elemento Activation-Condition	79
4.14. Tabla de datos del elemento Poke	83
4.15. Tabla de datos del elemento Peek	84
4.16. Tabla de datos del elemento Peek-Ref	85
4.17. Tabla de datos del elemento Expression	86
4.18. Tabla de datos del elemento Condition	87
4.19. Tabla de datos del elemento MathExpression	88
4.20. Tabla de datos del elemento Term	89
4.21. Tabla de datos del elemento Factor	89
4.22. Tabla de datos del elemento relExpression	90
4.23. Tabla de datos del elemento LogicExpression	92
4.24. Tabla de datos del elemento basicExpression	94

4.25. Tabla de datos del elemento DateExpression	95
5.1. Adaptaciones estáticas en el IMS LD	108
5.2. Adaptaciones dinámicas en el IMS LD	109
5.3. Selección de características dinámicas observables	111
5.4. Casos de uso: momento de introducción de las adaptaciones y observaciones	114
5.5. Casos de uso: características capturadas de las adaptaciones	116
5.6. Relación de las distintas adaptaciones introducidas en los escenarios de la evaluación empírica	184
5.7. Relación de las distintas observaciones introducidas en los escenarios de la evaluación empírica	185
5.8. Validación de la expresividad del modelo de adaptación	185
5.9. Relación entre objetivos y acciones de la evaluación empírica	186
5.10. Relación entre objetivos y pruebas de evaluación	196
D.1. Intereses de los evaluadores	266
D.2. Categorías de los evaluadores	266
D.3. Respuestas a la pregunta 1 del cuestionario	267
D.4. Respuestas a la pregunta 2 del cuestionario	268
D.5. Respuestas a la pregunta 3 del cuestionario	269
D.6. Respuestas a la pregunta 4 del cuestionario	270
D.7. Respuestas a la pregunta 5 del cuestionario	271
D.8. Respuestas a la pregunta 6 del cuestionario	272
D.9. Respuestas a la pregunta 7 del cuestionario	273
D.10. Respuestas a la pregunta 8 del cuestionario	274
D.11. Respuestas a la pregunta 9 del cuestionario	275
D.12. Respuestas a la pregunta 10 del cuestionario	276
D.13. Respuestas a la pregunta 11 del cuestionario	277
D.14. Respuestas a la pregunta 12 del cuestionario	278
D.15. Respuestas a la pregunta 13 del cuestionario	279
D.16. Respuestas a la pregunta 14 del cuestionario	280

Capítulo 1

Introducción

Al observar la aplicación de un determinado diseño instruccional veremos que su uso en la práctica suele conllevar cierto grado de flexibilidad. El tutor utiliza el diseño como punto de partida, observando tanto las características y el estado de los componentes del entorno actual de ejecución, como la evolución de los distintos participantes a lo largo del proceso, y determinando si es apropiado ejecutar el siguiente paso en la planificación original o es necesario introducir alguna variación en la misma. El control de la ejecución del proceso educativo está en gran medida en manos del instructor, quien modifica los eventos que suceden en el entorno del proceso con objeto de favorecer el aprendizaje.

Fox [29] sugiere emplear los fundamentos del contextualismo funcional como base filosófica sobre la que construir una ciencia empírica de aprendizaje e instrucción. Los contextualistas funcionales buscan predecir e influenciar eventos usando conceptos y reglas fundamentados empíricamente. Para identificar las variables que afectan la probabilidad, incidencia o prevalencia de un evento, manipulan de manera sistemática dichos eventos dentro de su contexto, observando los efectos resultantes [10, 32]. El análisis debe estar siempre dirigido por un objetivo de aprendizaje explícitamente definido, ya que el éxito sólo se puede medir en relación con la consecución de un objetivo [23].

Por otro lado, los diseñadores instruccionales analizan el aprendizaje y diseñan, desarrollan, implementan, evalúan y gestionan situaciones, con objeto de mejorar el aprendizaje de los temas de estudio en diversas configuraciones [75]. La investigación instruccional conducida desde la perspectiva del contextualismo funcional conlleva por tanto un énfasis especial tanto en la especificación de los objetivos que los educadores tratan de conseguir mediante los entornos de aprendizaje que construyen, como en

la especificación de su satisfacción, así como en el estudio de las relaciones entre los resultados y las acciones llevadas a cabo en el entorno.

Desafortunadamente los entornos de aprendizaje existentes en la actualidad no ofrecen los mecanismos necesarios para poder desarrollar experiencias educativas desde este enfoque. Si bien permiten a los diseñadores del proceso capturar diversas condiciones y escenarios de ejecución y establecer diferentes respuestas para los mismos, dichas condiciones deben ser definidas de manera previa al comienzo del proceso. Por otro lado, el instructor no dispone de los medios suficientes para poder observar el desarrollo del proceso, vigilar que se vayan alcanzando los objetivos marcados y, en caso necesario, modificar el entorno para garantizar su consecución.

El objetivo de este capítulo es ofrecer una síntesis del propósito de la tesis, planteando el problema que dió origen a su desarrollo, definiendo los objetivos que se persiguen y describiendo el método de trabajo empleado para resolverlo.

1.1. Antecedentes

Desde los orígenes en los años sesenta de la enseñanza asistida por ordenador, muchas han sido las soluciones propuestas que han tratado de emplear los progresivos avances informáticos con el fin de mejorar la efectividad de los procesos de enseñanza/aprendizaje. De los limitados primeros sistemas CAIS (*Computer Assisted Instruction Systems*) de la época y del uso de grandes *mainframes* se pasó a almacenar grandes cantidades de contenido educativo en un sólo CD-ROM que podía ser ejecutado en PCs domésticos. De la escasa interactividad proporcionada por los primeros sistemas se ha pasado a Sistemas Tutoriales Inteligentes (ITS, *Intelligent Tutoring Systems*) [87] que combinan el uso de técnicas de Inteligencia Artificial (IA) con modelos psicológicos del estudiante y que son capaces de guiar al alumno a través de la resolución de problemas teniendo en cuenta su particular evolución. La aparición de la Web y la combinación de técnicas provenientes de los ITS con los sistemas adaptativos hipermedia propició el desarrollo de complejos Sistemas de Educación Adaptativos e Inteligentes en Web (AIES, *Adaptive Intelligent Educational Systems*) [12] que trataban de dar un paso más y de adaptar el contenido y el proceso educativo a cada estudiante según sus necesidades pedagógicas. Por otro lado el aprendizaje basado en CD-ROMs fue desplazado por un aprendizaje basado en Web (*Web-based Training*) permitiendo disminuir los costes asociados a la distribución, actualización y mantenimiento de los contenidos

educativos.

El empleo de contenidos educativos multimedia e interactivos facilita la asimilación de conceptos, ayuda a retener la atención del alumno y por lo general, deviene en procesos educativos más rápidos y eficientes. Sin embargo, su alto coste de desarrollo es, a día de hoy, uno de los principales obstáculos a la hora de implementar procesos educativos soportados mediante tecnología informática. Con el objeto de garantizar la interoperabilidad de los contenidos desarrollados entre plataformas de diversos fabricantes y proteger las inversiones realizadas, se han propuesto en los últimos años una serie de especificaciones y estándares que tratan de establecer patrones comunes en el desarrollo de los componentes de los sistemas educativos, incluyendo aspectos como el empaquetado de los contenidos, su marcado con metadatos, la interacción en tiempo de ejecución, especificación de preguntas y tests, perfil de alumnos, etc. Dentro de estas especificaciones se incluyen los lenguajes de modelado educativo (EMLs), los cuales proporcionan una herramienta para describir diseños de procesos de aprendizaje definiendo tanto las actividades que se llevarán a cabo como el uso que se hará de los distintos recursos. Estos diseños se pueden reutilizar una y otra vez para repetir el mismo proceso educativo con distintos participantes.

En ocasiones, sin embargo, la descripción de diseños mediante EMLs resulta demasiado cerrada y no proporciona la flexibilidad necesaria para el desarrollo del proceso educativo en un contexto real de aplicación. A menudo es necesario modificar el diseño original durante su ejecución de tal manera que se pueda dar respuesta a diversas situaciones y escenarios no previstos inicialmente. Por otro lado, la forma de trabajar de algunos instructores no se ajusta tampoco a ese esquema, prefiriendo delinear inicialmente el proceso y completar su definición durante su propio desarrollo, utilizando diversos materiales o llevando a cabo diferentes acciones según la evolución e intereses de los participantes. Es necesario por tanto que el instructor disponga de cierto grado de libertad a la hora de aplicar procesos de aprendizaje descritos mediante un determinado lenguaje de modelado educativo, de tal forma que pueda ser capaz de modificar la definición del proceso durante su propia ejecución.

1.2. **Objetivos**

El objetivo principal de esta tesis es desarrollar un modelo de adaptación que permita a los instructores de procesos de aprendizaje asistidos por ordenador cuyo diseño

ha sido especificado empleando un determinado EML, la aplicación de adaptaciones análogas a las que llevarían a cabo en un entorno de enseñanza tradicional.

Con este propósito se han planteado una serie de objetivos principales y secundarios que se presentan a continuación. Así, se han propuesto como objetivos generales de la tesis:

- (O1) Desarrollo de un modelo que permita la descripción de adaptaciones en tiempo de ejecución sobre la definición de procesos de aprendizaje supervisados por instructor y especificados mediante un lenguaje de modelado educativo. Este modelo debe proporcionar los elementos necesarios para:
 - (O1.1) Describir adaptaciones capaces de modificar tanto la definición original del proceso como su comportamiento en ejecución.
 - (O1.2) Definir el momento preciso en que la adaptación debe ser aplicada al proceso. Dicho momento podrá ser descrito empleando tanto expresiones temporales como condiciones o referencias a eventos.
 - (O1.3) Soportar la monitorización del desarrollo del proceso.
 - (O1.4) Facilitar el análisis del éxito de las adaptaciones introducidas.
 - (O1.5) Facilitar el análisis del éxito del proceso.

- (O2) Desarrollo de un método que organice tanto la introducción de las adaptaciones en tiempo de ejecución, como el resto de las tareas del instructor soportadas mediante elementos del modelo. Este método debe permitir:
 - (O2.1) Ordenar el desarrollo de las tareas del instructor soportadas mediante elementos del modelo de adaptación.
 - (O2.2) Recoger de forma explícita los cambios introducidos sobre la definición original del proceso.
 - (O2.3) Realizar las actividades de adaptación y monitorización del proceso.
 - (O2.4) Realizar las actividades de evaluación de los cambios introducidos y éxito del proceso.

Por otra parte se ha planteado como objetivo secundario:

- (O3) Desarrollo de un mecanismo que permita extender un motor de ejecución de procesos de aprendizaje de tal forma que sea capaz de interpretar y aplicar los elementos del modelo de adaptación. Entre las características que este mecanismo deberá reunir destacan:
- (O3.1) No intrusión. La extensión de los entornos de ejecución debe poder llevarse a cabo de manera no intrusiva, manteniendo separados los componentes propios del entorno de ejecución y los de la extensión, facilitando así su mantenimiento.
 - (O3.2) Consistencia. Debe garantizar la validez del diseño del proceso una vez que la adaptación ha sido aplicada, es decir, debe impedir la definición de adaptaciones correctas sintácticamente pero inconsistentes con la especificación del EML.
 - (O3.3) Flexibilidad. Debe minimizarse el impacto que futuros cambios sobre la especificación de los elementos del lenguaje de modelado puedan tener sobre las implementaciones del mecanismo.

El modelo propuesto podrá ser empleado para extender la especificación de un EML existente e incrementar tanto su capacidad de adaptación en tiempo de ejecución como el soporte proporcionado al instructor del proceso. Esta extensión es el resultado de distintos trabajos desarrollados por el autor y que han sido recogidos en diversas publicaciones científicas. Así, por una parte, en [93] se describe un mecanismo que posibilita la modificación de la definición de procesos de aprendizaje en tiempo de ejecución así como una arquitectura para extender motores de ejecución de *IMS Learning Design* [41]. Por otra parte, en [94], se recoge un modelo de información que permite capturar y describir las características de la adaptación a introducir, así como definir distintas observaciones y monitorizaciones y facilitar la evaluación del éxito del proceso. Por último, en [95] se define un método que organiza la introducción de las adaptaciones en tiempo de ejecución.

La aportación fundamental de esta tesis será por tanto la definición de un modelo y un método que permita a los instructores introducir adaptaciones sobre un proceso de aprendizaje especificado mediante un EML. Como aportaciones significativas que surgen como resultado del proceso de investigación, caben destacar las siguientes:

- Desarrollo de una técnica que posibilite la definición y aplicación de adaptaciones

sobre el diseño de un proceso de aprendizaje y permita mantener separada la lógica de la adaptación de la información original de la UoL.

- Definición de un instrumento que posibilite la recuperación de información acerca del estado del proceso, sus componentes o las actividades desarrolladas por sus participantes.
- Elaboración de un mecanismo que permite la especificación tanto de los objetivos de aprendizaje definidos para el proceso, como de la forma de medir el grado de satisfacción de los mismos por los distintos participantes, así como el éxito de las adaptaciones introducidas.
- Definición de un método de diseño de procesos de aprendizaje que organice el desarrollo de las diferentes tareas del instructor del proceso soportadas mediante elementos del modelo.
- Descripción de un mecanismo de validación de diseños de procesos de aprendizaje basado en el uso de ontologías.

La tercera aportación constituye la realización práctica del principio propuesto por el contextualismo funcional, tal y como se describió en la introducción de este capítulo, al permitir al instructor del proceso establecer y evaluar los objetivos marcados para el proceso, así como facilitar el análisis de los efectos de la manipulación de sus componentes.

1.3. Método de Trabajo

En esta sección se describen brevemente las distintas fases llevadas a cabo para la realización de la tesis, así como el método de evaluación que se ha utilizado para validar sus resultados.

1.3.1. Planificación general

La planificación general del trabajo a realizar en la tesis ha sido definida según la metodología de investigación presentada en [79]:

Determinación del problema

1. Descripción del contexto del problema: Definición del contexto del problema y los motivos que dan origen al desarrollo de la tesis.
2. Estudio del estado de la cuestión: Se llevará a cabo un análisis de las diversas tareas y necesidades de un instructor de procesos de aprendizaje soportados computacionalmente, así como de las necesidades de adaptación y distintos tipos y características de las mismas. A continuación se analizarán los dos enfoques principales de desarrollo de sistemas de soporte a la enseñanza, evaluando su adecuación a los procesos en los cuales existe un rol de instructor. Asimismo, se describirán brevemente los principales estándares y especificaciones empleados en el desarrollo de material educativo y se analizarán las posibilidades que ofrece el uso de EMLs, haciendo especial hincapié en la especificación del *IMS Learning Design*.
3. Definición del problema: Tomando como punto de partida el estudio realizado se analizarán las principales carencias y problemáticas que pueden surgir a la hora de implementar mediante un EML un proceso educativo supervisado. Dichas problemáticas vienen principalmente originadas por la no disponibilidad de recursos suficientes para hacer posible la definición e introducción de adaptaciones en el diseño del proceso durante la propia ejecución del mismo. El problema abordado con el desarrollo de esta tesis se definirá de manera concreta.

Formulación de las hipótesis

1. Elaboración de los objetivos perseguidos con la realización de este trabajo, así como enumeración de las aportaciones que se prevee obtener.
2. Diseño de una solución verificable al problema. Esta solución estará formada por un modelo de adaptación que permita extender la especificación de un EML, un método que determine la manera de utilizar y aplicar dicho modelo, y un mecanismo que habilite su implementación.

Validación de las hipótesis

Para llevar a cabo la validación de las hipótesis se emplearán un conjunto de métodos de evaluación que combinan tanto pruebas observacionales como experimentales, con el fin de demostrar tanto la consecución de los objetivos planteados, como la utilidad,

factibilidad y calidad de la solución propuesta, y la generalidad y expresividad del modelo de adaptación.

Análisis de los resultados

Se expondrán las distintas conclusiones alcanzadas tanto durante la labor de investigación asociada al desarrollo de la tesis como durante el análisis de los resultados obtenidos en las diferentes actividades de evaluación y se detallarán diversas líneas de ampliación del trabajo.

1.4. Estructura del Documento

En esta sección se detalla la estructura del documento describiendo el objetivo fundamental de cada capítulo.

- Capítulo 1: Introducción al trabajo, resumen de objetivos y descripción de la metodología de trabajo.
- Capítulo 2: Estado de la cuestión en materia de entornos de aprendizaje adaptativos.
- Capítulo 3: Planteamiento del problema abordado y descripción detallada de los objetivos perseguidos y las aportaciones previstas.
- Capítulo 4: Descripción del modelo, método y mecanismo propuestos.
- Capítulo 5: Evaluación de la solución propuesta.
- Capítulo 6: Descripción de las conclusiones obtenidas, principales aportaciones del trabajo y líneas de investigación futuras.
- Apéndice A: Implementación del modelo de información propuesto mediante XML schema.
- Apéndice B: Descripción del formato de los diagramas y tablas del documento, así como de los distintos tipos de datos del modelo de información propuesto.
- Apéndice C: Cuestionarios utilizados para llevar a cabo la evaluación experta.
- Apéndice D: Resultados de la evaluación experta.

Capítulo 2

Estado de la cuestión

Las tecnologías de la información y comunicación se hallan a día de hoy presentes en prácticamente todos los ámbitos de la vida cotidiana. Su integración dentro de los procesos de enseñanza-aprendizaje es por tanto una necesidad: por una parte, es necesario formar al alumno para que adquiera competencias básicas en su uso, y por otra, es deseable aprovechar las ventajas derivadas de su empleo como soporte del proceso educativo. Entre algunas de dichas ventajas cabe destacar:

- Uso de material multimedia: facilita la retención y asimilación de los conceptos y aumenta la motivación del alumno al posibilitar la representación de la información de diferentes formas e involucrando diferentes sentidos al mismo tiempo.
- Individualización del proceso: el ordenador puede adaptar el material presentado a las características del alumno, estilo de aprendizaje, conocimientos previos, etc.
- Adaptación y control del proceso en manos del alumno: se puede proporcionar al alumno la posibilidad de controlar el desarrollo del proceso y tomar decisiones sobre el ritmo de aprendizaje, secuenciación del contenido, etc.
- Acceso a un gran volumen de información proporcionado por Internet.
- Liberar los procesos de aprendizaje de restricciones de espacio y de tiempo: las actuales tecnologías permiten la articulación de distintos procesos sociales a distancia. La rapidez alcanzada por los mecanismos de comunicación permiten implementar procesos de enseñanza-aprendizaje en los que los diversos participantes se encuentren ubicados en distintos puntos geográficos e interactúen entre sí de manera asíncrona.

Quizás, de entre todas las ventajas mencionadas, las más perseguidas hayan sido las de conseguir procesos de aprendizaje que se adapten a las características del alumno, que puedan ser repetidos un número indeterminado de veces para distintos participantes y que puedan ser ejecutados con independencia de la ubicación física del alumno. En cualquier caso, el uso de una nueva herramienta provoca cambios dentro del propio proceso, creando nuevas posibilidades y enriqueciéndolo. En palabras de Lev Vigotsky [85]: “Toda actividad depende del material con el que opera”. Así pues, si bien el uso de nuevas tecnologías puede proporcionar ventajas estratégicas importantes a las instituciones educativas de todos los niveles, su utilización efectiva implica un re-planteamiento o re-diseño a fondo, no sólo en los métodos de enseñanza y planes curriculares, sino también en las prácticas de trabajo y en los papeles desempeñados por profesores y alumnos.

En el resto de este capítulo se llevará a cabo un análisis de los dos principales enfoques posibles a la hora de dotar de soporte computacional a un proceso de aprendizaje supervisado por instructor, así como de algunas tecnologías relacionadas. Previamente se analizarán las distintas formas de interacción existentes en un proceso de enseñanza, así como el papel del tutor y la función que desempeña en dicho proceso. También se estudiará la importancia de la adaptación del diseño de este tipo de procesos, así como las características de las mismas.

2.1. Modelos de educación on-line y tipos de interacción

En [77] se describen cuatro modelos de educación on-line que sirven para ilustrar cuatro niveles distintos de integración de las TIC dentro del proceso educativo: el modelo naive, el estándar, el evolucionado y el radical.

- Modelo naive o de iniciación: la tecnología se utiliza como un mero complemento de la clase tradicional. El material utilizado en las clases presenciales es transformado mínimamente para poder ser puesto a disposición de los alumnos en la Web.
- Modelo estándar: trata de aprovechar ciertas ventajas ofrecidas por las TIC, de tal manera que al material propio del modelo de iniciación se añaden recursos electrónicos diseñados específicamente para su uso en la Web. También se emplean

herramientas que posibiliten la evaluación del conocimiento del alumno de forma *on-line*. En definitiva, parte de la actividad del alumno durante el proceso es desplazada al ámbito tecnológico.

- Modelo evolucionado: va un paso más allá del modelo estándar e incorpora herramientas que permiten a los instructores llevar a cabo un seguimiento más preciso de la actividad del alumno.
- Modelo radical: El instructor actúa como mera guía a petición del alumno, quien toma la iniciativa en el proceso y desarrolla un papel activo a lo largo del mismo. El aprendizaje se alcanza a partir de la interacción con otros alumnos y la utilización de gran cantidad de recursos Web existentes que el propio alumno transforma. La comunicación entre los participantes del proceso se lleva a cabo fundamentalmente a través de correo electrónico y otras tecnologías.

El papel de la tecnología por tanto, varía de mero añadido a la clase tradicional en el modelo de iniciación, a principal soporte del proceso de aprendizaje en los modelos evolucionado y radical, donde proporciona un entorno virtual sobre el que tienen lugar las distintas interacciones entre los participantes del proceso.

Por otra parte, el papel que los participantes del proceso desempeñan dentro del mismo también varía dependiendo del modelo implementado. Moore [70] discute tres tipos de interacción esenciales en la educación a distancia y que están presentes en los cuatro modelos de educación arriba descritos:

- Alumno-contenido: el alumno aumenta su conocimiento sobre una materia particular interactuando con ciertos contenidos didácticos.
- Alumno-instructor: el instructor trata de estimular el interés del alumno, aumentar su motivación diseñando un plan de estudio y supervisa el desarrollo del proceso.
- Alumno-alumno: intercambio de información e ideas acerca del contenido entre los participantes de un determinado curso o proceso.

Además de estos tres tipos de elementos Hillman, Willis y Gunawardena [34] añadieron un cuarto tipo de interacción alumno-interfaz, es decir, interacción entre el alumno y la tecnología empleada para soportar el proceso, que consideran crítica en la implementación un proceso de enseñanza-aprendizaje sobre un entorno virtual.

A la hora de preparar un proceso de enseñanza se debe decidir y determinar el nivel al que se desarrollarán cada uno de los tipos de interacción para conseguir que el aprendizaje sea efectivo. Cada tipo de interacción deberá ser analizado desde distintos puntos de vista. Así, para la interacción alumno-contenido se deberá tener en cuenta las características del propio alumno, la usabilidad del contenido a emplear y las características de la plataforma que dará soporte al proceso entre otras. Para la interacción alumno-instructor la frecuencia, intensidad y tipo de la influencia del segundo deberá ser determinada, así como los medios que se proveerán para el establecimiento de la comunicación entre ambos. Igualmente, el tipo de servicio empleado a utilizar en la comunicación deberá ser tenido en cuenta a la hora de analizar la interacción alumno-alumno, que además deberá ser abordada desde la perspectiva del aprendizaje colaborativo, estableciendo cómo y a qué nivel se va a producir la cooperación entre los participantes, cuál será la forma de gestión de los grupos, etc. Dado que este trabajo se encuentra enmarcado en los procesos de enseñanza supervisados por instructor, en el siguiente punto nos analizaremos más en detalle distintas posibilidades de interacción alumno-instructor.

2.1.1. Tipos de procesos según la interacción alumno instructor

Según el tipo de interacción existente entre alumno e instructor podemos clasificar los procesos de enseñanza-aprendizaje en tres grupos distintos, cada uno de los cuales demandará tecnología con distintas características:

- Procesos en los cuales el instructor desempeña un papel activo: Para garantizar el éxito del proceso contamos por un lado con una serie de materiales didácticos, por otro con un determinado diseño o planificación del proceso, y por otro con el conocimiento específico del dominio y capacidad docente del instructor. Éste será el encargado de presentar los materiales, dirigir las actividades a realizar así como llevar a cabo las intervenciones necesarias para adecuar el desarrollo del proceso a las actuales circunstancias de ejecución. Este tipo de procesos se corresponde con el modelo clásico de enseñanza presencial, donde la mayor parte del peso del proceso docente recae sobre los hombros del instructor.

En este contexto, la mayoría de las aportaciones tecnológicas pueden ser englobadas en dos grandes grupos: aquéllas dirigidas a dar soporte a la instrucción en un dominio determinado (simuladores, programas específicos, etc.) y aquéllas diri-

gidas a complementar y facilitar la actividad del docente (presentaciones, correo electrónico, servidor de ficheros para publicar material, etc.)

- Procesos en los cuales el instructor desempeña un papel de *moderador* o *supervisor*: En estos procesos el instructor se limita a guiar el proceso, garantizando que se desarrolla según el plan establecido y proveyendo, cuando es requerido, asistencia a los alumnos. La capacidad docente del tutor cobra menos relevancia de cara al éxito del proceso siendo más importante el diseño del proceso y de los contenidos educativos a utilizar. Como ejemplo de este tipo de procesos podemos destacar la enseñanza a distancia, un modelo que si bien existe desde hace ya muchas décadas, ha experimentado un auge espectacular debido a los avances en la tecnologías de la comunicación de los últimos años.

En cuanto a tecnología utilizada debemos añadir a los dos grupos citados en el punto anterior un tercero formado por aquellas aplicaciones cuyo objetivo no es proveer facilidades que complementen la actividad docente, sino proporcionar un entorno virtual que permita el completo desarrollo del mismo. Las funcionalidades ofrecidas por dichos entornos varían y pueden incluir tanto mecanismos que permitan la gestión y control de los distintos cursos implementados, como herramientas para la interpretación y ejecución de diseños de instrucción especificados mediante un determinado lenguaje; así como servicios para facilitar la comunicación entre los distintos participantes del proceso o la monitorización de las actividades que llevan a cabo.

- Procesos en los cuales no existe la figura del tutor. En este tipo de procesos el alumno alcanza los objetivos de aprendizaje sin disponer de la asistencia o supervisión de un instructor, bien siguiendo un determinado diseño de instrucción pre-establecido, bien empleando diversos tutoriales o bien utilizando sus propios medios y llevando a cabo las oportunas consultas bibliográficas. Al igual que en el caso anterior, este tipo de procesos de aprendizaje se ha visto favorecido por los avances en las tecnologías de la información, que han vuelto accesibles de manera sencilla e inmediata grandes volúmenes de información.

Dentro de las diferentes tecnologías utilizadas en este tipo de procesos, cabe destacar aquellos programas informáticos que tratan en mayor o menor medida de implementar de manera computacional las tareas que un tutor real lleva a cabo en la enseñanza tradicional. Las soluciones cubren desde sistemas poco interactivos,

Tabla 2.1: Tecnologías aplicables según el rol del tutor en el proceso de enseñanza

Aplicaciones	Instructor	Supervisor	Sin tutor
Programas específicos dominio	X	X	X
Internet	X	X	X
Content Management Systems	X	X	X
Learning Management Systems		X	X
Intelligent Tutoring Systems			X
AIES			X
Electronic Performance Support (EPSS)			X
Text-Based Computer-Based Training (CBT)			X

que únicamente presentan una secuencia determinada de materiales didácticos, aplicaciones capaces de analizar las respuestas del alumno a diversos ejercicios y de proporcionar observaciones adecuadas, hasta sofisticados tutores virtuales capaces de adaptar y personalizar el desarrollo del proceso a las características del usuario actual. Dentro de este grupo se incluirían tanto los Sistemas Tutoriales Inteligentes (ITSs) como los Sistemas de Educación Adaptativos e Inteligentes en Web (AIES).

La tabla 2.1 muestra una relación entre los distintos tipos de procesos de enseñanza-aprendizaje clasificados según el tipo de interacción alumno-instructor y diversas tecnologías que se pueden emplear como soporte a la experiencia educativa.

2.1.2. Necesidad e importancia de la figura del instructor

Cuando se trata de aprovechar los avances tecnológicos para mejorar los procesos educativos, resulta especialmente atrayente la idea de desarrollar sistemas informáticos que implementen agentes pedagógicos virtuales capaces de desarrollar tareas análogas a la que un instructor lleva a cabo y actúen como guía durante el desarrollo del proceso, adaptando el mismo a las características particulares de cada estudiante e incluso al objetivo o tipo de aprendizaje perseguido. Sustituir la acción de un instructor real por un programa informático reportaría innumerables ventajas, eliminando costes asociados al desplazamiento de los alumnos y a las necesidades de profesorado, y permitiendo el desarrollo de procesos individualizados para cada estudiante que pudiesen ser

adaptados a su horario particular y reproducidos independientemente de su ubicación geográfica.

Sin embargo, aún pudiendo disponer de un sistema de estas características, a la hora de diseñar la estrategia pedagógica de un proceso educativo, el modelo de aprendizaje individual no siempre resulta el más adecuado. Desde la perspectiva sociocultural del aprendizaje, la interacción social y el discurso son elementos básicos para el desarrollo de los procesos cognitivos superiores [85]. El aprendizaje constituye un fenómeno social donde los alumnos adquieren los elementos necesarios para alcanzar el conocimiento a través de la interacción con otros alumnos, con los profesores y con el material. Incluso en la educación a distancia se reconocen como elementos claves de la calidad del proceso la interacción, la dimensión social y el diálogo [35].

En cualquier caso, disponer del auxilio de un profesional que dirija la interacción del alumno con el contenido presenta grandes ventajas ya que no es posible prever de antemano cuál será la respuesta de distintos alumnos a un mismo contenido y puede ser necesaria la colaboración de un tutor que aclare dudas y malentendidos, proporcione analogías, recomiende lecturas complementarias y, en general, tome las medidas oportunas para garantizar la comprensión de los conceptos presentados y proporcione soporte a lo largo del proceso. Igualmente, a la hora de aplicar los conocimientos por primera vez, supone una ventaja adicional para el alumno disponer de una figura que guíe el proceso e indique si la forma de aplicarlos es correcta o si están siendo aplicados para todos los casos posibles o tan intensivamente como sería adecuado. Además, el instructor vigila el desarrollo del proceso y organiza evaluaciones que midan el progreso del alumno y ayuden a decidir si es necesario introducir modificaciones en la estrategia o planificación original.

Evidentemente, la calidad y tipo de soporte del instructor dependerá de la personalidad, filosofía y capacidad docente del mismo, pero en cualquier caso resulta difícil imaginar un sistema informático capaz de llevar a cabo esta gama de acciones con la misma eficacia que un tutor real. Esta sustitución total o parcial de las funciones de un tutor por un programa informático resulta especialmente difícil de llevar a la práctica en procesos de las siguientes características:

- Procesos de larga duración: Debido a la propia naturaleza dinámica del contexto del proceso, cuanto mayor sea su duración más fácil será que se produzcan variaciones o cambios en sus componentes. Así, ciertos recursos pueden no encontrarse disponibles, la planificación original puede sufrir alteraciones, se pueden producir

bajas de alumnos, etc. No es posible por lo tanto automatizar la respuesta a la amplia gama de situaciones que pueden tener lugar a lo largo de todo su desarrollo, siendo necesario que la tarea de supervisión del proceso recaiga en manos de un tutor real.

- Procesos de reciente diseño: A pesar de las diferentes pruebas que se hayan llevado a cabo para garantizar el éxito y buen funcionamiento de los nuevos diseños de procesos de aprendizaje o de aquellos diseños que hayan sufrido importantes modificaciones, suele ser necesario supervisar el desarrollo durante las primeras ejecuciones para confirmar que el comportamiento coincide con el esperado.
- Procesos de aprendizaje en los que el éxito es crítico: En estos casos no es posible delegar por completo la responsabilidad en herramientas informáticas, siendo necesario que un tutor humano supervise el desarrollo e introduzca, si es necesario, medidas correctoras que garanticen la consecución de los objetivos marcados.

2.1.3. Tareas del instructor

Una vez analizada la necesidad e importancia de la figura del instructor en un proceso de aprendizaje, pasaremos a estudiar las tareas que desempeña a lo largo del desarrollo del mismo. Tanto estas tareas como el propio rol que el instructor ejerce a lo largo de las diferentes fases del proceso variarán en gran medida dependiendo del modelo de proceso implementado, así como del entorno en que se ejecuta. A pesar de ello, en [27] el autor recogía un conjunto de labores típicas del instructor, que por lo general suele estar presentes en la mayoría de los procesos de enseñanza *on-line*:

- Dar la bienvenida a los participantes del proceso
- Motivar a los alumnos a lo largo del proceso
- Monitorizar su progreso
- Asegurar que los alumnos trabajan al ritmo esperado
- Proporcionar información, complementar, clarificar y explicar
- Proporcionar al alumno realimentación sobre su trabajo
- Asegurarse de que se están alcanzando las metas previstas

- Garantizar el éxito de las reuniones y conferencias
- Facilitar el establecimiento de una comunidad de aprendizaje
- Proporcionar soporte técnico
- Dar el curso por terminado

Este conjunto de tareas pueden clasificarse dentro de las cuatro áreas de actuación o roles distintos del instructor de proceso de aprendizaje que Collins y Berge propusieron en [8]:

- Área pedagógica: Diseñar y proporcionar el material adecuado para la experiencia de aprendizaje, promover la participación y fomentar la reflexión y el aprendizaje en profundidad.
- Área social: Promover la creación de una comunidad de aprendizaje que facilite la construcción de conocimiento, fomentar la cohesión del grupo, monitorizar el progreso de los participantes y participación, etc.
- Área gestión: establecer la agenda, objetivos perseguidos, procedimientos a emplear, etc.
- Área técnica: proporcionar soporte técnico con el objetivo de conseguir hacer transparente a los usuarios la tecnología empleada para soportar el curso.

Estas cuatro áreas fueron posteriormente extendidas por los mismos autores en [20]: facilitador, administrador, promotor, ayudante, participante y *apaga-fuegos*, si bien las cuatro agrupaciones originales son aún ampliamente aceptadas.

En cualquier caso, de una manera más general, podemos decir que la principal función del instructor, independientemente del modelo de proceso de aprendizaje y metodología implementada, es la de velar por el éxito del proceso, vigilando su desarrollo y proveyendo los medios necesarios para facilitar que se alcancen los objetivos de aprendizaje inicialmente marcados para los alumnos.

Cuando el proceso de aprendizaje va a ser soportado por un entorno virtual es necesario garantizar que éste proporciona los medios necesarios para que los diferentes actores del proceso puedan desempeñar sus tareas adecuadamente. Así, desde el punto de vista del soporte a las tareas del instructor, es necesario que el entorno de ejecución proporcione tres tipos distintos de servicios fundamentales:

- Servicios de comunicación: permiten implementar tanto la interacción profesor-alumno como la interacción alumno-alumno. Esta comunicación debe poder llevarse a cabo tanto de manera síncrona como asíncrona.
- Servicios de monitorización: permiten al instructor recuperar información acerca de la evolución del proceso y progreso de sus participantes.
- Servicios de entrega de contenido o gestión del flujo de trabajo: permiten al instructor presentar las actividades a realizar, así como hacer llegar al alumno los materiales necesarios para su desarrollo, y proporcionan un marco para la interacción alumno-contenido.

Una vez más, podemos generalizar la cuestión diciendo que los servicios que el entorno virtual debe proporcionar al instructor deben ser los suficientes para que éste pueda disponer de un control total sobre el desarrollo del proceso, y desempeñar al menos las mismas acciones que llevaría a cabo en un entorno no computacional.

2.2. Adaptaciones en procesos educativos supervisados por instructor

A la hora de implementar un proceso de enseñanza-aprendizaje, el instructor suele utilizar un determinado diseño instruccional que puede incluir la definición de los distintos roles de los participantes del proceso, las tareas asociadas a los mismos, el calendario del proceso, el material que va a ser empleado, evaluaciones, etc. En cualquier caso, es necesario permitir al instructor cierto grado de libertad a la hora de aplicar este plan inicial del proceso a un contexto real de ejecución, entendiendo por contexto la definición proporcionada por Dey: *cualquier información que puede ser empleada para caracterizar la situación de ciertas entidades consideradas relevantes para la interacción entre un usuario y una aplicación, incluyendo al propio usuario y la propia aplicación* [24]. Así, en el caso que nos ocupa, contexto puede ser cualquier información empleada para caracterizar un proceso de aprendizaje, es decir, cualquier información relevante que pueda influenciar su desarrollo. Esta definición de contexto cubre, por tanto:

- Información sobre el entorno computacional del proceso (disponibilidad de dispositivos como impresoras o escáneres, ancho de banda de red, características del ordenador, resolución del monitor, etc).

- Información sobre las características físicas y psicológicas o requisitos de formación previa del usuario (objetivo del aprendizaje, grado de dificultad requerido, interés del alumno, idioma, etc).
- Información sobre las características del entorno físico actual del usuario (localización, condiciones atmosféricas, posibilidad de interacción con otros usuarios, etc).

Al llevar a cabo el diseño del proceso de aprendizaje, los autores deben tener en mente un determinado contexto de ejecución en el que se especificará tanto el perfil del alumno como la necesaria disponibilidad de tiempo y recursos. Tratar de ejecutar el diseño en un entorno de características distintas puede requerir la introducción de adaptaciones para su adecuación, pero aún cuando sea aplicado sobre un contexto similar, la introducción de modificaciones sobre la planificación inicial puede ser necesaria. Por una parte, dos alumnos con un perfil similar pueden responder de manera distinta al desarrollo del proceso. Aspectos como la motivación, personalidad o, de manera más general, el estado mental del alumno pueden influir en su capacidad para asimilar los conceptos presentados. La interacción entre los participantes también varía de un proceso a otro, de manera que el comportamiento de unos influye en el resto modificando el resultado de las actividades. Por otra parte, la disponibilidad de los recursos y materiales didácticos también puede variar durante el propio desarrollo del proceso de aprendizaje, especialmente en aquellos procesos de cierta duración.

Así, por todo ello, los instructores suelen utilizar el diseño instruccional como punto de partida del proceso de aprendizaje, observando la evolución de los participantes durante su desarrollo e introduciendo, en caso necesario, las apropiadas modificaciones para garantizar la consecución de los objetivos pre-establecidos. La importancia de estas modificaciones o adaptaciones radica en que, en la mayoría de los casos, el éxito del proceso depende de ellas.

2.3. Características de las adaptaciones

En [71] los autores distinguen varios aspectos críticos a la hora de llevar a cabo una modificación sobre un sistema software en ejecución: la política del cambio, que determina cómo se introduce la modificación en el sistema en ejecución, el ámbito, que determina los elementos que serán afectados por la modificación, la separación de funcionalidades, que mide el grado de independencia entre las partes afectadas por el

cambio y su influencia en el comportamiento en ejecución del sistema, y el grado de abstracción empleado en la descripción del cambio. Además distinguen dos tipos de cambios: cambios en los requisitos del sistema y cambios en la implementación que no afectan a los requisitos.

Tomando como punto de partida esta clasificación podemos distinguir una serie de características de las adaptaciones sobre un proceso de aprendizaje soportado computacionalmente que deben ser tenidas en cuenta a la hora de implementar un mecanismo que permita su introducción:

- **Predictibilidad:** La necesidad de introducir la adaptación sobre la definición original del proceso de aprendizaje puede o no ser conocida previamente al comienzo del proceso.
- **Momento de introducción:** La adaptación se puede introducir en tres momentos distintos del ciclo de vida del proceso de enseñanza:
 - **Tiempo de diseño:** La adaptación se lleva a cabo modificando directamente el diseño del proceso.
 - **Tiempo de publicación:** La adaptación se introduce en el diseño del proceso en el momento en que éste se carga en el gestor.
 - **Tiempo de ejecución:** La adaptación se lleva a cabo durante la propia ejecución del proceso sin que éste sea interrumpido.
- **Frecuencia:** Una adaptación puede ser introducida de manera puntual en un momento del proceso o por el contrario su aplicación debe ser repetida cada cierto tiempo o cada vez que un determinado evento tiene lugar.
- **Ámbito:** Algunas de las adaptaciones tienen como objetivo modificar el desarrollo del proceso educativo de todos los participantes mientras que otras únicamente deben afectar a la visión que algunos de ellos tienen del mismo. Por otra parte una adaptación puede afectar a la definición de uno o de varios elementos del proceso.
- **Objeto de la modificación:** Ciertas adaptaciones afectan al comportamiento en ejecución del proceso mientras que otras afectan a la definición del mismo. Así

una adaptación puede obligar a los participantes del proceso a repetir una determinada actividad mientras que otra adaptación añade una nueva o modifica el tiempo para completar una ya existente.

- **Perdurabilidad:** Algunas adaptaciones modificarán el desarrollo del proceso educativo quedando integradas de manera permanente en la definición del mismo, mientras que otras lo harán únicamente durante un determinado periodo de tiempo o durante la ejecución de una determinada instancia del proceso.
- **Modificación de objetivos:** Podemos distinguir tres tipos distintos de adaptaciones según cuál sea su influencia en los objetivos de aprendizaje marcados para el proceso:
 - **Adaptaciones leves:** Son aquellas adaptaciones que no modifican los objetivos marcados y que en principio tampoco deberían tener influencia alguna en su consecución. Por ejemplo incrementar el tiempo asignado para completar una evaluación debido a que el recurso que debía ser empleado durante la misma no se ha encontrado disponible momentáneamente.
 - **Adaptaciones moderadas:** Son aquellas que modifican la manera en que se tratarán de alcanzar los objetivos de aprendizaje. Cada material educativo y cada actividad propuesta en la definición del proceso está relacionada con la consecución de uno o más objetivos de aprendizaje, de manera que cualquier modificación en la definición de los componentes del proceso podrá repercutir en la consecución de los objetivos relacionados con los mismos. De igual manera, también influirán aquellas adaptaciones que alteren la secuencia de acciones establecida en el plan inicial del proceso, como por ejemplo, forzar a los participantes a repetir cierta actividad.
 - **Adaptaciones graves:** Son aquellas que introducen cambios en la definición de los objetivos de aprendizaje originales del proceso. Por ejemplo, cuando debido a un reajuste en el calendario el tutor del proceso se ve obligado a eliminar parte del temario original, suprimiendo de esta forma algunos de los objetivos de aprendizaje inicialmente marcados.
- **Propósito:** La introducción de la adaptación obedecerá a distintos objetivos. Partiendo de la clasificación de los diferentes tipos de mantenimientos de software propuesta por [81] podemos clasificar las modificaciones en:

- Correctivas: Aquellas adaptaciones introducidas con el fin de solucionar un determinado problema o error.
 - Adaptativas: Aquellas que persiguen adaptar el proceso a nuevas características del entorno de ejecución, usuario, etc.
 - Perfectivas: Adaptaciones llevadas a cabo con el objeto de mejorar la calidad del proceso.
 - Evolutivas: Las que introducen nuevas funcionalidades con el objeto de dar respuesta a nuevos requisitos.
- Evaluabilidad: dependiendo del propósito y características de la adaptación puede ser posible estimar su grado de éxito. Así por ejemplo, cuando una adaptación es de tipo correctiva, puede ser posible conocer si el error que dio origen a la necesidad de adaptación ha sido subsanado y por tanto evaluar la acción de la adaptación. En cambio, cuando la adaptación es consecuencia de una circunstancia externa al proceso, por ejemplo si una reasignación de aulas obliga a efectuar cambios en el plan original, su evaluación no es posible.

2.4. Enfoques de diseño de sistemas computacionales de soporte al proceso de enseñanza

A la hora de analizar las soluciones informáticas que en las últimas décadas se han desarrollado con objeto de dar soporte a los procesos de enseñanza podemos distinguir dos tipos de enfoques distintos: por un lado el seguido por los *Intelligent Tutoring Systems* (ITS) y los Sistemas de Educación Adaptativos e Inteligentes en Web (AIES) y por otro el empleado en el desarrollo de *Computer-Based Training* (CBT) y *Learning Management Systems* (LMS). En el primer caso el énfasis está en la obtención de sistemas capaces de adaptar automáticamente el proceso de aprendizaje a las características y necesidades del alumno, mientras que en el segundo se persigue, empleando un enfoque más comercial, el desarrollo de entornos virtuales que proporcionen los servicios necesarios para el desarrollo de experiencias educativas.

2.4.1. Sistemas de Educación Adaptativos e Inteligentes en Web

A lo largo de la pasada década se han venido desarrollado una serie de aplicaciones que tratan, en mayor o menor medida, de implementar entornos de aprendizaje

no ligados a la instrucción de un dominio concreto en los cuales la figura del tutor de la enseñanza tradicional es sustituido por un agente pedagógico virtual capaz de actuar como guía en el desarrollo del proceso de aprendizaje y adaptar el mismo a las características del estudiante. Estas aplicaciones tienen en común el uso de tecnologías relacionadas con la Web y técnicas provenientes tanto de los ITSs como de los Sistemas Hipermedia Adaptativos (SHA) y son denominados AIEs [12].

Un ITS es una aplicación informática que provee al alumno de una instrucción individualizada en un dominio concreto empleando como estrategia de aprendizaje fundamental la tutorización: el sistema presenta material al alumno, lo evalúa y en caso necesario le provee de la apropiada respuesta o material de refuerzo. Para poder implementar dicho método normalmente hacen uso de diversas técnicas de inteligencia artificial que facilitan, entre otras cosas, la representación del conocimiento de un área concreta. Además, las estrategias más comunes empleadas en este tipo de soluciones son:

- Secuenciación del currículum: proveer al alumno de un orden de presentación de unidades y tareas de aprendizaje adecuado a sus características y conocimientos.
- Análisis inteligente de las soluciones: consiste en estudiar las respuestas de los alumnos a diversos problemas planteados, valorando si son correctas o incorrectas e indicando posibles causas de los errores cometidos.
- Soporte de resolución de tareas interactivo: proveer al alumno de un mecanismo de auxilio y soporte en la resolución de determinados problemas.
- Soporte colaborativo: usar el conocimiento del sistema sobre diversos usuarios para formar grupos de aprendizaje colaborativo de tal manera que los alumnos compartan información y se ayuden entre sí.

El uso de técnicas como el análisis inteligente de soluciones y el soporte de resolución de tareas interactivo hace que las soluciones proporcionadas por este tipo de sistemas estén fuertemente ligadas al dominio del problema, no siendo re-utilizables, normalmente, para la enseñanza en otros dominios. Además, el tipo de instrucción que implementan restringe el tipo de áreas de conocimiento donde este tipo de sistemas pueden ser empleados.

Por otra parte los SHA son sistemas hipermedia capaces de adaptar tanto el contenido de sus páginas (adaptación de la presentación) como la manera de guiar a través

de ellas (adaptación de la navegación) de la forma más adecuada a las necesidades, conocimiento y características del usuario que accede [11].

De todas estas técnicas las más comunes en los AIES son la secuenciación del currículum, adaptación de la presentación, adaptación de la navegación y soporte colaborativo, apareciendo a menudo todas ellas combinadas entre sí con un mismo fin: proveer al alumno con la secuencia de material educativo que optimice su proceso de aprendizaje. Así, algunos sistemas son capaces de presentar diferentes versiones del mismo material educativo dependiendo de las características del alumno (SHIVA [96], ARTHUR [31]), de diseñar rutas didácticas distintas según cuáles sean sus conocimientos iniciales (INSPIRE [72]), de modificar dichas rutas iniciales teniendo en cuenta los conocimientos que el alumno va adquiriendo a lo largo del proceso (TANGOW [17]), o incluso de ofrecer distintos caminos didácticos según el grado de profundización o tipo de conocimiento que se desee alcanzar sobre el mismo objetivo de aprendizaje y re-planificar el camino si el alumno encuentra dificultades para asimilar un determinado concepto presentado (DCG [84]).

Una característica común que comparten casi todos los sistemas pertenecientes a esta categoría, y que es en gran parte heredada de los ITS, es la necesidad de separar los diferentes tipos de información y representarla empleando diversos modelos. Estos modelos, por lo general, suelen ser tres: modelo de dominio o de conocimiento, modelo de usuario, y modelo pedagógico o de adaptación; aunque algunos sistemas como APeLS [21] ofrecen la posibilidad de definir modelos adicionales que permitan la representación de información contextual y adaptar así el proceso al tipo de dispositivo empleado por el usuario, por ejemplo. En cualquier caso, la implementación de cada tipo de modelo difiere de un sistema a otro, ofreciendo cada uno de ellos soluciones distintas. Así, los tipos de relaciones entre conceptos capturadas en los modelos de dominio pueden bien limitarse a meras relaciones de pre-requisito, incluir relaciones más complejas como las de agregación (por ejemplo ACE [80] y TANGOW), o bien, ser representadas en forma de grafos AND/OR (DCG). La representación del conocimiento del usuario puede llevarse a cabo mediante modelos de estereotipos o mediante *modelos overlay* [16] más precisos pero difíciles de manejar, y por último, los modelos de adaptación pueden permitir diferentes grados de actualización de las reglas que rigen la estrategia pedagógica implementada por el sistema.

Esta diferencia de enfoques entre las distintas soluciones hace que la interoperabilidad de contenidos entre ellos sea difícil de conseguir y que el material deba ser

desarrollado teniendo en cuenta las características específicas de cada sistema. Además, si bien la mayoría de ellos han sido desarrollados tratando de obtener soluciones generales, el uso de ontologías en los modelos de conocimiento suele mantenerlos ligados a dominios concretos.

2.4.2. Sistemas de gestión del aprendizaje (LMS)

Los *Learning Management Systems* (LMS) o Sistemas de Gestión del Aprendizaje son aplicaciones software que automatizan las tareas de administración, gestión y monitorización de procesos educativos. Las prestaciones y componentes de un LMS varían según la plataforma, pudiendo incluir:

- Gestión del calendario del curso.
- Gestión de los contenidos del curso.
- Elementos que faciliten la comunicación entre los participantes del curso: foros, chats, videoconferencia, etc.
- Mecanismos para generar visiones personalizadas del curso.
- Monitorización de las actividades de los estudiantes.
- Mecanismos de evaluación.

El origen de los LMS se encuentra en los *Content Management Systems* (CMS), o sistemas de gestión de contenidos, que se comenzaron a desarrollar a mediados de la década de los noventa y que tenían por objeto facilitar la creación de forma colaborativa de documentos y contenidos, así como su gestión, publicación y presentación. La mayoría de los CMSs están formados por un repositorio central donde se almacenan los contenidos y una aplicación Web que permite la gestión de los mismos, de tal forma que el aspecto del *website* podía ser actualizado por cualquier usuario autorizado desde cualquier ordenador conectado a Internet. Ejemplos de CMS comerciales son Vignette, Interwoven, Documentum, Microsoft CMS o Hyperwave y dentro del grupo de software libre OpenACS, Plone, PHP-Nuke y OpenCMS.

El uso de CMSs fue rápidamente adoptado en el entorno educativo tanto como complemento a la educación presencial, permitiendo a los usuarios acceder y recuperar los recursos empleados durante el curso, como dando soporte a la educación no presencial,

sirviendo para almacenar y distribuir los mismos cursos que antes eran comercializados empaquetados en CD-ROMS. Pronto se desarrollaron nuevas tecnologías con el fin de aumentar las prestaciones ofrecidas por los CMSs, de tal forma que se permitiese administrar los distintos cursos y los recursos asociados a los mismos, interactuar con los alumnos, obtener informes acerca de su progreso y resultados, etc, dando lugar así al nacimiento de los primeros *Learning Management System* (LMS).

Dentro de este tipo de sistemas se suele distinguir bajo el término *Learning Content Management Systems* (LCMS) aquellos sistemas que, proveyendo prestaciones parecidas a los LMSs, se centran en la gestión de los contenidos didácticos más que en la gestión de cursos. En la práctica, la distinción entre los dos tipos de sistemas es mínima y a menudo se utilizan ambas expresiones para designar a las mismas aplicaciones. Dentro de esta categoría de sistemas los más conocidos serían WebCT y Blackboard, como entornos comerciales, y Moodle y Claroline como software libre.

De manera paralela al desarrollo de este tipo de aplicaciones surgió la necesidad de establecer estándares para el desarrollo de los contenidos educativos, de tal forma que se garantizase la interoperatividad de componentes entre plataformas desarrolladas por distintos fabricantes y de esa forma se preservasen las inversiones realizadas.

En el resto de esta sección analizaremos algunos de los estándares y especificaciones más difundidos y dos de los LMS de uso más extendido en la actualidad: Moodle y Blackboard.

2.4.2.1. Estándares

Dado el alto coste asociado a la generación de recursos educativos y la proliferación de sistemas desarrollados por distintos fabricantes, surge la necesidad de establecer acuerdos a la hora de desarrollar plataformas y contenidos de tal forma que se garantice la interoperabilidad y reutilización de los recursos desarrollados.

Con este fin, diversas organizaciones han desarrollado propuestas para la estandarización de diversos aspectos del *eLearning*, entre las cuales cabe destacar:

- IEEE LTSC (*Learning Technology Standards Committee of the Institute of Electrical and Electronics Engineers*) [58]: Comité formado por más de una docena de grupos de trabajo y grupos de estudio encargados de desarrollar estándares técnicos, recomendaciones y guías para la tecnología educativa.
- AICC (*Aviation Industry Computer-Based Training Comitee*) [3]: Asociación in-

ternacional que desarrolla directrices para la producción y evaluación de tecnologías relacionadas con la formación para el campo de la industria de la aviación, pionera en la estandarización de materiales para la formación profesional.

- *IMS Global Consortium Inc* [39]: Reúne a un conjunto de organizaciones comerciales, educativas y gubernamentales con el propósito de desarrollar especificaciones que sean ampliamente aceptadas y que permitan la interoperabilidad entre contenidos y entornos de aprendizaje desarrollados por distintos fabricantes.
- *ADL (Advanced Distributed Learning)* [1]. Iniciativa del Departamento de Defensa del gobierno de los Estados Unidos y de la Oficina de Ciencia y Tecnología de la Casa Blanca para desarrollar principios y directrices de trabajo para el desarrollo e implementación de formación educativa sobre tecnologías Web.

Los estándares y especificaciones propuestos por estos organismos cubren diversos aspectos relacionados con el desarrollo de material educativo incluyendo la manera de describir sus características y contenido, formatos de empaquetado, comunicación con el LMS, descripción de evaluaciones, descripción de perfiles de alumno, etc. A continuación describiremos aquéllos cuyo uso está más extendido.

Learning Object Metadata (LOM)

Es un esquema de metadatos para la descripción de recursos educativos desarrollado por el IEEE LTSC. Partiendo de la definición de *Learning Object* como *cualquier entidad, digital o no digital, que puede ser usada, re-usada o referenciada durante un proceso educativo tecnológicamente soportado* [37], este estándar define un conjunto de atributos para su precisa descripción y categorización de tal forma que puedan ser buscados, clasificados y fácilmente recuperados según criterios a través de catálogos e inventarios, así como compartidos e intercambiados a través de distintas plataformas de aprendizaje. El modelo de datos de LOM está compuesto por una jerarquía de elementos de datos, simples o agregados, que son agrupados en nueve categorías principales: *general, lifecycle, meta-metadata, technical, educational, rights, relation, annotation y classification*.

Actualmente existen en el mercado diversas aplicaciones tanto de software libre (*Reload Editor* [76]) como comerciales (*Macromedia-Authorware* [60]), que facilitan la tarea de descripción de objetos de aprendizaje según el estándar LOM, pero a pesar de

ello, y dada la gran cantidad de entradas a completar, lo más común es que los autores únicamente cumplimenten una mínima parte de los más de 100 metadatos posibles.

IMS Content Package

De la necesidad de intercambiar contenidos entre diversos sistemas de aprendizaje, de autoría de contenidos y entornos de ejecución, surge esta especificación cuyo propósito es acordar un formato para el empaquetado de recursos educativos. El elemento clave de esta especificación es el *paquete*, el cual está compuesto por un conjunto de ficheros físicos que pueden corresponderse con varios recursos independientes, un curso completo o varios cursos a la vez, y un fichero XML denominado manifiesto que, por una parte, describe los diversos contenidos y, por otra, una o más posibles organizaciones de los mismos [43]. Estas organizaciones describen de manera jerárquica las relaciones entre los contenidos, sin especificar ningún tipo de orden de navegación o presentación de los mismos. Cuando un paquete se encuentra comprimido en un solo fichero (e.g. .zip, .jar, .cab) se denomina *Package Interchange File (pif)*. Esta especificación ha sido comercializada por Microsoft bajo el nombre de LRN (*Learning Resource Interchange*[66]).

IMS QTI

La especificación *IMS Question & Test Interoperability* [44], es una iniciativa del IMS que tiene como objeto proporcionar un modelo de datos que permita la representación de tests así como sus correspondientes resultados. Está compuesta por dos modelos de información distintos: por un lado el modelo de información ASI (*Assessments Sections Items*), que especifica cómo definir los tests y sus componentes, y por otro el modelo de *Information Results Reporting*, que especifica el modo de describir las respuestas a las posibles interacciones del usuario con los componentes de los test.

La mínima unidad intercambiable en el modelo de información ASI es el *item*, mediante el cual se especifica una determinada pregunta y se proveen instrucciones acerca de cómo debe presentarse, cómo debe tratarse su respuesta y qué información de *feedback* se debe proporcionar a la misma. Los *items* pueden ser agrupados para formar secciones y test (*assessments*) y todos ellos son susceptibles de ser apropiadamente marcados con metadatos y almacenados en bancos de objetos así como intercambiados entre distintos LMSs.

Existen multitud de editores para la generación de elementos QTI algunos de los cuales, como por ejemplo *Canvas Learning* [15], permiten almacenar automáticamente los items generados en bases de datos y así generar fácilmente bancos de objetos.

IMS Learner Information Package

La especificación *IMS Learner Information Package* [40] está basada en la propuesta de estandarización de información de alumno PAPI (*Public and Private Information*) [36] desarrollada por el IEEE. Su objetivo es definir una estructura de datos que permita describir, almacenar y compartir información acerca de un alumno, grupo de alumnos o productos de contenido educativo, entre distintos entornos de aprendizaje u otros sistemas que participen en el proceso educativo. De esta manera podemos disponer de un perfil que incluya tanto información sobre las características y preferencias del alumno, como un registro del progreso y formación obtenida según vaya participando en distintas experiencias educativas.

La información sobre el alumno se almacena así en un fichero XML con su mismo nombre y que se encuentra dividido en once secciones distintas según el tipo de información que almacenen: *identification*, *goal*, *qcl* (*qualifications, certifications and licenses*), *activity*, *transcript*, *interest*, *competency*, *affiliation*, *accessibility*, *securitykey* y *relationship*. El propio alumno es el responsable de definir qué parte de la información puede compartirse entre sistemas y cuál no.

SCORM, Shareable Content Object Reference Model

SCORM [2] es una propuesta de ADL que integra una colección de especificaciones y estándares desarrollados por organizaciones como el IMS, AICC, ARIADNE [6] y el IEEE LTSC en un único modelo de referencia. Utiliza el término SCO para designar la mínima agrupación de contenidos capaz de ser ejecutada y comunicarse con un LMS. La especificación está compuesta por:

- Entorno de ejecución: compuesto por un protocolo específico que seguirán los LMS en la ejecución de los SCOs, un modelo de datos que especifica un conjunto mínimo de información sobre el SCO que podrá registrarse en diferentes LMS y un API (*Application Programming Interface*) que establece un conjunto de funcionalidades predefinidas que permitirán la comunicación entre el SCO y el

LMS.

- Modelo de agregación de contenidos: permite la composición, etiquetado y empaquetado del contenido educativo. Emplea la especificación *IMS Content Package* para agrupar los contenidos educativos bajo una determinada organización descrita en el fichero manifiesto.
- Modelo de secuenciación y navegación: incluye un modelo de información derivado de la especificación *IMS Simple Sequencing* [42], que permite a los desarrolladores describir diferentes secuenciaciones del material educativo, y un modelo de navegación que define un conjunto de eventos que podrán ser disparados por el alumno y que a su vez activarán las distintas secuencias de contenidos.

IMS Common Cartridge

Para primavera de 2007 está prevista la publicación de una nueva especificación, *IMS Common Cartridge* [38], que combina tres de las especificaciones de *eLearning* más difundidas en la actualidad (*IMS Content Packing*, *IMS Question & Test Interoperability* y *Learning Object Metadata*) y proporciona compatibilidad con SCORM 1.2 y SCORM 2004. Este nuevo estándar nace con el propósito de convertirse en el estándar de contenidos de mayor aceptación del mercado y cuenta para ello con el apoyo tanto de los más importantes productores de contenido (McGraw-Hill, Digital Spirit, Thomson, etc.) como de las más importantes plataformas de *eLearning* (ANGEL, Blackboard, Desire2Learn, etc.)

2.4.2.2. Herramientas

Moodle

Moodle (*Modular Object-Oriented Dynamic Learning Environment*) [69] es un LMS disponible *open source* que permite el diseño de cursos y proporciona un entorno para su ejecución en la Web. Moodle está basado en los principios pedagógicos constructivistas [26], y su diseño modular permite la fácil inserción y modificación de la estructura de los cursos.

Los escenarios de aprendizaje contruidos mediante esta herramienta están contruidos por una secuencia de actividades definidas por el administrador del curso, cada una de las cuales puede hacer uso de los distintos tipos de módulos proporcionados

por Moodle: tareas, chats, foros, lecciones, cuestionarios, talleres, etc. Además de proporcionar facilidades para la gestión de contenidos, permite que los administradores y profesores asignados al curso puedan llevar a cabo las distintas tareas relacionadas con la administración de alumnos (creación de grupos, calificación, seguimiento, etc) así como la gestión del calendario del curso.

Moodle proporciona un mecanismo sencillo para que profesores sin amplios conocimientos tecnológicos sean capaces de crear y gestionar sus propios cursos Web. El instructor puede partir de un diseño pre-establecido del curso que luego puede modificar según sea su desarrollo, o bien ir generando dicho diseño sobre la marcha publicando las actividades del curso semana a semana.

En cuanto a compatibilidad de Moodle con estándares de aprendizaje, el modulo SCORM permite el uso de paquetes de este tipo como una actividad más de Moodle. Además es posible importar y convertir test en formato IMS QTI. Próximas versiones de Moodle permitirán el uso de recursos de tipo *IMS Content Package* y la exportación de diseños de cursos en un formato compatible con *IMS Learning Design*. En un futuro se contempla ampliar la compatibilidad con esta última especificación y permitir también la importación de UoLs.

Blackboard

Blackboard Learning System integra un conjunto de aplicaciones software que permite a instituciones académicas, corporaciones y proveedores comerciales de educación ofrecer sus servicios educativos a través de un entorno virtual en línea. Es una herramienta de carácter privado financiada por un grupo de inversores, medios y compañías especializadas en tecnología educativa. El sistema está disponible con diversas opciones de licencia y es con probabilidad el LMS comercial con mayor presencia en el mercado, ofreciendo un entorno virtual de aprendizaje robusto donde llevar a cabo tareas de evaluación, seguimiento del alumno, colaboración virtual, asignación y gestión de “portafolios”, etc. Entre las claves de su popularidad se suelen mencionar la simplicidad de su interface y su alto grado de escalabilidad.

De manera similar a Moodle, Blackboard está constituido sobre una arquitectura de *building blocks*, de tal forma que el núcleo de la funcionalidad del sistema está incluido en la versión estándar del mismo, que puede ser extendida posteriormente a través de distintos servicios y prestaciones ofrecidos por los correspondientes *bloques*. La flexibilidad proporcionada por esta arquitectura constituye una ventaja sobre los

competidores directos de la plataforma, a la par que permite facilitar la integración de la misma con el entorno informático de la institución que la implemente, ya que de hecho, muchos de los servicios ofrecidos por los *bloques* disponibles tratan de enlazar Blackboard con funcionalidades de software relacionado, como por ejemplo, permitir la integración con foros creados y ejecutados desde Webboard. De igual forma, este sistema es el empleado para soportar la compatibilidad de Blackboard con los emergentes estándares de la industria del *eLearning*, siendo a día de hoy capaz de importar y trabajar con estándares de contenido como SCORM, *IMS Content Packaging* o *IMS Question & Test Interoperability* o estándares de marcado de metadatos como *IMS Metadata*, y en el futuro con nuevos estándares como *IMS Commom Cartridge*.

2.5. Lenguajes de Modelado Educativo

Para describir un proceso de enseñanza-aprendizaje no es suficiente con especificar cuales son los componentes y recursos que van a ser empleados en el mismo es también necesario describir explícitamente las actividades que serán llevadas a cabo y el uso que de los recursos educativos se hará en cada una de ellas, los roles de los participantes en dichas actividades, así como cuáles son las características que el entorno del proceso debe cumplir para que esas actividades puedan ser llevadas a cabo de forma adecuada. Los Lenguajes de Modelado Educativo (EMLs) surgen con el fin de poder llevar a cabo este tipo de descripciones cubriendo diferentes tipos de enfoques pedagógicos. Según la definición propuesta por el *European Committee for Standardization* (CEN) en el estudio que sobre este tipo de lenguajes llevó a cabo [74], un lenguaje de modelado se compone de *un modelo semántico de información y una recomendación para su implementación, que permiten la descripción del contenido y modo de procesamiento de unidades de aprendizaje de tal manera que se facilite su reutilización e interoperabilidad*. En esta sección se analizarán las características principales de dos especificaciones que satisfacen plenamente esta definición, *IMS Learning Design* y *PALO*, si bien actualmente existen otras propuestas de EMLs como:

- *PALO* [78]: Desarrollado por la Universidad Nacional de Educación a Distancia (UNED), organiza los cursos en distintos módulos a su vez compuestos de actividades de aprendizaje y contenido. Los elementos del lenguaje se agrupan en plantillas instruccionales que definen distintos escenarios de aprendizaje con distintas propiedades pedagógicas.

- OUNL-EML [53]: Desarrollado por la *Open Universiteit Nederland* con el fin de permitir la descripción de diseños instruccionales que fuesen capaces de capturar diferentes enfoques pedagógicos y que a su vez permitiesen la especificación de cursos que serían impartidos en modalidad no presencial. Fue usado como punto de partida para el desarrollo del *IMS Learning Design*.
- LAMS (*Learning Activity Management System*) [56]: Inicialmente construido basándose en la especificación IMS LD, finalmente adoptó sus propias soluciones para posibilitar la creación de diseños de aprendizaje formados por conjuntos de actividades secuenciales, opcionales o paralelas. Ofrece tanto un entorno de ejecución para los procesos como una aplicación visual que permite su diseño y creación mediante la selección de una serie herramientas que ofrecen los servicios necesarios para el desarrollo de distintos tipos de actividades, como debates, entregas de material, discusiones, etc.
- LDL (*Learning Design Language*) [62, 63]: Iniciativa elaborada desde un enfoque centrado en el instructor desarrollada por las Universidades de Savoie y Grenoble como una alternativa al IMS LD. Parten del concepto de actividad pedagógica como conjunto de intercambios que suceden entre los participantes de la actividad, del cual el instructor sólo puede proporcionar una descripción a priori del desarrollo deseado. Dado el carácter evolucionario e impredecible de la actividad, proponen un EML que integre completamente dentro de la propia especificación las actividades de adaptación y observación y que facilite el modelado de actividades colaborativas.
- CDF [6]: Construido siguiendo el mismo enfoque que *Ariadne Educational Metadata Scheme*, permite generar cursos *on-line* a partir de descripciones contenidas en ficheros XML que un LMS se encargará de interpretar a la vez que recupera los correspondientes recursos en el fichero referenciados. Únicamente permite trabajar con material didáctico en formato texto.
- LMML (*Learning Material Mark-up Language*) [57]: Generado a partir de una implementación en XML de un meta-modelo de gestión de conocimiento para el dominio específico de la enseñanza, agrupa el contenido educativo en distintos módulos que a su vez son organizados jerárquicamente. Provee sub-lenguajes para distintos áreas de la educación

- XEDU [28]: Incluye un modelo para llevar a cabo diseños instruccionales que disocian actividades de aprendizaje y contenidos, permitiendo la especificación de escenarios de aprendizaje genéricos que pueden ser enlazados con contenidos específicos en tiempo de ejecución.

2.5.1. PALO

Es un Lenguaje de Modelo Educativo desarrollado por el departamento de lenguajes y Sistemas de la Universidad Nacional de Educación a Distancia UNED.

Los cursos se encuentran organizados en módulos que a su vez están compuestos de actividades de aprendizaje, contenido y plan de estudio. El modelo pedagógico de PALO está basado en el uso de plantillas instruccionales formadas por un conjunto de elementos del lenguaje que proveen una funcionalidad específica y un determinado esquema pedagógico. La descripción de un proceso de aprendizaje empleando PALO supone primero, la selección de la plantilla instruccional apropiada y segundo, la descripción del proceso empleando los elementos de la misma.

Los elementos del modelo de información se encuentran clasificados en cinco niveles distintos de abstracción, de tal forma que una determinada plantilla contendrá una selección de elementos de cada nivel:

- Nivel de Contenido: Contenido educativo del curso.
- Nivel de Tareas: Descripciones de las distintas actividades educativas que se pueden llevar a cabo durante el proceso.
- Nivel de estructura: Define la estructura jerárquica del entorno de aprendizaje (curso, módulos, partes, etc).
- Nivel de planificación: Define las restricciones temporales, el orden de ejecución y las dependencias pedagógicas existentes entre los distintos elementos.
- Nivel de gestión: incluye información para administrar los elementos y desarrollar el entorno de aprendizaje en un determinado formato.

La implementación de los distintos escenarios se lleva a cabo empleando un lenguaje de marcado basado en SGML. PALO provee una librería de DTD's, cada una de las cuales se corresponde con una determinada plantilla instruccional. Un compilador de PALO procesará una instancia de una de las plantillas y generará a partir de la descripción en ella contenida el consiguiente entorno de aprendizaje.

2.5.2. IMS Learning Design

IMS Learning Design (IMS LD) permite la creación de descripciones completas, abstractas y portables del enfoque pedagógico a emplear a lo largo de un determinado curso, de forma que puedan ser luego puestas en práctica a través de motores de ejecución compatibles con la especificación [54]. Está basado en el OUNL EML [52] y proporciona los elementos necesarios para la descripción de las actividades que desempeñarán los distintos participantes de un proceso educativo. Además integra otras muchas especificaciones como el *IMS Content Packaging* [43], *IMS Simple Sequencing* [42], *Learning Objects Meta-Data Specification* [37], etc. Cuando una descripción de un LD se incluye como organización en el fichero manifiesto de un *Content Package* obtenemos una *Unit of Learning* (UoL). Una UoL contiene toda la información requerida para llevar a cabo un proceso educativo, incluyendo tanto información pedagógica como información sobre localización y empleo de los recursos necesarios. Una vez definida, el programa adecuado interpretará dicha información y proveerá a los participantes del proceso con el apropiado interfaz para desarrollar las actividades propuestas.

Para la descripción de un proceso de aprendizaje, el IMS LD utiliza una analogía con una obra de teatro de tal forma que, al igual que una obra, un proceso de aprendizaje puede ser interpretado por distintos participantes haciendo uso de distintos recursos [47]. Cada participante desempeñará un rol determinado, *learner* o *staff*, dentro de una serie de actos (*acts*) llevando a cabo ciertas actividades. Dichas actividades se desarrollan empleando los *learning objects* y/o servicios (chats, foros, buscadores, etc) que conforman el entorno (*environment*) de cada actividad. Un acto termina cuando todos los participantes han llevado a cabo las actividades especificadas para cada uno de sus roles o bien cuando se sobrepasa un determinado límite de tiempo. Al finalizar un acto da comienzo el siguiente. Los *plays* agrupan a los actos y sólo concluyen una vez que todos ellos finalizan. Dentro de un mismo *Learning Design* puede haber varios *plays* ejecutándose de manera concurrente.

La especificación del IMS LD define tres niveles distintos de implementación y conformidad. El nivel A contiene el núcleo del LD, proporcionando un vocabulario para especificar la secuencia de actividades que los profesores y alumnos que toman parte en el proceso deben llevar a cabo, mientras que los niveles B y C permiten a los desarrolladores definir una secuenciación más elaborada de las actividades empleando propiedades, condiciones y notificaciones. Las propiedades se utilizan para capturar información sobre la persona, rol o el propio desarrollo del proceso, pudiendo ser ex-

ternas o internas según su valor persista o no una vez que la ejecución del diseño de aprendizaje finalice. A partir de las propiedades, se pueden establecer condiciones que dividan el flujo de ejecución según el valor actual de las primeras. Por su parte, las notificaciones permiten la especificación de respuestas a la ocurrencia de ciertos eventos, como por ejemplo que cuando un estudiante termine una determinada actividad automáticamente se notifique la situación a un profesor comenzando una nueva para éste.

La introducción de adaptaciones en el diseño de aprendizaje se consigue por medio del nivel B, de tal forma que los diseñadores pueden definir diferentes comportamientos en tiempo de ejecución que representen diferentes escenarios y que se disparan según determinados valores de ciertas propiedades. De esta manera, dentro de la misma UoL es posible implementar diferentes formas de interacción y orden de presentación de materiales para diferentes perfiles de alumnos. Esto contribuye a incrementar la reusabilidad de las UoLs ya que su diseño se adecuará a un mayor rango de posibles escenarios. En cualquier caso, la implementación de la lógica de adaptación dentro de la propia UoL comprende ciertas limitaciones ya que, una vez terminada la etapa de diseño, el manifiesto no se puede modificar para incluir nuevas estrategias de adaptación [82]. Esto significa que todas las posibles variaciones del proceso se deben especificar de manera previa a la ejecución de la UoL o ésta se deberá re-diseñar cada vez que sea necesario actualizar la lógica de la adaptación.

La especificación incluye un modelo de información [41] que detalla en lenguaje natural las características de todos los elementos del *Learning Design*, incluyendo las restricciones que cada uno de ellos debe cumplir, y por otra parte un esquema XML donde el vocabulario y las relaciones entre los elementos quedan definidos.

Actualmente existen varias aplicaciones que permiten la autoría de UoL. La más completa es el editor desarrollado dentro del marco del proyecto RELOAD, el cual permite la edición de manifiestos de nivel A, B y C mediante una interfaz gráfica. El diseñador debe rellenar uno por uno los distintos formularios correspondientes a los distintos elementos del *Learning Design*, asociando diferentes recursos a los mismos. Por otra parte, el editor de modelado de conocimiento MOT PLUS(tm) [19], desarrollado por el centro de investigación LICEF de la Universidad a distancia de Montreal, permite la creación de manifiestos de nivel A empleando representaciones gráficas.

Una vez que la UoL ha sido construida es necesario disponer de un software que se encargue de interpretar su contenido y presentar las diferentes actividades y recur-

so a los participantes del proceso de aprendizaje, controlando al mismo tiempo sus interacciones. Dentro de este tipo de software, denominado *Learning Design Player*, la OUNL ha desarrollado una aplicación *open source* basada en tecnología J2EE llamada CopperCore [64] que es capaz de interpretar los tres niveles del *IMS Learning Design*. Dicha aplicación ha sido diseñada para ser integrada dentro de un sistema e-learning e incluye herramientas que permiten la validación de UoLs, así como una interfaz y línea de comandos a través del cual es posible crear y borrar instancias de una UoL, así como dar de alta usuarios y asignarles determinados roles. El proyecto Reload también cuenta con un desarrollo de este tipo, si bien a día de hoy únicamente es capaz de reproducir manifiestos de nivel A y no es posible dar de alta nuevos usuarios aparte del *dummy* incluido por defecto.

2.6. Tecnologías relacionadas

Para terminar este capítulo se revisarán algunas tecnologías que bien por sus objetivos, enfoque o sus características se encuentran relacionadas con la problemática planteada en el presente documento. Por un lado, se analizará el sistema desarrollado dentro del marco del proyecto aLFanet, el cual combina el uso de técnicas de inteligencia artificial con el empleo de estándares educativos con el objetivo de proveer soporte a todos los actores del ciclo de vida de un proceso educativo, haciendo especial hincapié en la adaptación del mismo a las características del alumno. Por otro lado, se examinará LOCO-Cite, un sistema que emplea ontologías para conseguir la adaptación de los procesos de aprendizaje a su contexto de uso.

2.6.1. aLFanet

Active Learning For Adaptive interNET (aLFanet) es un proyecto IST subvencionado por la Comisión Europea que propone un sistema para proporcionar soporte a todos los participantes de un proceso educativo, es decir tanto diseñadores como tutores y alumnos, a lo largo de todas las fases del mismo. Su objetivo es ofrecer un entorno de aprendizaje capaz de adaptarse a las características específicas del alumno que lo utiliza. Ha sido desarrollado combinando el uso de técnicas de Inteligencia Artificial con estándares del campo educativo así como tecnologías XML y Java.

La definición de los procesos de aprendizaje se lleva a cabo empleando la especificación IMS LD. Los diseñadores disponen de una herramienta de autoría y diferentes

plantillas correspondientes a distintos escenarios, de manera que se simplifica el proceso de creación de cursos. IMS LD provee las bases para poder llevar a cabo las distintas tareas de personalización y adaptación del proceso y permite además la integración de contenidos descritos empleando otras especificaciones. Además de IMS LD, aLFanet utiliza:

- *IMS Learner Information Package* (LIP): El modelo de usuario del alumno es implementado empleando una extensión de este estándar. Dicho modelo almacena tanto información acerca de las características del alumno (modalidad cognitiva y estilo de aprendizaje) como información acerca del grado de conocimiento que sobre los diversos objetivos de aprendizaje marcados va alcanzando a lo largo del progreso. Algunos de los datos del modelo se obtienen mediante preguntas directas al alumno, mientras que otros se calculan a partir de la monitorización de sus acciones y aplicación de diversas reglas de inferencia y técnicas de aprendizaje automático.
- *IMS Question and Test Interoperability* (QTI): se utiliza para la definición de las preguntas y los test de las distintas evaluaciones.
- *IMS Meta Data*: Se emplea para el marcado del material educativo con información relativa a los objetivos de aprendizaje con los que se encuentra relacionados, parte del curso a cuyo conocimiento contribuyen y tipo de utilización (material opcional, de refuerzo u obligatorio), permitiendo de esta forma su selección a la hora de generar recomendaciones. De igual manera, los QTI *items* se clasifican incluyendo, entre otros datos, el objetivo de aprendizaje que evalúan y grado de dificultad. También se emplea para marcar el propio curso con las normas derivadas de los factores críticos de éxito.

El sistema es capaz de generar cuatro tipos de adaptaciones distintas a partir de un diseño genérico de un curso:

- Adaptación a los conocimientos previos del usuario: Cada curso está asociado con una serie objetivos de aprendizaje relacionados a su vez con diferentes actividades (IMS LD) y representados mediante propiedades que simbolizan el grado de conocimiento alcanzado por el alumno para cada uno de ellos. Previamente al comienzo del curso el alumno completa un test de tal manera que se recupera información acerca de su conocimiento actual sobre el dominio de estudio. De

esta forma las propiedades que representan los objetivos de aprendizaje se cargan con un valor inicial que determinará si las actividades asociadas al objetivo no se muestran, si únicamente se presentan las evaluaciones o si se muestra el contenido en su totalidad.

- Adaptación a la modalidad cognitiva y estilo de aprendizaje del usuario: Para conseguir este tipo de adaptación el curso incluye varias versiones del mismo contenido desarrolladas de acuerdo a las combinaciones entre distintas modalidades cognitivas (inductiva o deductiva) y estilo de aprendizaje (verbal o visual) de los alumnos. Previamente al comienzo del curso un test determinará la modalidad y estilo del alumno en cuestión de tal forma que únicamente se le presentarán aquellos contenidos que se correspondan a sus características.
- Adaptación de los test: A lo largo del curso el alumno llevará a cabo al final de cada módulo una serie de evaluaciones que el sistema empleará para determinar el grado de asimilación de los conceptos presentados. Además tendrá a su disposición una serie de test de autoevaluación que cumplen una doble función: por una parte sirven al alumno para conocer su propia evolución durante el desarrollo del proceso de aprendizaje y por otra parte los resultados obtenidos son analizados por el sistema de tal forma que automáticamente adapta las evaluaciones para no incluir en ellas preguntas sobre temas que el alumno ha mostrado dominar.
- Adaptación mediante recomendaciones al alumno: A lo largo del curso una arquitectura de agentes software permite monitorizar en todo momento las acciones del usuario así como diagnosticar la situación del mismo frente a un determinado objetivo de aprendizaje distinguiendo entre dos tipos de situaciones posibles: falta de conocimiento y alto grado de interés. Según el alumno se encuentre en una u otra situación, el sistema podrá sugerirle cierto material marcado como de refuerzo o extensión de esa temática, mostrar las decisiones que otros alumnos tomaron en casos parecidos o bien le pondrá en contacto con otros estudiantes que alcanzaron una alta puntuación en ese módulo.

Además, el sistema proporciona soporte a los diseñadores mediante un módulo de auditoría cuyo objetivo es permitir que conozcan si el comportamiento real del curso coincide con el programado. Para ello, se deberá determinar en tiempo de diseño cuáles son los Factores Críticos de Éxito (*Critical Success Factors*), que reflejarán los objetivos

de alto nivel que se han marcado para el curso y a partir de los cuales se derivarán una serie de normas que serán incluidas en los metadatos del diseño. El sistema se encargará de recuperar información acerca del desarrollo del curso centrándose especialmente en la duración del uso de cada módulo de curso, así como el orden de empleo de los mismos. Esa información servirá como base para la generación de una serie de informes que los diseñadores utilizarán para comparar los resultados obtenidos por un determinado grupo de alumnos con los esperados.

2.6.2. LOCO-Cite

Sistema desarrollado por el *Laboratory for Ontological Research* (LORE) de la Universidad British Columbia de Canada cuyo objetivo es incrementar la reusabilidad de *learning designs* permitiendo su uso en varios contextos distintos. Siguiendo este objetivo han desarrollado tres ontologías distintas:

- *ALOCoM Content Structure* (ALOCoMCS): Permite la representación formal de la estructura de un *Learning Object*.
- LOCO: Ontología que comprende los conceptos y relaciones del modelo de información del *Learning Design*.
- LOCO-cite: Ontología empleada en la formalización de contexto de un *Learning Object*.

El sistema propone una arquitectura de repositorios de LOs, LDs y contextos de uso almacenados de manera acorde a las ontologías ALOCoMCS, LOCO y LOCO-cite respectivamente. Los LDs no referencian directamente LOs sino que contienen una consulta que especifica las características del LO que requieren. De esta forma, cuando se genera una instancia de un LD, el sistema ejecuta dichas consultas contra el repositorio de contextos devolviendo o recomendando los LOs más adecuados para el desarrollo de la actividad de aprendizaje y el usuario en cuestión. A su vez, cada uso de un LO genera una nueva instancia de contexto que se almacenará en el correspondiente repositorio y cuya información será empleada a la hora de generar posteriores recomendaciones.

2.6.3. Resumen del capítulo

En este capítulo se han analizado diversas investigaciones y trabajos desarrollados en el área de la enseñanza asistida por ordenador con objeto de conocer las distintas

posibilidades existentes a la hora de implementar un proceso de aprendizaje soportado computacionalmente y supervisado por instructor. Se comenzó estudiando el papel que el instructor desempeña en un proceso educativo, así como las tareas y funciones que lleva a cabo y su grado de importancia dependiendo del tipo de proceso a implementar. Igualmente se analizó la importancia que, de cara al éxito del proceso, tiene la posibilidad de adaptar la definición del mismo tanto a las características y necesidades de sus participantes como a las propias circunstancias de su ejecución. Una vez conocidas y analizadas las características más relevantes de los procesos de aprendizaje supervisados por instructor, se estudiaron dos enfoques distintos a la hora de implementar un sistema computacional que le dé soporte: el primero, seguido por los AIES, combina técnicas provenientes del campo de la inteligencia artificial y la hipermedia con el propósito de obtener sistemas capaces de generar procesos educativos personalizados para cada alumno. Las posibilidades de adaptación del proceso vienen dadas por la propia configuración del sistema y el contenido educativo a utilizar debe ser desarrollado teniendo en cuenta las mismas. El segundo, seguido por los LMS, se centra en el desarrollo de sistemas que proporcionen los servicios necesarios para posibilitar el desarrollo de experiencias educativas dentro de un entorno virtual. El sistema no implementa la lógica de adaptación, sino que se limita a poner a disposición del usuario los recursos necesarios en cada momento. El vínculo entre sistema y contenido es más débil que en el primer caso, por lo que resulta más sencillo conseguir la interoperabilidad del material entre distintas aplicaciones empleando estándares y especificaciones que establezcan acuerdos sobre su formato. Dada la importancia que la reutilización e intercambio de contenido educativo tiene de cara a amortizar el alto coste de desarrollo del material, se analizaron las propuestas de estandarización más importantes y de mayor aceptación en la actualidad, como LOM, SCORM o IMS Content Package entre otras. A continuación se analizaron las posibilidades que ofrecen los Lenguajes de Modelado Educativo (EML), que permiten llevar a cabo descripciones del propio proceso educativo, especificando cómo debe ser su desarrollo y qué uso debe hacerse de cada recurso, de tal forma que dichas descripciones puedan ser posteriormente procesadas e interpretadas por la aplicación correspondiente. De entre las distintas propuestas, la más completa y de uso más extendido en la actualidad es la del IMS LD, por lo que sus características fueron analizadas con más detenimiento, estudiando las posibilidades que ofrece para la especificación de procesos adaptativos. Como se observó, el lenguaje incluye elementos que permiten especificar distintos flujos de ejecución adaptando su

desarrollo bien a las acciones del propio usuario o bien a las solicitudes del instructor. En cualquier caso, dichas adaptaciones han de ser siempre especificadas previamente al comienzo del proceso no siendo posible introducir variaciones una vez éste ha tenido lugar. El capítulo finalizó con la revisión de dos tecnologías que de distinta forma se encuentran relacionadas con la problemática del presente documento: por una parte, aLFanet, un sistema que combina características de los dos enfoques previamente analizados y que usa técnicas de inteligencia artificial para dar soporte a los diferentes actores de procesos educativos especificados mediante IMS LD, y por otra, LOCO-cite, que emplea ontologías y repositorios de almacenamiento de material educativo con el fin de aumentar la reusabilidad de los diseños de procesos de aprendizaje.

Capítulo 3

Planteamiento del problema

La presente tesis doctoral se enmarca dentro del conjunto de trabajos relacionados con procesos de aprendizaje asistido por ordenador en los cuales es necesaria la figura de un tutor o instructor que supervise el proceso. Si bien, a día de hoy, en algunos tipos de procesos esta figura puede ser reemplazada por algún tipo de agente inteligente o algún otro mecanismo, en otros casos, tal y como se mencionó en el capítulo anterior, sigue siendo aún necesario y conveniente que el control y supervisión del proceso esté en manos de un tutor real.

Uno de los mayores problemas existentes a la hora de utilizar un soporte computacional para implementar un determinado proceso educativo es sin duda el alto coste de desarrollo o adaptación de los materiales didácticos al mismo, especialmente si queremos aprovechar en ellos las ventajas que el uso de la multimedia ofrece [51, 90]. Resulta innegable la utilidad que el uso de estándares en la creación de materiales educativos ofrece de cara a rentabilizar dichos costes de desarrollo, al garantizar la interoperabilidad de los contenidos en plataformas de distintos fabricantes. Así pues, a la hora de elegir un soporte sobre el que implementar un proceso de enseñanza supervisado por un instructor, parece natural decantarse por el empleo de entornos virtuales o LMSs donde se puedan aprovechar las ventajas derivadas de dicho uso de estándares, frente a los complejos Sistemas de Educación Adaptativos e Inteligente en Web (AIES) que no posibilitan la reutilización e intercambio de contenido educativo entre ellos [13]. Este hecho, unido a que en estos sistemas tanto el contenido como los escenarios de aprendizaje suelen ser diseñados a priori para dar soporte a un determinado enfoque pedagógico [48], hace pensar que, a pesar de las ventajas y posibilidades que los AIES ofrecen, su uso quede relegado a dominios de instrucción muy concretos.

Por otra parte, el empleo de lenguajes de modelado educativo para el diseño de

procesos de aprendizaje permite llevar a cabo una descripción detallada de la secuencia de actividades que los distintos participantes del proceso deberán seguir durante el desarrollo del mismo. Una vez construido el diseño, éste se puede emplear una y otra vez para repetir el mismo proceso de aprendizaje con distintos participantes. Dada la imposibilidad de prever de antemano cuáles serán sus reacciones durante el desarrollo del proceso, es necesario la utilización de lenguajes de modelado que permitan la creación de diseños de aprendizaje adaptativos, que posibiliten, tanto la aplicación en tiempo de ejecución de adaptaciones previamente definidas, como la introducción de nuevas modificaciones que alteren la definición original.

Igualmente, es necesario el desarrollo de modelos, métodos y herramientas que permitan a los instructores el control y seguimiento de la ejecución de dichos diseños adaptativos, de tal forma que puedan ser adecuados a las diferentes circunstancias que tengan lugar durante el desarrollo del proceso.

En este capítulo se analizan las principales carencias y problemas que pueden surgir al tratar de plasmar un proceso de aprendizaje supervisado por instructor mediante un lenguaje de modelado educativo y que han llevado a abordar el desarrollo de esta tesis. Este trabajo se ha llevado a cabo dentro del marco del EML más completo y aceptado en la actualidad: el *IMS Learning Design*.

3.1. Soporte insuficiente para el desarrollo de las funciones del tutor

La especificación del *IMS Learning Design* permite la distinción de varios roles entre los participantes dentro de un proceso de aprendizaje y ofrece mecanismos para implementar servicios tanto para permitir la comunicación entre los participantes del proceso como para permitir la recuperación de información acerca de su desarrollo. Estos últimos harán posible que, dependiendo del tipo de definición del objeto de monitorización que el diseñador lleve a cabo, sea posible que los distintos participantes puedan inspeccionar durante el desarrollo del proceso el valor que toman las propiedades de su propio dossier o las de otros participantes. En cualquier caso, el servicio proporcionado presenta algunas limitaciones, de tal forma que puede no satisfacer las necesidades reales de monitorización del instructor. Por una parte, únicamente se pueden inspeccionar los valores actuales de los elementos propiedad o *property*, no siendo posible acceder a información precisa acerca del estado de una actividad, tiempo que

el usuario ha pasado en ella, etc. Es posible preparar la UoL para que parte de esta información sea representada mediante dichas propiedades, de tal forma que su valor pueda ser posteriormente recuperado en ejecución, pero esto incrementa notablemente la complejidad del diseño. Por otra parte, las características a observar deben ser definidas de manera previa al comienzo de la ejecución, no siendo posible modificar la definición del servicio de monitorización si surge la necesidad de recuperar información acerca de aspectos o elementos del proceso que no fueron previstos inicialmente. Aunque se puede argumentar que la razón por la que IMS LD no implementa más servicios de monitorización y soporte al instructor es que no es un LMS, está claro que la utilidad de las UoLs desarrolladas mediante esta especificación dependerá en gran medida de la facilidad para su integración con componentes externos que permitan desarrollar estas acciones [9].

Por otro lado, la especificación IMS LD permite al diseñador establecer cuál es la secuencia de actividades que los participantes deberán desarrollar a lo largo del proceso. Esta definición del proceso, junto con los objetos de aprendizaje necesarios para el desarrollo de las actividades y las referencias a los recursos externos que pueden ser necesarios, se encapsula dentro de un fichero comprimido que se puede compartir o almacenar en un determinado repositorio. A la hora de ejecutar el proceso, el motor de ejecución apropiado se encargará de interpretar la información almacenada en el fichero y presentar las diversas actividades y materiales de la forma precisa especificada por el diseñador. Este enfoque presenta algunas limitaciones y, si bien se adecúa a procesos de aprendizaje que no implementan la figura del instructor, no contempla la posibilidad de que, en aquellos procesos que sean supervisados, el instructor necesite introducir modificaciones en la definición original del proceso para responder a circunstancias no previstas inicialmente. Exceptuando la asignación de valores a elementos propiedad o *property*, que se puede conseguir mediante *global elements* cuando dichos cambios hayan sido previstos en tiempo de diseño, el resto de modificaciones requerirá de la interrupción del proceso para su introducción, no siendo posible recuperar el estado original del mismo una vez que la ejecución sea reanudada. Así pues, existe un conflicto entre la naturaleza prescriptiva del IMS LD y la naturaleza flexible y dinámica del *eLearning* [65].

Teniendo en cuenta estos problemas se puede afirmar que **[P1]** *las especificaciones de EMLs no proporcionan al instructor las herramientas suficientes para poder llevar a cabo las mismas funciones que desarrollaría en un proceso no soportado por un entorno*

computacional. Concretamente, [P1.1] no proporcionan suficiente soporte para la monitorización del estado del proceso, [P1.2] ni proporcionan suficiente soporte para la actuación sobre su estado.

3.2. Lógica de adaptación encapsulada dentro de la UoL

Tal y como se ha comentado en la sección anterior, una vez que se dispone de una UoL puede ser necesario introducir modificaciones sobre su definición tanto de manera previa al comienzo de la ejecución como durante el desarrollo de la misma, con el fin de poder adaptar el diseño del proceso de aprendizaje al perfil del usuario, contexto actual de ejecución, entorno computacional, etc. Van Rosmalen [83] distingue dos formas distintas de implementar la introducción de adaptaciones en la definición de un proceso de aprendizaje que se corresponden con dos enfoques distintos a la hora de planear el desarrollo del mismo: incluir la lógica de adaptación dentro del diseño del proceso de aprendizaje o mantenerla externa al mismo.

El desarrollo de UoLs empleando la especificación del *IMS Learning Design* únicamente permite implementar el primer enfoque ya que la lógica de la adaptación se debe encontrar siempre dentro de la propia UoL. Esto presenta serios inconvenientes ya que el diseñador debe ser capaz de prever todas las posibles variaciones que el proceso pueda sufrir en ejecución y proporcionar una respuesta predefinida a los posibles eventos que durante el desarrollo del mismo se produjeran. Por otra parte, no se corresponde siempre con la manera de actuar de algunos instructores, que en ocasiones prefieren partir de un diseño preliminar abierto en el cual ciertas partes se concretan durante la propia ejecución del proceso [9]. De esta forma es posible responder a los intereses concretos que vayan surgiendo durante la ejecución, proporcionando ejemplos cercanos a los participantes o haciendo especial hincapié en unos u otros aspectos. Este enfoque resulta sencillo de implementar en un LMS como Moodle, donde el instructor puede manipular en todo momento la definición del proceso incluyendo nuevas secciones, evaluaciones o recursos.

A menudo, en la práctica, la solución más apropiada combina características de ambos enfoques. Por una parte, resulta útil disponer de un diseño previo, más o menos exhaustivo, que capture de antemano y en la medida de lo posible el mayor número de escenarios de ejecución posibles. Por otra parte, conviene que dicho diseño sea lo

suficientemente flexible como para que el tutor pueda modificarlo a lo largo de su desarrollo. De esta forma, debe ser posible manipular tanto la definición original del proceso, permitiendo así dar respuesta a los intereses y necesidades que puedan surgir durante su desarrollo, como su comportamiento en ejecución, permitiendo insistir sobre conceptos previamente expuestos u obviar actividades para las que los participantes han demostrado competencias. Esta manera de actuar se asemeja más a la forma en que los instructores llevan a cabo su labor en la enseñanza tradicional: tomando como punto de partida un diseño previo del proceso, el instructor observa la evolución de los distintos participantes del mismo y determina si el siguiente paso a efectuar se adecúa a las características actuales del entorno de ejecución o bien es necesario introducir alguna variación en el plan original. El control de la ejecución del proceso educativo suele estar principalmente en manos del instructor, siendo el diseño del proceso una herramienta más a su disposición.

Mantener separados el contenido o información educativa propiamente dicha de la información sobre la adaptación simplifica tanto el diseño de la UoL como su mantenimiento, de tal manera que se desliguen las modificaciones llevadas a cabo sobre un tipo de información de las modificaciones llevadas a cabo sobre el otro. Por otra parte, permite variar el nivel de granularidad y compleción de un diseño desarrollado específicamente para un contexto determinado, de tal manera que pueda ser refinado y ensamblado con distintos contenidos incrementándose así sus posibilidades de reutilización y amortizándose los costes de desarrollo. Este enfoque de autoría de UoLs ya ha sido analizado por diversos autores [22]. En cualquier caso, e independientemente de las ventajas de este enfoque, es necesario que el proceso de modificación garantice la consistencia de la UoL resultante tras la adaptación con la especificación del EML utilizado para su descripción, y sea suficientemente flexible para acomodarlo de forma sencilla a posibles cambios en futuras versiones de la misma.

Para poder aunar las ventajas que la especificación de procesos educativos mediante EML ofrece, y las derivadas de la creación de diseños no cerrados, **[P2]** *es necesario proveer al diseñador con los mecanismos necesarios para que pueda introducir de manera sencilla y rápida variaciones tanto sobre el diseño original como sobre su comportamiento en ejecución.*

3.3. Falta de un método que organice la introducción y evaluación de las adaptaciones

Es posible que a lo largo del desarrollo del proceso de aprendizaje se deban llevar a cabo modificaciones tanto sobre la planificación original del mismo como sobre los recursos educativos empleados. La necesidad de introducir este tipo de modificaciones es común tanto en los procesos soportados computacionalmente como en los no soportados, y en ambos tipos de procesos suelen ser llevadas a cabo por el instructor a cargo de la supervisión del mismo y responsable en último término de su éxito. La habilidad de rediseñar inconscientemente el proceso es una práctica tan inherente y útil que está implícita dentro de la propia rutina de la enseñanza [9]. Estas modificaciones o adaptaciones de rediseño se suelen introducir normalmente “sobre la marcha”, en respuesta a diversas situaciones que tienen lugar y que no se pudieron prever inicialmente, de tal forma que, dependiendo de la cantidad de adaptaciones aplicadas y de la severidad de los cambios introducidos, el diseño instructivo finalmente aplicado puede diferir en distinto grado del diseño original. Sin embargo, en la práctica, debido al modo en que estas adaptaciones se introducen, a menudo no queda registrada la modificación llevada a cabo ni la causa o propósito perseguido con ella.

Por otra parte, en estos casos, resulta necesario llevar a cabo un análisis posterior que determine y evalúe en qué medida el diseño finalmente aplicado ha supuesto realmente una mejora respecto al diseño inicialmente planteado. Es decir, es necesario evaluar tanto el éxito del diseño aplicado, como el efecto y grado de satisfacción del objetivo perseguido con cada una de las adaptaciones introducidas, de tal forma que sea posible identificar e integrar dentro de la definición del proceso aquellas modificaciones que supongan algún tipo de mejora sobre la definición original del proceso. En cualquier caso, si resulta poco habitual registrar el propósito del cambio, aún lo es menos definir de antemano la fórmula o mecanismo que debería utilizarse para evaluar su éxito. Así pues, resulta difícil de acometer con éxito la tarea de análisis previamente descrita.

[P3] *Es por tanto necesario el establecimiento de un método que organice la introducción y evaluación del éxito de las adaptaciones aplicadas y facilite al instructor la tarea de evaluación del éxito del proceso.*

3.4. Relación entre problemáticas y objetivos de la tesis

La tabla 3.1 muestra la relación entre las problemáticas descritas y los objetivos de la tesis planteados en el primer capítulo.

Tabla 3.1: Relación entre problemáticas y objetivos de la tesis

Objetivo	Problemática	Casos
01. Modelo		
(O1.1) Adaptación definición proceso y comportamiento del proceso	2	Imposibilidad de adaptar la definición del proceso en ejecución. Imposibilidad de variar el comportamiento pre-definido para el proceso.
(O1.2) Especificar momento de introducción de la adaptación	1.1	Imposibilidad de definir y aplicar adaptaciones durante la ejecución.
(O1.3) Soporte monitorización proceso	1.2	Insuficiente capacidad de los servicios de monitorización.
(O1.4) Facilitar análisis éxito adaptación	1.2	Imposibilidad de definir nuevas monitorizaciones en ejecución.
(O1.5) Facilitar análisis éxito proceso	3	Dificultad evaluación éxito adaptación. Dificultad evaluación éxito proceso.
02. Método		
(O2.1) Ordenar el desarrollo de las tareas del instructor soportadas mediante el modelo de adaptación	3	Falta de un método que ordene el desarrollo de las actividades de monitorización, adaptación, evaluación e integración de cambios.
(O2.2) Recoger de forma explícita el cambio introducido	3	A menudo no suele registrarse el cambio efectuado ni su propósito.
(O2.3) Realizar las actividades de adaptación y monitorización del proceso	1.1	Imposibilidad de definir y aplicar adaptaciones durante la ejecución.
(O2.4) Realizar las actividades de evaluación de cambios introducidos y éxito del proceso	1.2	Imposibilidad de definir nuevas monitorizaciones en ejecución.
	3	Dificultad evaluación éxito proceso.
03. Mecanismo implementación		
(O3.1) No intrusivo	2	Imposibilidad de separar contenido y lógica adaptación.
(O3.2) Consistente	2	Necesidad de garantizar la consistencia de la ToL modificada con la especificación del EML empleado en su descripción.
(O3.2) Flexible	2	Necesidad de flexibilidad para acomodar el mecanismo a posibles cambios en futuras versiones del EML.

Capítulo 4

Adaptación, observación y evaluación de procesos de aprendizaje especificados mediante EMLs

El objetivo del presente capítulo es ofrecer una solución a las distintas problemáticas expuestas en el capítulo anterior. Para ello, **se desarrollará un modelo de adaptación que permita extender lenguajes de modelado educativo con los elementos necesarios para dar soporte al instructor en el desempeño de sus tareas a lo largo del proceso de aprendizaje y salvar las limitaciones actuales previamente descritas.** Esta extensión será definida mediante *un modelo conceptual* que describirá la finalidad y relaciones de sus principales elementos empleando lenguaje natural y diagramas de clases UML, y *un modelo de información*, que concretará la composición y atributos de cada elemento mediante tablas de información y diagramas jerárquicos, de tal forma que se facilite su posterior traducción y representación mediante lenguajes apropiados para su implementación.

Asimismo, será necesario **definir un método que organice tanto la introducción de adaptaciones durante el desarrollo del proceso de aprendizaje, como la ejecución del resto de tareas soportadas mediante los elementos del modelo de adaptación previamente definido**, es decir, la monitorización del desarrollo del proceso y la evaluación del éxito de las adaptaciones introducidas y satisfacción de los objetivos marcados para el proceso.

Por último, será necesario el desarrollo de tres mecanismos distintos que permitan la implementación de los elementos del modelo:

- Por una parte, será necesario desarrollar un **mecanismo que permita definir y aplicar adaptaciones tanto sobre diseños de procesos de aprendizaje**

descritos mediante un determinado EML, como sobre su comportamiento en ejecución, y que permita mantener separada la lógica de la adaptación del contenido original de la UoL.

- Por otra parte, para que el instructor pueda detectar la necesidad de introducir la adaptación, será necesario proveer los medios para que pueda evaluar el desarrollo del proceso y medir el éxito y progreso de los participantes.
- Por último, y con objeto de poder implementar el mecanismo de evaluación, es necesario que el instructor disponga de un mecanismo de observación que permita monitorizar las acciones de los participantes así como el estado actual de los recursos empleados.

Estos tres mecanismos se describen en las tres primeras secciones del capítulo y se encuentran estrechamente relacionados entre sí, de tal forma que un sistema que implemente el mecanismo de adaptación puede ser fácilmente extendido para que soporte también el mecanismo de observación y a la inversa. Igualmente, la solución descrita para llevar a cabo la evaluación del éxito del proceso se basa en el empleo de observaciones para la recuperación de la información necesaria.

Una vez los mecanismos propuestos sean definidos se pasará a especificar las distintas partes del modelo de adaptación propuesto, y a continuación se definirá un método para el diseño iterativo de procesos de aprendizaje. El capítulo finalizará con la definición de una arquitectura que permita incorporar la capacidad de interpretar y aplicar los elementos del modelo a motores de ejecución ya existentes.

4.1. Adaptaciones del proceso de aprendizaje

Basándose en la información recuperada observando el desarrollo del proceso, los instructores pueden definir las variaciones que necesitan introducir para garantizar el éxito del mismo. Jacobson [46] define punto de variación como *“lugares en el diseño o implementación que identifican posiciones en las cuales puede tener lugar una variación”*. Los puntos de variación pueden vincularse al sistema en diferentes estadios del ciclo de vida del producto. Svanhberg [59] presenta una taxonomía de técnicas de introducción de variabilidad que definen diferentes maneras mediante las cuales se pueden implementar puntos de variación. Una de estas técnicas es la superposición de fragmentos de código, la cual consiste en desarrollar una solución software que permite

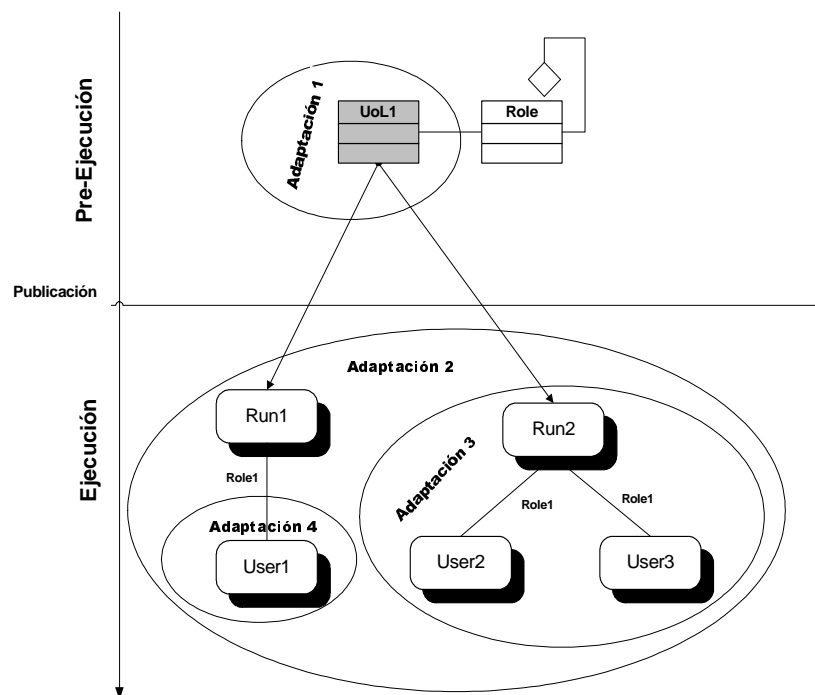


Figura 4.1: Adaptaciones de un proceso de aprendizaje

solucionar un problema genérico, y sobre la cual se superponen fragmentos de códigos que resuelven problemáticas específicas. Esta superposición se alcanza por medio de diferentes técnicas como por ejemplo la Programación Orientada a Aspectos (AOP, *Aspect Oriented Programming* [49]), permitiendo al diseñador introducir la modificación tanto en la fase de compilación como durante la propia ejecución de la aplicación.

Estos conceptos pueden ser directamente trasladados al área de la adaptación de procesos de aprendizaje: los autores pueden mantener por una lado una única definición de la UoL y por otro una serie de descripciones de adaptaciones, cada una de las cuales encierra todos los cambios asociados con una particular variación. Posteriormente, todas las descripciones se procesarán conjuntamente, dando a los participantes del proceso la impresión de que los cambios formaban parte de la definición original de la UoL. De esta forma se mantiene separada la funcionalidad principal de la UoL de la del resto de las adaptaciones.

La figura 4.1 describe una visión general de este proceso. De entre un conjunto de posibles adaptaciones definidas para una UoL particular, el diseñador elige aquella que mejor encaja para la situación actual y la aplica. Esta adaptación puede ser llevada a cabo tanto de manera previa a la ejecución — adaptación 1 — como durante el desarrollo de la misma — adaptaciones 2, 3 y 4 —. En el último caso la adaptación

puede aplicarse a todas las instancias en ejecución de la UoL — adaptación 2 —, a todos los usuarios de una instancia particular — adaptación 3 — o bien únicamente a la visión personal de un único usuario — adaptación 4 —.

Las adaptaciones introducidas pueden incluir descripciones del objetivo perseguido con la modificación de tal manera que el éxito de la adaptación se pueda evaluar posteriormente. También se puede tratar de medir la influencia de la adaptación en la consecución de los objetivos de aprendizaje de tal manera que aquellas adaptaciones que hayan probado suponer una mejora para el desarrollo del proceso se puedan incluir en la definición del proceso de manera definitiva.

4.1.1. Definición de *Adaptation poke*

Llamamos *adaptation pokes* a las descripciones de pequeñas modificaciones sobre algunos elementos de un proceso de aprendizaje. La definición de un *adaptation poke* implica:

- Descripción de la acción de adaptación: Definir el conjunto de cambios que se van a aplicar secuencialmente a la definición original de la UoL. Es la única parte obligatoria en la definición de un *adaptation poke*.
- Descripción de nuevos elementos: Cuando la adaptación implica la introducción de nuevos elementos en el diseño del proceso de aprendizaje será necesario proveer la definición de dichos elementos.
- Nuevos recursos: La adaptación puede requerir la introducción de nuevos contenidos o la sustitución de los existentes.

Una adaptación estará constituida por la aplicación de uno o más *adaptation pokes*. Un *adaptation poke* está a su vez constituido por una serie de acciones adaptativas o modificaciones que tras ser procesadas en su totalidad dejan la UoL en un estado consistente, es decir, que no infringe ninguna de las restricciones de la especificación empleada para la definición de su manifiesto.

Los *adaptation pokes* pueden ser aplicados a la UoL en tiempo de publicación si son incluidos dentro del propio *content package*. El motor de ejecución se encargará de procesar su información junto con la información original del paquete, aplicando los cambios antes de presentar el contenido a los participantes del proceso. Los *pokes*

deben ser numerados, de tal manera que si hay más de uno incluido en el paquete sean aplicados en orden secuencial.

Para llevar a cabo la adaptación de una UoL en ejecución, los *adaptation poke* deberán ser publicados indicando la UoL, ejecución y usuarios para los cuales se debe aplicar la modificación.

4.1.2. Tipos de adaptaciones

En principio, todos los elementos descritos en el diseño de un proceso de aprendizaje son susceptibles de ser adaptados. Sin embargo, el mecanismo de adaptación propuesto en este documento únicamente tiene sentido cuando implica pequeñas variaciones en los diseños. Cuando el objetivo es llevar a cabo modificaciones importantes en la estructura del proceso, como por ejemplo la inclusión de nuevos roles, la estrategia pedagógica implementada puede sufrir serias modificaciones por lo que resulta más conveniente llevar a cabo un re-diseño completo del proceso en el que se estudien y analicen los efectos del cambio.

Podemos distinguir dos tipos de adaptaciones según el objeto de la modificación:

- Adaptaciones estáticas o estructurales: son aquellas adaptaciones que modifican la definición del proceso de tal forma que añaden o eliminan elementos a su estructura o bien modifican la definición de los ya existentes. Ejemplos de adaptaciones de este tipo serían la introducción de material auxiliar, la supresión de ciertas partes del proceso, o la sustitución de un recurso por otro.
- Adaptaciones dinámicas o de estado: son aquellas adaptaciones que modifican el estado de los elementos del proceso en un momento particular de una ejecución. Como ejemplos de este tipo se considerarían los cambio de estado de una actividad de *incompleta* a *completa*, la modificación de la visibilidad de un recurso, o la asignación de un valor a una variable.

4.1.3. Clasificación de los *adaptation pokes*

Dependiendo del propósito de la adaptación, momento de introducción, alteración de los objetivos de aprendizaje originales y posibilidad de integración en la definición del proceso podemos clasificar los *adaptation pokes* en cuatro grupos distintos: *pokes de retoque*, *pokes de reajuste*, *pokes correctivos* y *pokes de variación*.

Los *pokes de retoque* (tabla 4.1) describen modificaciones puntuales sobre el proceso de aprendizaje que no deben ser tenidas en cuenta en el resto del proceso. Así por ejemplo, puede ser necesario modificar el límite de tiempo de una evaluación debido a la no disponibilidad momentánea de un recurso, o forzar un cambio de estado de una actividad a “completada” por diversas causas. Por su naturaleza, el diseño de un *poke* de este tipo sólo tiene sentido en ejecución y su definición no incluirá objetivos evaluables ya que su introducción es consecuencia de una situación específica y puntual. Por el mismo motivo, no se deberá considerar la integración de manera permanente de este tipo de *pokes* en la definición del proceso, ni su introducción será tenida en cuenta a la hora de evaluar la satisfacción de los objetivos del mismo.

Tipo de poke	Poke de retoque
Momento de introducción	Ejecución
Objetivos del poke	No
Influencia objetivos UoL	No
Modificación objetivos UoL	No
Posibilidad de Integración	No

Tabla 4.1: Características de los pokes de retoque

Tipo de poke	Poke de reajuste
Momento de introducción	Pre-Ejecución y en ejecución
Objetivos del poke	No
Influencia objetivos UoL	Posible
Modificación objetivos UoL	Posible
Posibilidad de Integración	No

Tabla 4.2: Características de los pokes de reajuste

El propósito de un *poke de reajuste* (tabla 4.2) es introducir una rectificación en el proceso de aprendizaje después de que una incidencia de naturaleza externa el proceso tenga lugar. Por ejemplo, cuando debido a condiciones climatológicas adversas se debe modificar el plan de un curso y eliminar algunas actividades y objetivos de aprendizaje. Al igual que en el caso anterior, la acción de este tipo de *poke* no será evaluable ni se considerará su integración puesto que su introducción responde a causas de origen externo. Su definición puede requerirse durante la ejecución del proceso o ser prevista de manera previa al comienzo de éste. Dependiendo de la gravedad del cambio podrán influir en la consecución de los objetivos marcados para el proceso o incluso modificar

Tabla 4.3: Características de los pokes correctivos

Tipo de poke	Poke correctivo
Momento de introducción	Ejecución
Objetivos del poke	Si
Influencia objetivos UoL	Posible
Modificación objetivos UoL	Posible
Posibilidad de Integración	Si

Tabla 4.4: Características de los pokes de variación

Tipo de poke	Poke de variación
Momento de introducción	Pre-Ejecución
Objetivos del poke	No
Influencia objetivos UoL	Posible
Modificación objetivos UoL	Posible
Posibilidad de Integración	No

la propia definición de los mismos.

Por otro lado, un *poke correctivo* (tabla 4.3) define un conjunto de acciones necesarias para la corrección de un determinado problema que surge durante la ejecución del proceso y es de naturaleza interna al mismo. Así por ejemplo, el instructor puede decidir introducir diverso material complementario con el fin de aclarar ciertos errores de comprensión, o necesitar modificar el tiempo límite asignado al detectar que la estimación inicial asignada no es adecuada para el actual perfil del participante. La introducción del *poke*, por tanto, tiene un objetivo definido, y el grado de satisfacción del mismo deberá ser evaluado para estimar la utilidad de la adaptación. Podrán ser integrados en la definición del proceso y su introducción deberá considerarse a la hora de evaluar los objetivos del mismo.

Por último, los *pokes de variación* (tabla 4.4) definen una transformación del proceso con el objeto de adecuarlo a un contexto o propósito distinto al original. A partir de su definición original se pueden generar versiones de un determinado proceso donde se supriman, por ejemplo, las actividades de carácter práctico o donde los recursos Web sean sustituidos por otros de tal forma que el curso pueda ser seguido sin conexión a Internet. La acción de este tipo de *pokes* no es por tanto evaluable, por lo que su definición tampoco deberá incluir objetivos. Además, los objetivos del proceso resultante

de su introducción pueden diferir de los originales.

Un *poke* correctivo puede convertirse en un *poke* de variación en las siguientes ejecuciones del curso. Por ejemplo, considérese el caso de una instrucción en la que, debido a un error en la asignación del perfil del alumnado, ha sido necesaria la introducción de una serie de *pokes* correctivos que, a lo largo de la ejecución del proceso, han ido subsanando las lagunas de conocimiento que el tutor detectó en los alumnos y que impedían el desarrollo de las actividades propuestas. Una vez finalizado el proceso y analizados los resultados del mismo, los diseñadores pueden optar por agrupar los *pokes* introducidos bajo un sólo *poke* de variación que servirá para directamente adaptar la UoL al nuevo perfil de alumno.

4.1.4. Validación de la UoL adaptada

Cada vez que una UoL es publicada en un determinado motor de ejecución, se dispara un proceso de validación para garantizar su conformidad con la especificación correspondiente y la disponibilidad de los recursos en ella referenciados. De igual manera, cada vez que la definición de una UoL es modificada por medio de un proceso de adaptación, se debe repetir el mismo proceso de validación para asegurar que su definición sigue siendo válida y es consistente con la especificación del EML empleado.

4.1.4.1. Conflictos y tipos de inconsistencias

Una adaptación particular está compuesta por la aplicación de un determinado número de *adaptation pokes*. Un *adaptation poke* está compuesto a su vez por una serie de acciones adaptativas descritas mediante un conjunto de comandos. En ambos casos su aplicación debe ser considerada como una transacción completa, resultando por tanto en su completa aplicación o en la invalidación de la modificación. Teniendo en cuenta esto, debemos considerar dos tipos distintos de problemas de integridad:

- Inconsistencia introducida por un único *adaptation poke*: la modificación introducida por un *adaptation poke* puede resultar en una UoL inconsistente.
- Conflictos entre distintos *adaptation pokes*: para garantizar el éxito en la aplicación de un determinado *adaptation poke*, se deben tener en cuenta posibles modificaciones llevadas a cabo sobre la UoL con anterioridad. Así, la definición original de la UoL puede haber sido modificada por otras adaptaciones, eliminando o alterando ciertas actividades, entornos o recursos, y provocando de esta

forma el fallo de un *adaptation poke* diseñado según la definición original de la UoL.

El segundo tipo de problemas puede solucionarse mediante programación: bastaría con mantener una copia del estado y definición de la UoL antes de la modificación que pudiera ser restaurada en caso de que se produzca algún conflicto. Sin embargo, los problemas de primer tipo son más difíciles de remediar debido a que la aplicación del *poke* no concluye con su fallo. Dentro de este grupo se pueden distinguir cuatro tipos de inconsistencias:

- Inconsistencias relacionadas con el progreso de los participantes en el proceso de aprendizaje: al aplicar el mismo *poke* a diferentes sesiones de una misma UoL se debe tener en cuenta que diferentes participantes del proceso pueden encontrarse en diferentes estados del mismo. Así, en IMS LD por ejemplo, acciones sobre las definiciones *complete-activity* o *feedback* de las actividades pueden tener distintas consecuencias para los distintos participantes según el estado de dichas actividades para sus respectivas instancias sea *completadas* o no.
- Inconsistencias relacionadas con los contenidos y objetivos de aprendizaje: Cambios en los contenidos de la UoL, como por ejemplo recursos o actividades, pueden provocar problemas de comprensión en el alumno a la hora de seguir otras partes del curso. Por un lado, el desarrollo de ciertas actividades es fundamental para poder abarcar con posterioridad otras o la información necesaria para poder desarrollar correctamente cierto ejercicio de evaluación puede aparecer únicamente en un recurso particular. En estos casos, la eliminación de estos elementos puede provocar la aparición de inconsistencias en el curso. Por otro lado, los objetivos de aprendizaje originales deben ser tenidos en cuenta durante el proceso de adaptación o de lo contrario la UoL adaptada puede no corresponderse con ellos.
- Inconsistencias relacionadas con la definición del proceso: cada acción del *poke* debe ser validada para garantizar que la definición de la UoL modificada se mantiene conforme con la especificación del EML correspondiente.
- Inconsistencias relacionadas con la ejecución del proceso: las acciones de los *pokes* debe ser validadas para garantizar que el estado de las instancias de las UoLs modificadas no vulnera ninguna de las restricciones de integridad definidas para

el comportamiento en ejecución de la UoL por la especificación del EML correspondiente. Así, por ejemplo, la especificación del IMS LD determina que si se definen varios *actos* en una misma UoL, estos deben ser ejecutados de manera secuencial.

Para prevenir la introducción de inconsistencias del primer tipo, se pueden implementar diferentes servicios de monitorización del proceso (ej: *monitor service* [41]), de tal forma que el diseñador pueda obtener información precisa acerca del estado del proceso para cada participante y adoptar, de esa forma, la acción más apropiada para cada caso.

La prevención de introducción de inconsistencias de segundo tipo implica conocer de antemano qué contenidos afectan a qué objetivos y de qué manera. Para ayudar al diseñador a prevenirlas se podría hacer uso de una ontología educativa [68] que capturase las relaciones entre las distintas partes, contenidos y objetivos del curso y que fuese empleada para validar la corrección del curso modificado. En cualquier caso, lo habitual entre las plataformas educativas es que la responsabilidad de evitar la introducción de este tipo de inconsistencias descansa en el instructor. Este enfoque resulta menos costoso que implementar todo un proceso de validación y formalización de las relaciones entre las partes del curso, y suele resultar adecuado ya que, por definición, el instructor suele ser tanto un experto en el dominio como un profundo conocedor del diseño del proceso.

En cuanto a las inconsistencias de tercer y cuarto tipo, pueden ser resueltas de manera automática codificando las características y propiedades que un diseño de aprendizaje válido para una determinada especificación debe cumplir. En cualquier caso, esta es una tarea costosa y futuros cambios o extensiones en la especificación del EML implicarían modificaciones en el código. Un enfoque más efectivo para validar la consistencia de la UoL modificada consideraría llevar a cabo dicha validación empleando de nuevo una ontología, pero que en esta ocasión capturase la semántica de la especificación utilizada en la descripción del proceso. Este es el enfoque adoptado en esta tesis en su aplicación a IMS LD.

4.2. Recuperación de información sobre la ejecución del proceso de aprendizaje

Para poder ser capaces de introducir las adaptaciones apropiadas es necesario que los instructores dispongan de los medios adecuados para monitorizar las acciones de los distintos participantes del proceso de aprendizaje a lo largo del desarrollo del mismo. Al igual que ocurre con las adaptaciones, existen dos tipos distintos de observaciones que se pueden efectuar sobre un proceso de aprendizaje:

- Observaciones estáticas o estructurales: recuperan información acerca de la definición de los componentes del proceso, es decir, información acerca de los atributos definidos para cada elemento en la definición del manifiesto, como por ejemplo el número de actividades simples que es necesario completar para considerar una actividad compuesta como terminada, o la lista de identificadores de las actividades que la componen. Este tipo de observaciones son de especial utilidad cuando se trata con manifiestos complejos, con un gran número de elementos, o cuando se han aplicado múltiples *adaptation pokes* a una UoL y el instructor quiere conocer cuál es el valor actual de cierto atributo.
- Observaciones dinámicas o de estado: permiten consultar el valor actual de las características dinámicas de los elementos del proceso, como por ejemplo, saber si una actividad determinada ha sido ya completada por un cierto usuario.

Combinando ambos tipos de observaciones es posible que los instructores puedan conocer, por ejemplo, cuáles son las actividades que faltan por llevar a cabo para completar cierta sección del curso.

A continuación se describe un mecanismo que permite a un instructor ser capaz de definir qué elementos y características desea observar, así como el momento en que dicha observación se debe llevar a cabo.

4.2.1. Progress Watcher

Llamamos *Watcher Peeks* a pequeñas descripciones que especifican los componentes y características cuyo valor desean observar los instructores. Los *Watcher Peeks* serán leídos y procesados por un *Progress Watcher*, cuya función consistirá en interpretar las descripciones en ellos contenidas y recuperar la información correspondiente a partir

de una determinada instancia de una UoL. Los diseñadores pueden considerar el implementar un servicio *ProgressWatcher* como alternativa o complemento a los servicios de monitorización que proporcione la especificación del EML utilizado para la definición del diseño.

Los *WatcherPeeks* se pueden introducir en ejecución para obtener información acerca de la instancia de un usuario particular, todos los usuarios de una ejecución de una UoL, o bien de todas las instancias en ejecución de una UoL, y pueden contener tanto definiciones de observaciones estáticas como definiciones de observaciones dinámicas.

Las acciones de monitorización descritas en los *WatcherPeeks* pueden ser aplicadas sobre el proceso de diversa forma:

1. Monitorización puntual en un instante de tiempo: el instructor recupera el valor de la característica a observar en el momento de introducir el *Watcher Peek*.
2. Monitorización continuada: se recupera el valor que la característica o propiedad va a adoptando desde el momento en que se introdujo el *Watcher Peek*, de tal forma que el instructor pueda conocer su evolución a lo largo de la ejecución.
3. Condición de activación: el momento en el cual se debe llevar a cabo la lectura de la característica es definido mediante una determinada condición. Así, por ejemplo, puede ser de utilidad para el instructor conocer en qué instante de tiempo ha completado un determinado alumno una determinada actividad, o cuál era la puntuación que llevaba acumulada al acceder a cierto elemento. Esto puede conseguirse utilizando los valores recuperados de unas lecturas en las condiciones de activación de otras.
4. Condición de activación relativa a otra instancia del proceso: en ocasiones puede ser interesante para el instructor poder relacionar las observaciones sobre los elementos del proceso de un determinado participante con las de otro participante. Así, puede necesitar conocer en qué punto del proceso se encontraba cierto alumno cuando otro daba por finalizada cierta actividad.

4.3. Evaluación del éxito del proceso de aprendizaje

Para poder determinar el éxito del diseño del proceso de aprendizaje es necesario especificar de antemano cuáles son las consecuencias que se esperan obtener de la ejecución del mismo, es decir, especificar cuáles son los objetivos de aprendizaje marcados

para los alumnos y, una vez concluido el proceso, medir el grado de consecución de los mismos.

En [61] el autor define *objetivo de aprendizaje* como la descripción de una capacidad que se desea que los alumnos sean capaces de exhibir una vez terminado el proceso, haciendo hincapié en que un objetivo describe un resultado de la instrucción más que el proceso de la instrucción en sí. El modelo de Mager define tres partes distintas en la definición de un objetivo: (1) debe tener un verbo medible (un verbo de acción), (2) debe incluir una especificación de con qué se provee al alumno para que sea capaz de llevar a cabo la acción, (3) debe incluir una especificación del criterio utilizado para definir si la acción es llevada a cabo de manera exitosa o con suficiente habilidad.

Es responsabilidad del diseñador del proceso de aprendizaje tanto la correcta definición de los objetivos que se pretenden alcanzar, como el incluir dentro del propio diseño los medios necesarios para hacer posible la correcta medición del éxito en la consecución de los mismos. Sin embargo, se puede facilitar al instructor dicha tarea de medición si se provee al diseñador de una notación que permita la descripción del criterio de éxito de cada uno de los objetivos de una manera que sea automáticamente procesable. En la sección anterior se describía un mecanismo que permite la recuperación de información acerca del desarrollo del proceso y que se puede emplear para obtener información acerca del desarrollo de las diferentes actividades, ejercicios prácticos o test incluidos en el diseño con el objeto de posibilitar la evaluación de la satisfacción de un determinado objetivo. Así, la información recuperada se puede combinar entre sí mediante fórmulas previamente especificadas para cada objetivo de aprendizaje definido, y el valor resultante comparado con ciertos umbrales que determinen el nivel de éxito en su consecución.

Hay que tener en cuenta que el criterio de evaluación puede variar en función de ciertas características del perfil del alumno. Así por ejemplo, pueden establecerse distintos umbrales que determinen el nivel de éxito para un objetivo según cuál sea la edad o formación del alumno en cuestión, o emplear calificaciones obtenidas en procesos anteriores para establecer el grado de mejora que el resultado actual representa en relación al conocimiento previo del alumno. Es por tanto necesario, a la hora de especificar las fórmulas de evaluación de los objetivos de aprendizaje, poder combinar información relativa al desarrollo del proceso con información de carácter externo al mismo y que puede encontrarse almacenada en dispositivos distintos.

También debe tenerse en cuenta que, en ocasiones, ciertos datos acerca del desarrollo

del proceso o la actuación de un determinado alumno resultan difícilmente recuperables únicamente a partir de la información registrada por el sistema, y que resulta más rápido y adecuado solicitar al instructor la introducción de una estimación del mismo o la confirmación del dato previamente obtenido.

Por tanto, para facilitar la evaluación del éxito del diseño del proceso de aprendizaje, se proporcionará al diseñador una notación que permita la especificación de fórmulas en las cuales sea posible combinar información recuperada acerca del desarrollo del proceso con otras informaciones de carácter externo al mismo o con estimaciones introducidas directamente por el tutor, de tal forma que el valor resultante constituya una valoración del grado de éxito del alumno en la consecución de un objetivo de aprendizaje determinado. A partir de los valores obtenidos para cada alumno en el conjunto de objetivos de aprendizaje marcados para el proceso, el instructor será capaz de determinar el éxito o fracaso de la experiencia educativa.

4.4. Modelo conceptual

En esta sección se describirán los distintos elementos que componen el modelo conceptual propuesto para dar soporte a la adaptación, observación y evaluación de un proceso de aprendizaje, así como las distintas relaciones existentes entre ellos tanto en cuanto a su definición como en cuanto a su comportamiento en ejecución.

4.4.1. Modelo conceptual estático

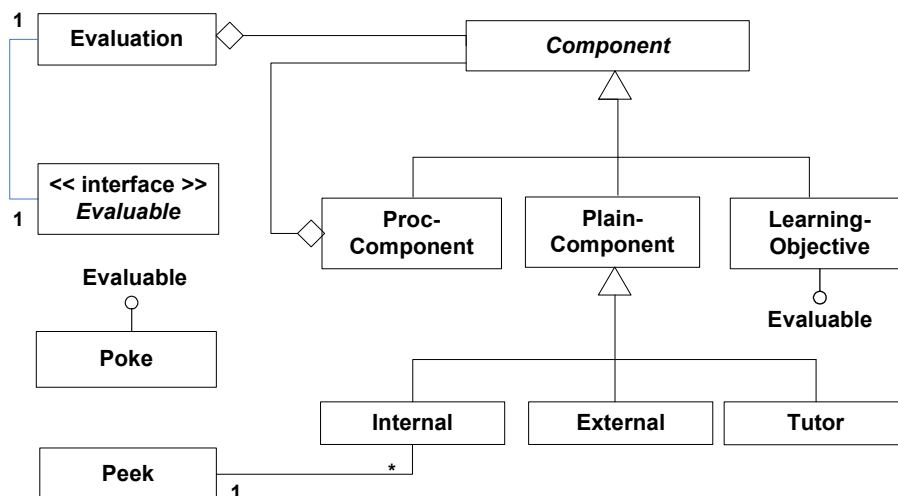


Figura 4.2: Modelo conceptual estático

La figura 4.2 muestra, en forma de diagrama UML, una representación gráfica de los elementos necesarios con los que extender un EML ya existente para poder implementar los tres mecanismos descritos en los puntos anteriores. Básicamente se pueden diferenciar tres tipos principales de elementos:

- Evaluaciones: Combinan diferentes tipos de información para componer las fórmulas mediante las que llevar a cabo las estimaciones necesarias para valorar tanto el grado de satisfacción de los objetivos de aprendizaje marcados, como el éxito de las adaptaciones introducidas.
- Adaptaciones o *pokes*: Describen los distintos cambios que sobre la definición original del proceso o sobre su estado en ejecución será necesario aplicar, bien para corregir desviaciones detectadas por medio de las evaluaciones, o bien para adaptar el desarrollo al actual contexto de ejecución.
- Observaciones o *peeks*: Podrán ser utilizadas por el instructor tanto para recuperar información acerca del desarrollo del proceso como para servir de base para las fórmulas de las evaluaciones.

4.4.2. Modelo conceptual dinámico

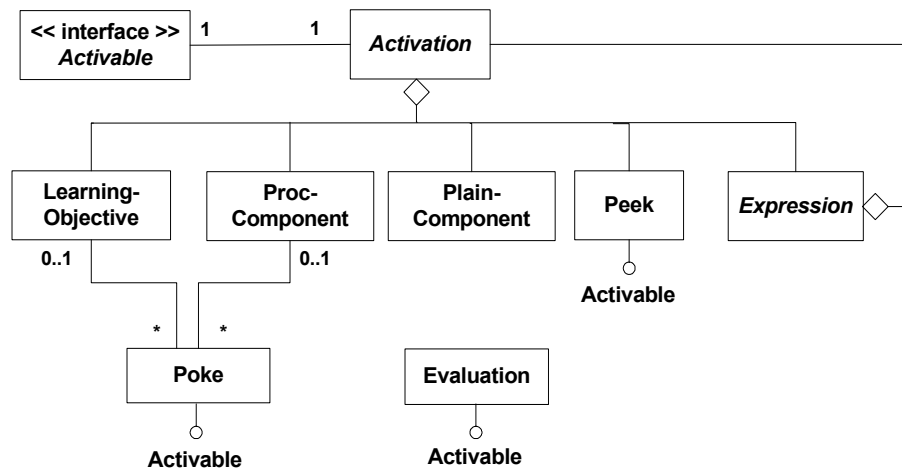


Figura 4.3: Modelo conceptual dinámico

Por otra parte, la figura 4.3 muestra, también en forma de diagrama UML, una representación gráfica de los elementos de la extensión de EML propuesta y de las relaciones que se establecen entre ellos durante el desarrollo del proceso, que básicamente pueden ser de dos tipos: activación o modificación. Así, los mismos elementos que sirven

para componer evaluaciones, pueden ser empleados para construir expresiones mediante las cuales determinar el momento exacto en que debe activarse la aplicación de una adaptación, la recuperación del valor de una observación o el cálculo del resultado de una fórmula de evaluación. Por otra parte, se debe tener en cuenta que dado que la introducción de adaptaciones puede modificar la estructura original del proceso, ciertas partes del mismo pueden sufrir variaciones o incluso ser suprimidas, de tal forma se afecte a la definición de los objetivos de aprendizaje y se eliminen algunos de ellos o se añadan nuevos.

4.4.3. Elementos del modelo conceptual

4.4.3.1. Perfil de evaluaciones o *evaluations-profile*

Para poder evaluar el éxito de un proceso de aprendizaje será necesario llevar a cabo un conjunto de evaluaciones con las que estimar el grado de consecución de ciertos objetivos de aprendizaje marcados. Dichas evaluaciones definirán una fórmula en la que se pueden combinar valores relacionados con los resultados obtenidos por los participantes del proceso durante el desarrollo del mismo junto a otros valores de naturaleza externa al mismo o los resultados obtenidos en otras evaluaciones de distintos objetivos de aprendizaje. Al conjunto de definiciones de objetivos de aprendizaje, evaluaciones de los mismos y componentes de las fórmulas de las evaluaciones le denominamos perfil de evaluaciones de un proceso o *evaluations-profile*.

Se pueden necesitar tres tipos distintos de elementos para poder especificar el perfil de evaluaciones de un proceso de aprendizaje:

- Componentes del proceso (*Proc-Components*): Representan elementos del proceso que contribuyen a la adquisición de ciertos objetivos de aprendizaje. Su definición está compuesta de un identificador, una referencia al elemento del proceso que representan y una expresión que será la empleada para estimar la actuación del participante del proceso en dicho componente. Esta expresión puede ser tanto un comando de monitorización, que permita recuperar el resultado de un test por ejemplo, o bien una expresión que combine referencias a otros componentes y elementos de información distintos para producir un valor. También es posible definir componentes del proceso abstractos, que agrupen definiciones de otros elementos pero no se correspondan con una parte concreta de la definición del proceso.

- Componentes simples de la evaluación (*Plain-Components*): Definen datos no relacionados con un elemento particular del proceso pero que son necesarios para poder componer las fórmulas de las evaluaciones de los objetivos de aprendizaje. Estos datos pueden ser obtenidos mediante los procesos de monitorización, recuperados de un recurso externo o bien introducidos directamente por el instructor. De esta manera, por ejemplo, el número de mensajes del alumno en los foros del curso puede ser utilizado para estimar su grado de interacción con otros alumnos, los resultados obtenidos en cursos anteriores y almacenados en el perfil del alumno se pueden emplear para estimar su progreso en una determinada materia; o bien el tutor puede introducir su valoración personal acerca de la capacidad de trabajo en grupo del alumno.
- Objetivos de aprendizaje (*Learning-Objective*): Cada objetivo de aprendizaje estará relacionado con un elemento *evaluación* y, opcionalmente, con una lista de *componentes de proceso* que contribuyen a la consecución de dicho objetivo de aprendizaje. El elemento *evaluación* se compone de dos expresiones: la primera consiste en una combinación de referencias a componentes, objetivos de aprendizaje, valores y operadores matemáticos y lógicos, y la segunda es una condición que determina el momento en el tiempo en el cuál se deberá analizar e interpretar la expresión primera, de tal forma que se recupere el valor que en ese momento tenga cada uno de los elementos referenciados y se obtenga la estimación del grado de satisfacción del objetivo de aprendizaje. En caso de proveerse, la lista de componentes de proceso asociados al objetivo de aprendizaje servirá para facilitar la tarea de análisis de los resultados del proceso, indicando al instructor que la lleve a cabo qué adaptaciones introducidas en dichos componentes pueden haber tenido influencia directa en la consecución del objetivo de aprendizaje indicado.

4.4.3.2. Adaptación o *poke*

Describe una modificación a llevar a cabo sobre un elemento del proceso de aprendizaje. Esta modificación puede ser de tres tipos distintos:

- Asignar: Dar un valor a un atributo de un elemento del proceso.
- Añadir: Introducir un nuevo elemento en la estructura. Normalmente requerirá la correspondiente definición *adaptation manifest* del nuevo elemento, y en ocasiones

indicar también el elemento padre y la posición que ocupará dentro de la secuencia de hijos del padre.

- Eliminar: Suprimir un elemento de una estructura concreta dentro del proceso o bien eliminar todas sus referencias a lo largo del proceso.

La descripción de la acción adaptativa puede incluir, además de la definición del propio cambio a realizar, un conjunto de identificadores de instancias de la UoL sobre los que restringir la modificación a realizar, una condición de activación del cambio, una expresión para la evaluación del éxito de la adaptación y referencias a elementos del perfil de evaluación que puedan verse afectados:

- Identificadores de instancias: En caso de que la definición de la adaptación o *poke* incluya un conjunto de identificadores de instancias de UoLs, la modificación será aplicada únicamente a dichas instancias en lugar de a todas las que en ese momento se encuentren en ejecución.
- Condición de activación: Permite especificar el momento en el cual la adaptación será aplicada al proceso. Para ello se puede proporcionar la fecha exacta de introducción o bien establecer una condición basada en la información recuperada por los elementos de monitorización, o combinar información de ambos tipos. De esta forma es posible crear una UoL genérica en la que todos sus elementos se encuentren inactivos, y especificar en un *adaptation poke* separado las fechas en las cuales se deberán ir activando y presentando las distintas actividades.
- Evaluación de la adaptación: Contiene una fórmula, similar a las empleadas para la evaluación de los objetivos del proceso, que puede ser empleada para estimar el éxito de la adaptación.
- Referencias a elementos del perfil de evaluación: Dado que la modificación del proceso por medio de adaptaciones puede tener influencia directa en el rendimiento del alumno en algunos de los objetivos de aprendizaje marcados, es conveniente incluir, cuando se conocen, referencias a los elementos de proceso u objetivos de aprendizaje que pueden verse afectados por la modificación, de tal forma que se facilite el análisis del impacto de la adaptación cuando se examinen los resultados del proceso.

La definición de la evaluación de la adaptación puede requerir la especificación de nuevos elementos en el modelo de evaluación del proceso. Además, dado que como resultado de las acciones de adaptación se pueden eliminar o incluir nuevos elementos en el proceso, puede ser necesario actualizar las fórmulas empleadas para la evaluación de los objetivos de aprendizaje definidos en el perfil de evaluaciones. En ambos casos será necesario introducir, junto con los tres tipos de definiciones del *adaptation poke* antes mencionados, una nueva definición en la que se recojan la descripción de los nuevos elementos del perfil de evaluaciones, así como posibles actualizaciones sobre las definiciones de los ya existentes.

4.4.3.3. Monitorización o *peek*

Describe una observación a llevar a cabo sobre la ejecución del proceso de aprendizaje. Para ello será necesario especificar el identificador del elemento del proceso a observar y el nombre del atributo o característica cuyo valor será recuperado. Además, deberán especificarse los identificadores de las instancias en ejecución de la UoL sobre las cuales se va a llevar a cabo la operación de monitorización y definir si el tipo de monitorización será de tipo *instantánea*, obteniéndose un valor de lectura para un momento dado en el tiempo, o *continua*, llevándose a cabo la observación de la característica o atributo de manera continuada a lo largo del tiempo. Es posible incluir también una *condición de activación* que identifique el momento a partir del cual se activará la observación. Esta condición puede especificarse empleando al valor de otro *peek* previamente introducido o bien utilizando una expresión de tipo temporal.

Por lo tanto, las referencias a otros *peeks* están compuestas de un identificador del *peek* y cero o más identificadores de las instancias a las cuales el *peek* fue aplicado. Así, por ejemplo, es posible crear condiciones que permitan activar una observación sobre la instancia de un participante que desempeña un determinado rol cuando el resto de participantes que desempeñan otros tipos de roles hayan completado una determinada actividad.

Basándose en las características y atributos propios de cada EML resulta sencillo crear plantillas que permitan al instructor la rápida definición de *WatcherPeeks*. Esta información puede ser mostrada por pantalla a través de la interfaz apropiada, a la vez que emplearse para la creación de condiciones para la activación de *pokes*. Así, no resulta difícil crear adaptaciones del tipo “*activar actividad X cuando la actividad Y sea completada*”.

4.5. Modelo de información

En esta sección se describe el modelo de información de la extensión para EMLs propuesta con el fin de permitir la implementación de los mecanismos previamente descritos. Para la descripción del modelo de información se ha seguido una notación similar a la que el *IMS Global Learning Consortium* emplea en la descripción de los modelos de información de sus especificaciones. Igualmente, los formatos de tipos de datos empleados son los mismos que los usados en dicha especificación para la declaración de sus elementos *property* [41]. El detalle de la notación empleada y formato de tipos de datos aparece recogido en el apéndice B.

4.5.1. Elementos relacionados con los procesos de evaluación

En esta sección se describen los distintos elementos que pueden incluirse dentro de la definición del perfil de evaluaciones de un determinado proceso de aprendizaje.

4.5.1.1. Tabla de información del elemento ‘Evaluations-Profile’

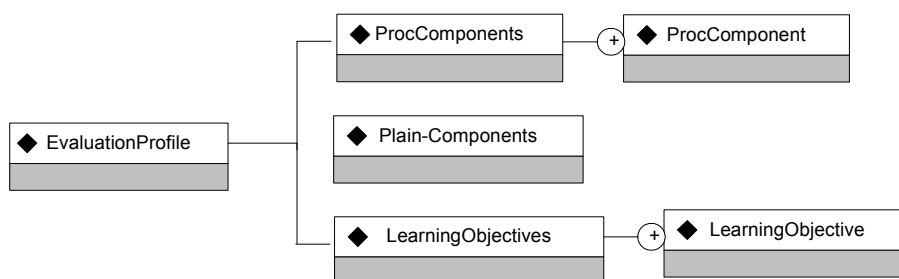


Figura 4.4: Diagrama del elemento Evaluation-Profile

Evaluations-Profile					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	Evaluations-Profile	Describe un modelo para la evaluación de un determinado proceso de aprendizaje	-	-	Sec.
0.1	<i>Identificador</i>	Identificador único dentro del modelo	S	1	ID
0.2	<i>Instancia</i>	Referencia a una instancia de una UoL	N	0..1	IDRef-EML
0.3	Proc-Components	Contenedor de las definiciones de Proc-Components	S	1	Cont.

0.3.1	Proc-Component	Representan una valoración de la actuación de un participante para un elemento o parte del proceso educativo que contribuye a la adquisición de un objetivo de aprendizaje.	S	1..*	Sec.
0.4	Plain-Components	Contenedor de las definiciones de <i>Plain-Components</i> , los cuales representan datos sobre la actuación de un participante en el proceso educativo no relacionados con una parte o elemento concreta del proceso.	S	1	Cont.
0.5	Learning-Objectives	Contenedor de las definiciones de Learning-Objectives	S	1	Cont.
0.5.1	Learning-Objective	Representa los objetivos de aprendizaje definidos para el proceso.	S	1..*	Sec.

Tabla 4.5: Tabla de datos del elemento Evaluations-Profile

4.5.1.2. Tabla de información del elemento 'Proc-Component'

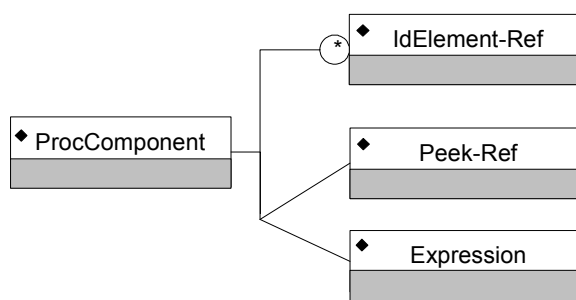


Figura 4.5: Diagrama del elemento Proc-Component

Proc-Component					
Nº	Nombre	Descripción	Obl.	M.	Tipo

0	Proc-Component	Representan una valoración de la actuación de un participante para un elemento o parte del proceso educativo que contribuye a la adquisición de un objetivo de aprendizaje.	-	-	Sec.
0.1	<i>Identificador</i>	Identificador único dentro del modelo	S	1	ID
0.2	<i>Descripción</i>	Descripción en lenguaje natural de la función del componente de proceso	N	1	String
0.3	<i>Tipo de datos</i>	Tipo de dato de la valoración del elemento	S	1	Token
0.4	<i>Valor inicial</i>	Valor inicial asignado al componente de proceso	N	1	String
0.5	IdElement-Ref	Identificador de un elemento del diseño con el que el componente se corresponde directamente	N	0..*	Vacia
0.5.1	<i>ref</i>	Referencia a un elemento del diseño	S	1	IDRef-EML
0.6		Elección	S	1	Elec.
0.6.1	Peek-Ref	Referencia a una observación	S	1	Sec
0.6.2	Expresión	Combinación de referencias a valores de componentes, u objetivos de aprendizaje con operadores aritméticos, relacionales o lógicos para producir un valor	S	1	String

Tabla 4.6: Tabla de datos del elemento Proc-Component

4.5.1.3. Tabla de información del elemento ‘Plain-Components’

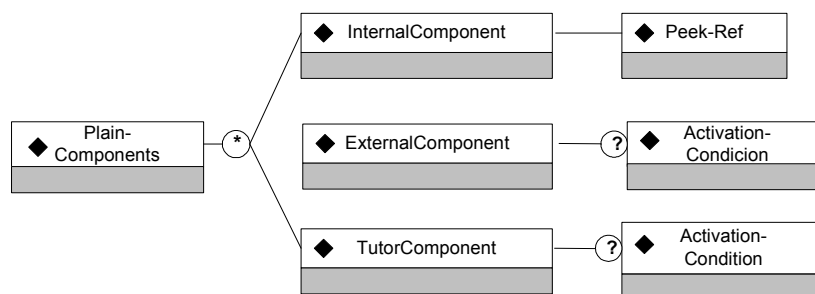


Figura 4.6: Diagrama elemento Plain-Component

Plain-Components					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	Plain-Components	Representa un dato sobre la actuación de un participante en el proceso educativo no relacionado con una parte o elemento concreta del proceso.	-	-	Elec.
0.1		Elección	N	0..*	Elec.
0.1.1	Internal-Component	Dato sobre el desarrollo del proceso que puede ser obtenido mediante observaciones automatizadas	S	1	Sec.
0.1.2	External-Component	Dato recuperado de un dispositivo externo	S	1	Sec.
0.1.3	Tutor-Component	Dato que se obtiene directamente del tutor o instructor del proceso	S	1	Sec.

Tabla 4.7: Tabla de datos del elemento Plain-Component

4.5.1.4. Tabla de información del elemento ‘Internal-Component’

Internal-Component					
Nº	Nombre	Descripción	Obl.	M.	Tipo

0	Internal-Component	Dato sobre el desarrollo del proceso que puede ser obtenido mediante observaciones automatizadas.	-	-	Sec.
0.1	<i>Identificador</i>	Identificador único dentro del modelo	S	1	ID
0.2	<i>Descripción</i>	Descripción en lenguaje natural de la función y objetivo del elemento	N	1	String
0.3	<i>Tipo de datos</i>	Tipo de dato de la valoración del elemento	S	1	Token
0.4	<i>Valor inicial</i>	Valor inicial asignado al elemento	N	0..1	String
0.5	Peek-Ref	Referencia a una observación	S	1	Sec

Tabla 4.8: Tabla de datos del elemento Internal-Component

4.5.1.5. Tabla de información del elemento ‘External-Component’

External-Component					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	External-Component	Dato recuperado de un dispositivo externo.	-	-	Sec.
0.1	<i>Identificador</i>	Identificador único dentro del modelo	S	1	ID
0.2	<i>Descripción</i>	Descripción en lenguaje natural de la función y objetivo del elemento	N	1	String
0.3	<i>Tipo de datos</i>	Tipo de dato de la valoración del elemento	S	1	Token
0.4	<i>Valor inicial</i>	Valor inicial asignado al elemento	N	0..1	String
0.5	<i>Comando</i>	Expresión que analizada por el motor de ejecución permite recuperar un dato almacenado externamente (repositorio, perfil de usuario, etc)	S	1	String

0.6	Activation-Condition	Condición que determina el momento en que debe recuperarse la información. En caso de no especificarse la información se recuperará al comienzo del proceso.	N	0..1	Sec
-----	----------------------	--	---	------	-----

Tabla 4.9: Tabla de datos del elemento External-Component

4.5.1.6. Tabla de información del elemento ‘Tutor-Component’

Tutor-Component					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	Tutor-Component	Dato que se obtiene directamente del tutor o instructor del proceso	-	-	Sec.
0.1	<i>Identificador</i>	Identificador único dentro del modelo	S	1	ID
0.2	<i>Descripción</i>	Descripción en lenguaje natural de la función y objetivo del elemento	N	1	String
0.3	<i>Tipo de datos</i>	Tipo de dato de la valoración del elemento	S	1	Token
0.4	<i>Valor inicial</i>	Valor inicial asignado al elemento	N	0..1	String
0.5	<i>Texto</i>	Texto que se mostrará al tutor al solicitar la introducción del valor para el elemento	S	1	String
0.6	Activation-Condition	Condición que determina el momento en que debe recuperarse la información. En caso de no especificarse la información se recuperará al comienzo del proceso.	N	0..1	Sec

Tabla 4.10: Tabla de datos del elemento Tutor-Component

4.5.1.7. Tabla de información del elemento ‘Learning Objective’

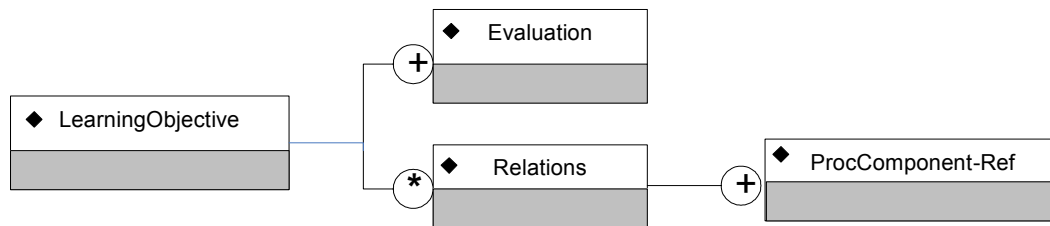


Figura 4.7: Diagrama del elemento Learning-Objective

Learning-Objective					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	Learning-Objective	Representa un objetivo de aprendizaje definido para el proceso	-	-	Sec.
0.1	<i>Identificador</i>	Identificador único dentro del modelo	S	1	ID
0.2	<i>Descripción</i>	Descripción en lenguaje natural de la función del componente de proceso	N	0..1	String
0.3	<i>Tipo de datos</i>	Tipo de dato de la valoración del elemento	S	1	Token
0.4	<i>Valor inicial</i>	Valor inicial asignado al componente de proceso	N	0..1	String
0.5	Evaluation	Define el mecanismo y el momento en el cual se llevará a cabo la evaluación del grado de satisfacción del objetivo de aprendizaje	S	1	Sec
0.6	Relaciones	Especifica qué componentes del proceso pueden influir en la adquisición del objetivo de aprendizaje	N	0..1	Cont.
0.6.1		Elección	S	1..*	Elec.
0.6.1.1	ProcComponent-Ref	Referencia a un componente de proceso definido en el perfil de evaluaciones	S	1..*	Vacia
0.6.1.1.1	<i>idref</i>	Referencia a un componente del modelo de la extensión	S	1	IDRef

0.6.1.2	LearningObjective-Ref	Referencia a otro objetivo de aprendizaje	S	1..*	Vacia
0.6.1.2.1	<i>idref</i>	Referencia a un componente del modelo de la extensión	S	1	IDRef

Tabla 4.11: Tabla de datos del elemento Learning-Objective

4.5.1.8. Tabla de información del elemento 'Evaluation'

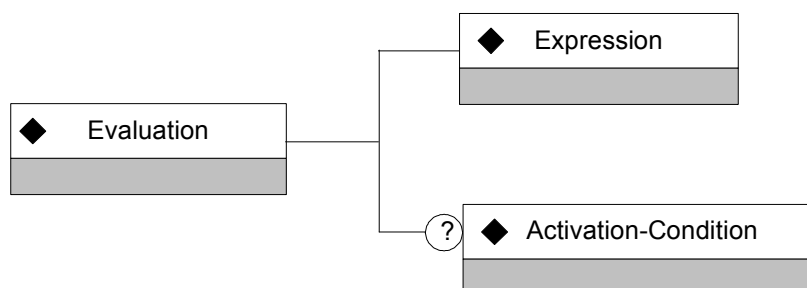


Figura 4.8: Diagrama del elemento Evaluation

Evaluation					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	Evaluation	Define el mecanismo y el momento en el cual se llevará a cabo la evaluación de un objetivo de aprendizaje o una adaptación	-	-	Sec.
0.1	Identificador	Identificador único dentro del modelo	S	1	ID
0.2	Expresión	Combinación de referencias a valores de componentes u objetivos de aprendizaje con operadores aritméticos, relacionales o lógicos para producir un valor	S	1	String

0.3	Activation-Condition	Determina el momento en el cual se debe llevar a cabo la evaluación. En caso de no especificarse la evaluación se llevará a cabo al final del proceso	N	1	Sec
-----	----------------------	---	---	---	-----

Tabla 4.12: Tabla de datos del elemento Evaluation

4.5.1.9. Tabla de información del elemento ‘Activation-Condition’

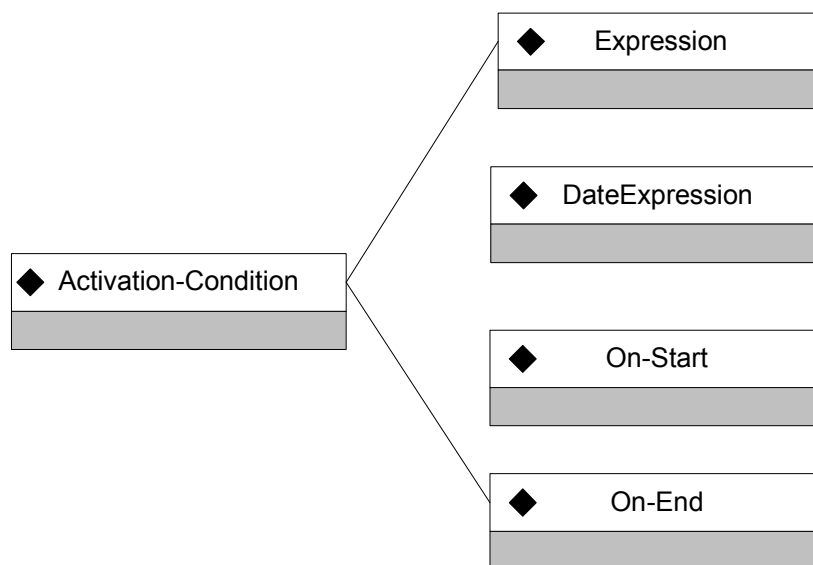


Figura 4.9: Diagrama del elemento Activation-Condition

Activation-Condition					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	Activation-Condition	Define una condición que al ser evaluada a <i>true</i> activará la lectura de una evaluación, componente de evaluación, <i>poke</i> o <i>peek</i> . La condición se puede especificar definiendo el momento de activación en el tiempo o bien empleando una combinación de valores y referencias a otros componentes del modelo.	-	-	Sec.

0.2		Elección	S	1	Elec.
0.2.1	Expression	Combinación de referencias a valores de componentes u objetivos de aprendizaje con operadores aritméticos, relacionales o lógicos para producir un valor	S	1	String
0.2.2	DateExpression	Expresión que identifica un instante de tiempo. Este puede ser una fecha exacta o estar referido al comienzo del proceso o al momento de creación del elemento	S	1	Sec
0.2.3	On-Start	La condición se evaluará a <i>true</i> al comenzar el proceso.	S	1	Vacia
0.2.4	On-End	La condición se evaluará a <i>true</i> al finalizar el proceso.	S	1	Vacia

Tabla 4.13: Tabla de datos del elemento Activation-Condition

4.5.2. Elementos relacionados con los procesos de adaptación

En esta sección se recoge la tabla de información y diagrama del elemento “*Poke*”.

4.5.2.1. Tabla de información del elemento ‘Poke’

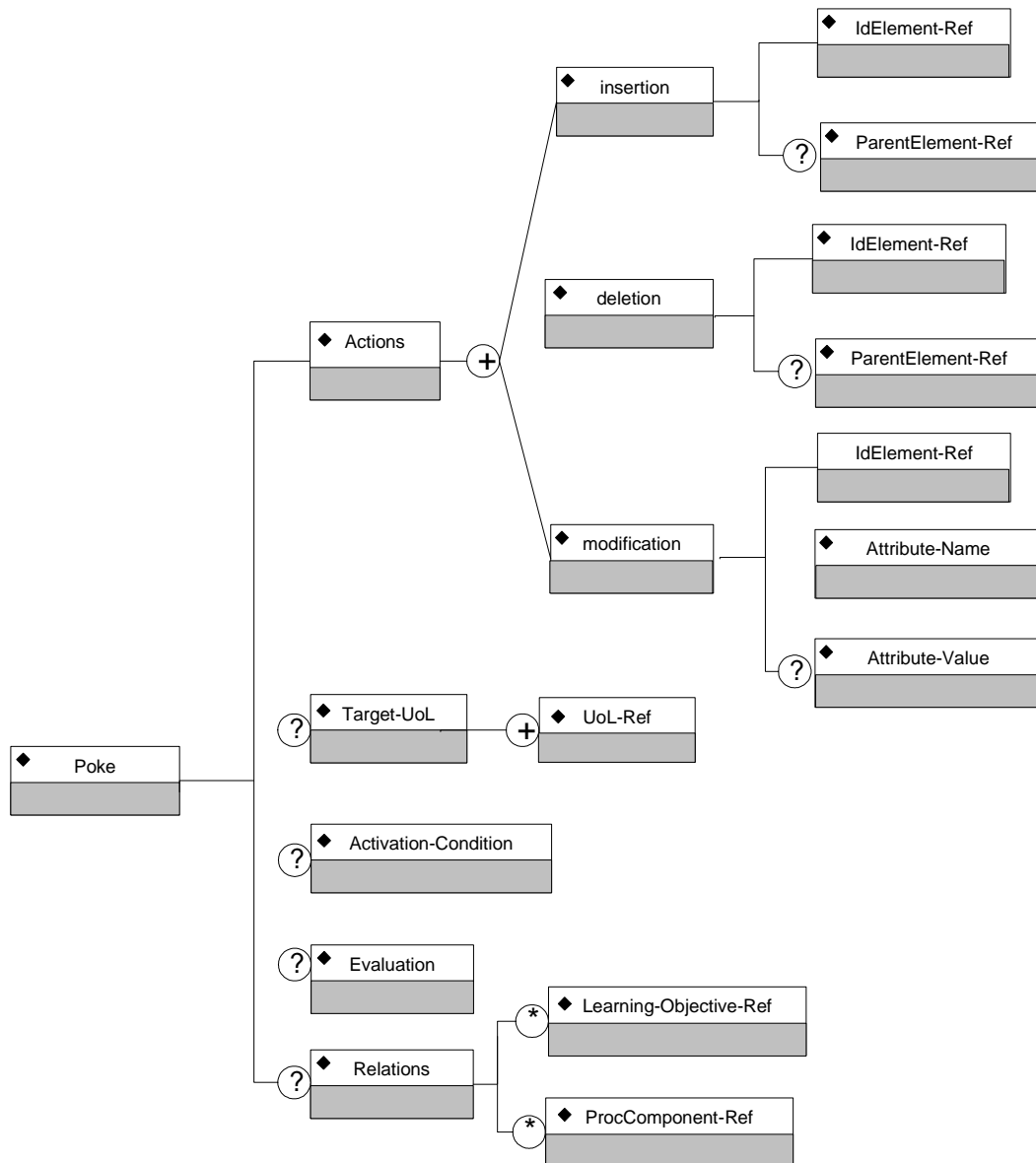


Figura 4.10: Diagrama elemento Poke

Poke					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	Poke	Describe una modificación sobre el diseño del proceso	-	-	Sec.
0.1	<i>Identificador</i>	Identificador único dentro del modelo	S	1	ID
0.2	<i>Descripción</i>	Descripción en lenguaje natural de la adaptación	N	1	String

0.3	Actions	Descripciones de los cambios a efectuar	S	1	Sec.
0.3.1		Elección	S	1..*	Elec.
0.3.1.1	Insertion	Descripción de una operación de inserción de un nuevo elemento en el diseño	S	1	Sec
0.3.1.1.1	IdElement-Ref	Identificador de un elemento del diseño que será añadido	S	1	Vacia
0.3.1.1.1.1	<i>ref</i>	Referencia a un elemento del diseño	S	1	IDRef-EML
0.3.1.1.2	Parent-Ref	Identificador del elemento de diseño padre del nuevo	N	0..1	Vacia
0.3.1.1.2.1	<i>pos</i>	Posición que ocupará el nuevo elemento dentro de los hijos	N	0..1	Number
0.3.1.1.2.2	<i>ref</i>	Referencia a un elemento del diseño	S	1	IDRef-EML
0.3.1.2	Deletion	Descripción de una operación de eliminación de un elemento del diseño	S	1	Sec
0.3.1.2.1	IdElement-Ref	Identificador de un elemento del diseño que será eliminado	S	1	Vacia
0.3.1.2.2	Parent-Ref	En caso de incluirse un identificador de un elemento padre, se eliminarán sólo las referencias en dicha estructura padre. Si no se incluye, se eliminarán todas las referencias en el proceso así como su definición.	N	0..1	Vacia
0.3.1.2.2.1	<i>ref</i>	Referencia a un elemento del diseño	S	1	IDRef-EML
0.3.1.3	Modification	Descripción de una operación de modificación de un elemento del diseño	S	1	Sec
0.3.1.3.1	IdElement-Ref	Identificador de un elemento del diseño que será modificado	S	1	Vacia

0.3.1.3.1.1	<i>ref</i>	Referencia a un elemento del diseño	S	1	IDRef-EML
0.3.1.3.2	Attribute-Name	Nombre del atributo del elemento a cambiar	S	1	String
0.3.1.3.3	Attribute-Value	Valor a asignar al atributo del elemento	N	0..1	String
0.4	Target UoL	Contiene los identificadores de cero o más instancias de la UoL. En caso de no especificarse, la operación se llevará a cabo sobre todas ellas.	N	0..1	Cont.
0.4.1	UoL-ref	Referencia a una instancia de una UoL	S	1..*	Vacia
0.4.1.1	<i>ref</i>	Referencia a un elemento del diseño	S	1	IDRef-EML
0.5	Activation-Condition	Define el momento en el que se introducirá la adaptación. Puede especificar el momento exacto en el tiempo o utilizar una expresión que dispare la adaptación al ser evaluada a verdadera	N	0..1	Elec.
0.6	Evaluación	Define el mecanismo y el momento en el cual se llevará a cabo la evaluación del grado de la adaptación	N	0..1	Sec
0.7	Relations	Especifica qué componentes del proceso u objetivos de aprendizaje pueden verse influidos por la introducción de la adaptación	N	0..1	Cont.
0.7.1	ProcComponent-Ref	Referencia a un componente de proceso definido en el perfil de evaluaciones	N	0..*	Vacia
0.7.1.1	<i>idref</i>	Referencia a un componente del modelo de la extensión	S	1	IDRef

0.7.2	Learning-Objective-Ref	Referencia a un objetivo de aprendizaje definido en el perfil de evaluaciones	N	0..*	Vacia
0.7.2.1	<i>idref</i>	Referencia a un componente del modelo de la extensión	S	1	IDRef

Tabla 4.14: Tabla de datos del elemento Poke

4.5.3. Elementos relacionados con los procesos de observación

En esta sección se recoge la tabla de información y diagrama del elemento “*Peek*”.

4.5.3.1. Tabla de información del elemento ‘Peek’

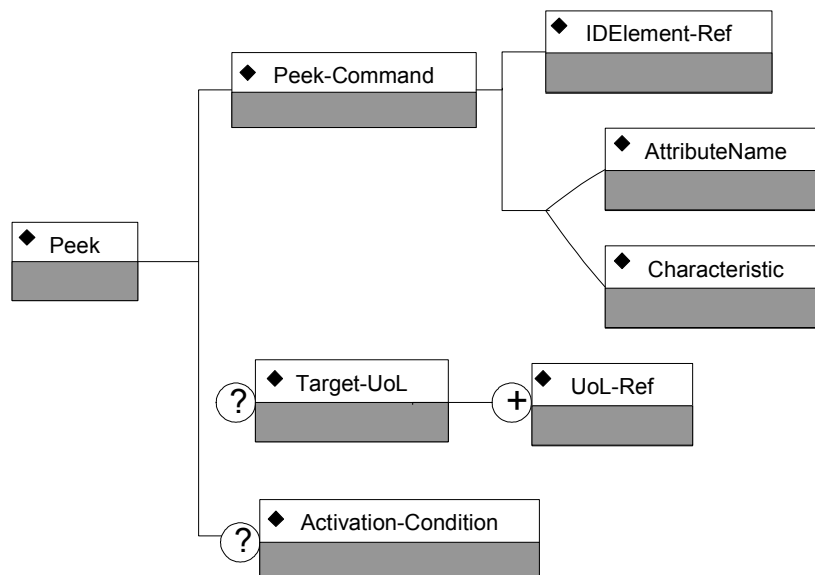


Figura 4.11: Diagrama del elemento Peek

Peek					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	Peek	Describe una observación a realizar sobre la estructura o desarrollo del proceso	-	-	Sec.
0.1	<i>Identificador</i>	Identificador único dentro del modelo	S	1	ID

0.2	<i>Descripción</i>	Descripción en lenguaje natural de la observación	N	1	String
0.3	<i>Tipo de datos</i>	Tipo de dato de la observación	S	1	Token
0.4	<i>Valor inicial</i>	Valor inicial asignado a la observación	N	1	String
0.5	<i>Tipo de peek</i>	Puede ser instantáneo (la información sólo se recupera en un momento preciso) o continuo	S	1	Token
0.6	Peek-Command	Describe la observación a llevar a cabo	S	1	Cont.
0.6.1	IdElement-Ref	Identificador del elemento de diseño a observar	S	1	Vacia
0.6.1.1	<i>ref</i>	Referencia a un elemento del diseño	S	1	IDRef-EML
0.6.2		Elección	S	1	Elec.
0.6.2.1	AttributeName	Nombre del atributo a observar	S	1	Token
0.6.2.2	Characteristic	Nombre de la característica a observar	S	1	Token
0.7	Target UoL	Contiene los identificadores de cero o más instancias de la UoL. En caso de no especificarse la observación se llevará a cabo sobre todas ellas.	N	0..1	Cont.
0.7.1	UoL-ref	Referencia a una instancia de una UoL	S	1..*	Vacia
0.7.1.1	<i>ref</i>	Referencia a un elemento del diseño	S	1	IDRef-EML
0.8	Activation-Condition	Define el momento en el cual la observación será aplicada. Puede especificar el momento exacto en el tiempo o utilizar una expresión que dispare la observación al ser evaluada a verdadera	N	0..1	Elec.

Tabla 4.15: Tabla de datos del elemento Peek

4.5.3.2. Tabla de información del elemento 'Peek-Ref'

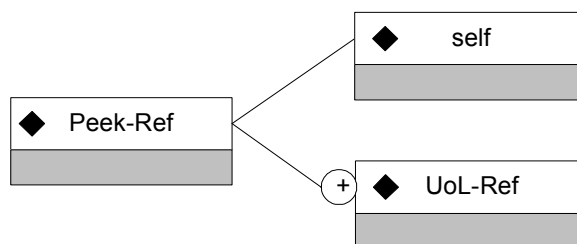


Figura 4.12: Diagrama del elemento Peek-Ref

Peek-Ref					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	Peek-Ref	Referencia a una observación. Si no se incluye ningún identificador de instancia ni el elemento 'self' se considera que es una referencia a las observaciones con ese identificador en todas las instancias	-	-	Sec.
0.1	<i>idref</i>	Referencia a un componente del modelo de la extensión	S	1	IDRef
0.2		Elección	S	1	Elec.
0.2.1	Self	La referencia se refiere a la propia instancia donde es procesada.	S	1	Vacia
0.2.2	UoL-ref	Referencia a una instancia de una UoL.	S	1..*	Vacia
0.2.2.1	<i>ref</i>	Referencia a un elemento del diseño	S	1	IDRef-EML

Tabla 4.16: Tabla de datos del elemento Peek-Ref

4.5.4. Elementos comunes a todos los procesos

En esta sección se recogen las tablas de información y diagramas de distintos elementos que pueden aparecer tanto en la definición del perfil de evaluaciones, como en la descripción de *peeks* o *pokes*.

4.5.4.1. Tabla de información del elemento 'Expression'

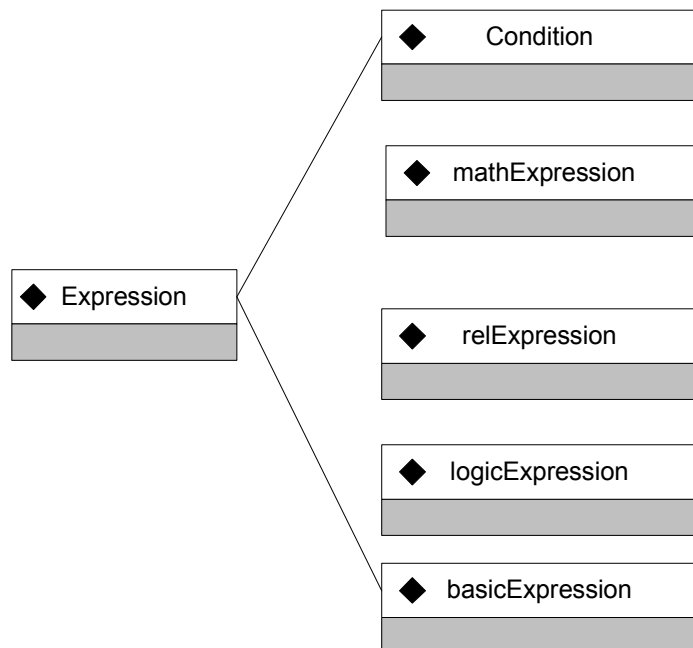


Figura 4.13: Diagrama del elemento Expression

Expression					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	Expression	Combinación de referencias a valores de componentes, informaciones u objetivos de aprendizaje con operadores aritméticos, relacionales o lógicos para producir un valor	-	-	Sec.
0.1		Elección	S	1	Elec.
0.1.1	condition	Expresión que denota una condición	S	1	Sec.
0.1.2	mathExpression	Expresión matemática	S	1	Sec.
0.1.3	relExpression	Expresión relacional	S	1	Sec.
0.1.4	logicExpression	Expresión lógica	S	1	Sec.
0.1.5	basicExpression	Expresión formada por un único componente básico	S	1	Sec.

Tabla 4.17: Tabla de datos del elemento Expression

4.5.4.2. Tabla de información del elemento 'Condition'

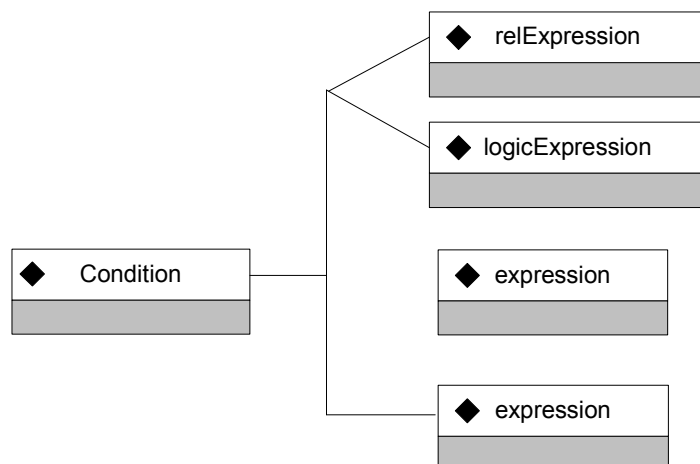


Figura 4.14: Diagrama del elemento Condition

Condition					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	Condition	Expresión que denota una condición	-	-	Sec.
0.1		Elección.	S	1	Elec.
0.1.1	relExpresión	Expresión relacional que representa la condición a evaluar.	S	1	Sec.
0.1.2	logicExpresión	Expresión lógica que representa la condición a evaluar.	S	1	Sec.
0.2	Expression	Expresión que será procesada en caso de que la condición sea evaluada a <i>true</i> .	S	1	Sec.
0.3	Expression	Expresión que será procesada en caso de que la condición sea evaluada a <i>false</i> .	S	1	Sec.

Tabla 4.18: Tabla de datos del elemento Condition

4.5.4.3. Tabla de información del elemento ‘MathExpression’

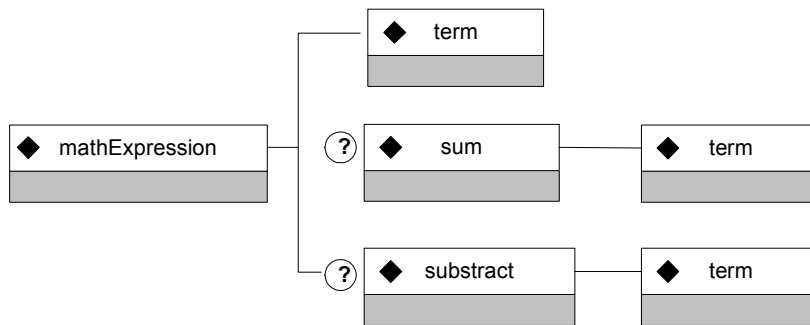


Figura 4.15: Diagrama del elemento MathExpression

MathExpression					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	MathExpression	Expresión matemática	-	-	Sec.
0.1	Term	Término de una expresión matemática	S	1	Sec.
0.2	Sum	Operador suma	N	0..1	Sec.
0.2.1	Term	Término de una expresión matemática	S	1	Sec.
0.3	Substract	Operador resta	N	0..1	Sec.
0.3.1	Term	Término de una expresión matemática	S	1	Sec.

Tabla 4.19: Tabla de datos del elemento MathExpression

4.5.4.4. Tabla de información del elemento ‘Term’

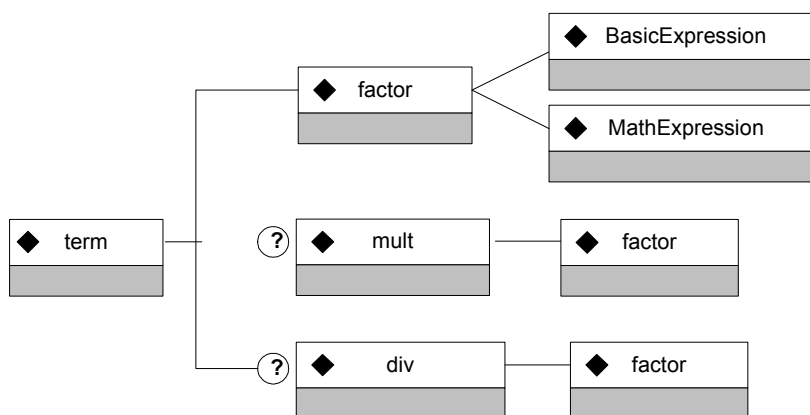


Figura 4.16: Diagrama del elemento Term

Term					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	Term	Término de una expresión matemática	-	-	Sec.
0.1	Factor	Factor de una expresión matemática	S	1	Elec.
0.1.1	BasicExpression	Expresión formada por un único componente básico	S	1	Elec.
0.1.2	MathExpression	Expresión matemática	S	1	Elec.
0.2	Mult	Operador multiplicación	N	0..1	Sec.
0.2.1	Factor	Factor de una expresión matemática	S	1	Elec.
0.3	Div	Operador división	N	0..1	Sec.
0.3.1	Factor	Factor de una expresión matemática	S	1	Elec.

Tabla 4.20: Tabla de datos del elemento Term

4.5.4.5. Tabla de información del elemento ‘Factor’

Factor					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	Factor	Factor de una expresión matemática	-	-	Elec.
0.1		Elección	S	1	Elec.
0.1.1	BasicExpression	Expresión formada por un único componente básico	S	1	Sec.
0.1.2	MathExpression	Expresión matemática	S	1	Sec.

Tabla 4.21: Tabla de datos del elemento Factor

4.5.4.6. Tabla de información del elemento ‘RelExpression’

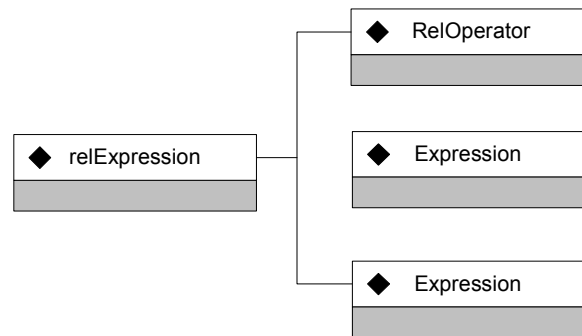


Figura 4.17: Diagrama del elemento relExpression

relExpression					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	RelExpression	Expresión relacional	-	-	Sec.
0.1	RelOperator	Operador de relación entre las dos expresiones	S	1	Vacía
0.1.1	<i>Operator</i>	Operador Relacional: <i>equalTo</i> , <i>notEqualTo</i> , <i>greatherThan</i> , <i>lessThan</i> , <i>greatherThanOrEqual</i> , <i>lessThanOrEqual</i>	S	1	Token
0.2	Expression	Combinación de referencias a valores de componentes u objetivos de aprendizaje con operadores aritméticos, relacionales o lógicos para producir un valor	S	1	Elec.
0.3	Expression	Combinación de referencias a valores de componentes u objetivos de aprendizaje con operadores aritméticos, relacionales o lógicos para producir un valor	S	1	Elec.

Tabla 4.22: Tabla de datos del elemento relExpression

4.5.4.7. Tabla de información del elemento ‘LogicExpression’

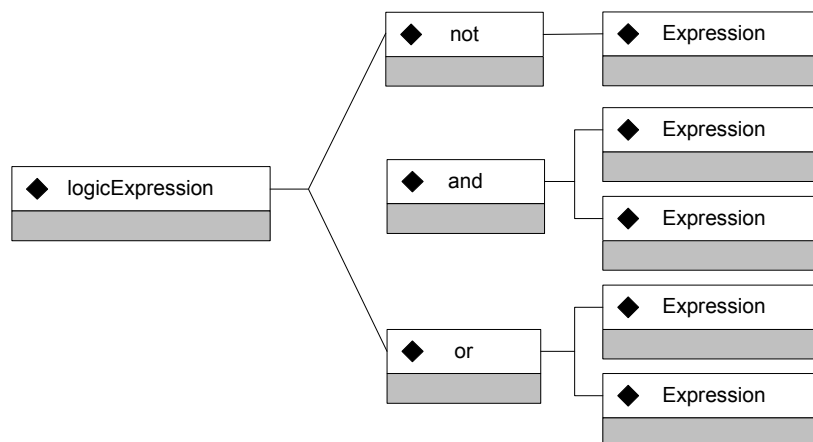


Figura 4.18: Diagrama del elemento LogicExpression

LogicExpression					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	LogicExpression	Expresión lógica	-	-	Sec.
0.1		Elección	S	1	Elec.
0.2	not	Operador negación	S	1	Sec.
0.2.1	Expression	Combinación de referencias a valores de componentes u objetivos de aprendizaje con operadores aritméticos, relacionales o lógicos para producir un valor	S	1	Elec.
0.3	and	Operador and	S	1	Sec.
0.3.1	Expression	Combinación de referencias a valores de componentes u objetivos de aprendizaje con operadores aritméticos, relacionales o lógicos para producir un valor	S	1	Elec.
0.3.2	Expression	Combinación de referencias a valores de componentes u objetivos de aprendizaje con operadores aritméticos, relacionales o lógicos para producir un valor	S	1	Elec.

0.4	or	Operador or	S	1	Sec.
0.4.1	Expression	Combinación de referencias a valores de componentes u objetivos de aprendizaje con operadores aritméticos, relacionales o lógicos para producir un valor	S	1	Elec.
0.4.2	Expression	Combinación de referencias a valores de componentes u objetivos de aprendizaje con operadores aritméticos, relacionales o lógicos para producir un valor	S	1	Elec.

Tabla 4.23: Tabla de datos del elemento LogicExpression

4.5.4.8. Tabla de información del elemento ‘BasicExpression’

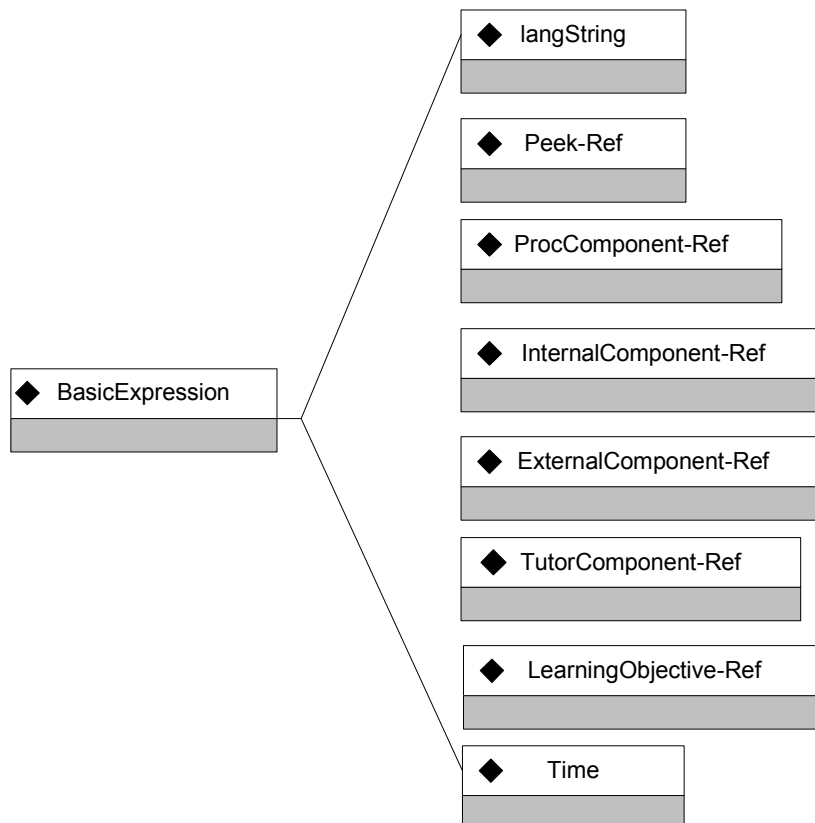


Figura 4.19: Diagrama del elemento BasicExpression

BasicExpression					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	BasicExpression	Expresión formada por un único componente básico	-	-	Sec.
0.1		Elección	S	1	Elec.
0.1.1	langString	String	S	1	Sec.
0.1.2	Peek-Ref	Referencia a una observación.	S	1	Sec.
0.1.3	ProcComponent-Ref	Referencia a un componente de proceso definido en el perfil de evaluaciones	S	1	Vacia
0.1.3.1	<i>idref</i>	Referencia a un componente del modelo de la extensión	S	1	IDRef
0.1.4	InternalComponent-Ref	Referencia a un componente de proceso simple de tipo interno definido en el perfil de evaluaciones	S	1	Vacia
0.1.4.1	<i>idref</i>	Referencia a un componente del modelo de la extensión	S	1	IDRef
0.1.5	ExternalComponent-Ref	Referencia a un componente de proceso simple de tipo externo definido en el perfil de evaluaciones	S	1	Vacia
0.1.5.1	<i>idref</i>	Referencia a un componente del modelo de la extensión	S	1	IDRef
0.1.6	TutorComponent-Ref	Referencia a un componente de proceso simple de tipo tutor definido en el perfil de evaluaciones	S	1	Vacia
0.1.6.1	<i>idref</i>	Referencia a un componente del modelo de la extensión	S	1	IDRef
0.1.7	Learning-Objective-Ref	Referencia a un objetivo de aprendizaje definido en el perfil de evaluaciones	S	1	Vacia

0.1.7.1	<i>idref</i>	Referencia a un componente del modelo de la extensión	S	1	IDRef
0.1.8	Time	Cadena de caracteres que representa un momento en el tiempo en formato dd/mm/yyyy hh24:mi:ss	S	1	String

Tabla 4.24: Tabla de datos del elemento basicExpression

4.5.4.9. Tabla de información del elemento ‘DateExpression’

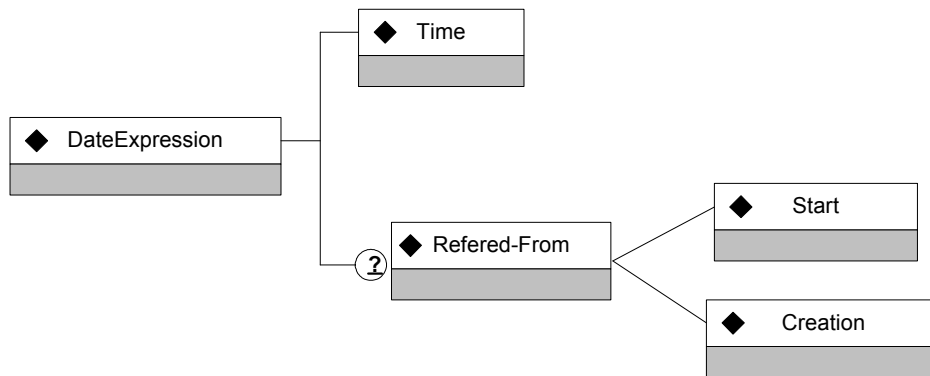


Figura 4.20: Diagrama del elemento DateExpression

DateExpression					
Nº	Nombre	Descripción	Obl.	M.	Tipo
0	DateExpression	Expresión que especifica un momento concreto en el tiempo	-	-	Sec.
0.1	Time	Cadena de caracteres que representa un momento en el tiempo en formato dd/mm/yyyy hh24:mi:ss	S	1	String
0.2	Referencia	Indica que el momento de tiempo especificado debe tomarse en referencia a otro momento	N	0..1	Elec.
0.2.1	Start	El momento de tiempo es especificado con respecto al inicio del proceso	S	1	Vacia

0.2.2	Creation	El momento de tiempo es especificado con respecto al momento de creación del elemento	S	1	Vacia
-------	----------	---	---	---	-------

Tabla 4.25: Tabla de datos del elemento DateExpression

4.6. Método para el diseño iterativo de procesos de aprendizaje

Como se ha mencionado en capítulos anteriores, la aplicación práctica de un diseño de aprendizaje conlleva siempre cierto grado de flexibilidad debido a la imposibilidad de conocer de antemano cuáles serán las reacciones de los participantes al mismo. Los instructores toman el diseño como base o punto de partida, observando la evolución y respuestas de los participantes a las actividades del mismo e introduciendo ejemplos o explicaciones extra que refuercen la asimilación de ciertos conceptos, repitiendo actividades, ajustando los tiempos estimados para completar los test, etc. En cualquier caso, a medida que un diseño de aprendizaje va siendo más utilizado, menor es el número de adaptaciones que se requiere introducir durante su aplicación ya que el propio diseño va siendo refinado mediante su uso: la experiencia obtenida en anteriores ejecuciones es capturada en la propia definición del proceso de tal forma que es capaz de dar respuesta a un mayor rango de reacciones de los participantes. Esto significa que la fase de diseño del proceso realmente no concluye hasta que no se requiere la introducción de nuevas modificaciones durante su ejecución. Este enfoque de desarrollo es característico de los métodos ágiles de desarrollo software [7, 18], donde es común la iteración a través de ciclos en los que se verifica y corrigen diversas partes del diseño [67].

La figura 4.21 muestra una representación gráfica de las diversas fases que componen el ciclo del diseño iterativo del proceso de aprendizaje soportado en un entorno computacional. El proceso comienza una vez que se termina una definición inicial del modelo del curso y comienza su ejecución. Los instructores observan las reacciones de los participantes y van introduciendo las apropiadas adaptaciones. El éxito de las adaptaciones introducidas es evaluado y, una vez que el proceso concluye, se mide también el grado de satisfacción de los objetivos de aprendizaje especificados. Basándose en esa información se genera una nueva versión del modelo del curso que incluye las adaptaciones que han demostrado introducir alguna mejora. Esta nueva versión seguirá

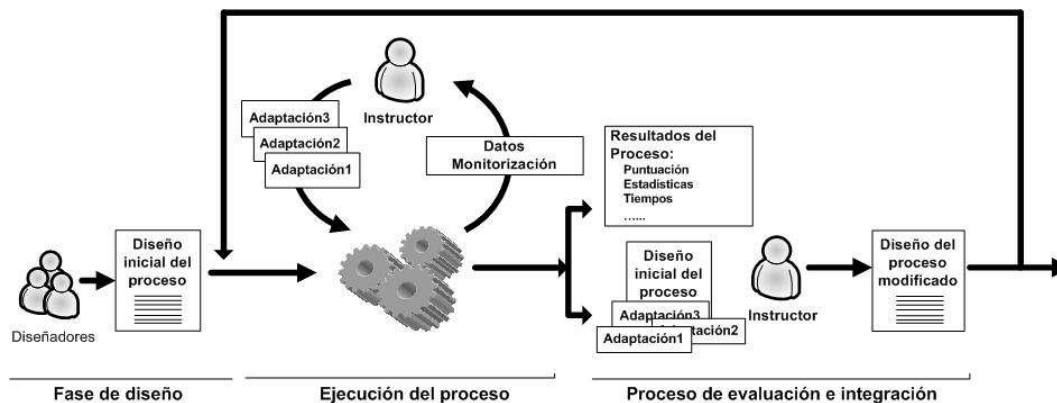


Figura 4.21: Fases del proceso de diseño iterativo de una unidad de aprendizaje

de nuevo el mismo proceso, y así sucesivamente hasta que deje de ser necesaria la introducción de nuevas adaptaciones.

4.6.1. Fases del método de diseño iterativo de procesos de aprendizaje

En esta sección se describen cada una de las fases que componen el método de diseño iterativo de procesos de aprendizaje: monitorización de la ejecución, introducción de adaptaciones, evaluación de adaptaciones, evaluación del proceso, y finalmente, integración de las adaptaciones.

4.6.1.1. Monitorización de la ejecución

Para poder detectar problemas en el diseño e introducir las adaptaciones apropiadas es necesario que los instructores sean capaces de monitorizar las interacciones de los participantes así como su progreso a lo largo del proceso de aprendizaje. Cuanta más información tengan a su disposición mejor podrán identificar las causas de los problemas que vayan surgiendo. Por ejemplo, si sólo pudiesen recuperar información acerca de la puntuación de los alumnos en las diferentes actividades, únicamente podrán detectar si su rendimiento está siendo adecuado o no. En cambio, si pudiesen recuperar información acerca de qué recursos han visitado y cuánto tiempo han pasado en cada uno de ellos, serán capaces de extraer conclusiones más precisas e introducir las recomendaciones y adaptaciones apropiadas. Por otro lado, la comparación de información recuperada de distintas instancias del proceso de aprendizaje correspondientes a diferentes participantes facilitará la identificación de la naturaleza del problema.

4.6.1.2. Introducción de adaptaciones

Basándose en la información recuperada en las actividades de monitorización, los instructores llevarán a cabo la descripción de las variaciones del proceso necesarias para garantizar el éxito del mismo.

4.6.1.3. Evaluación de las adaptaciones

Una vez que la adaptación ha sido introducida es necesario evaluar el efecto producido y si realmente se ha satisfecho el objetivo perseguido. La evaluación, por tanto, no está directamente basada en los resultados del alumno, sino en comparar las consecuencias esperadas de la adaptación con las actuales. La dificultad radica en identificar qué es consecuencia de la introducción de la adaptación y qué no lo es. La correlación entre el efecto de diferentes adaptaciones también debe tenerse en cuenta a la hora de la evaluación, así como la posible introducción de efectos colaterales.

Por otra parte también deben analizarse las posibles relaciones existentes entre distintas adaptaciones de cara a su futura integración o almacenamiento. Así, por ejemplo, deberá tenerse en cuenta la posibilidad de que una adaptación haya podido introducir un cambio sobre un recurso previamente introducido por otra, por ejemplo. Para identificar este tipo de interdependencias el instructor puede hacer uso de las relaciones definidas entre adaptaciones y componentes de proceso, estudiando aquéllas que han tenido como objeto del cambio los mismos componentes o partes relacionadas entre sí.

4.6.1.4. Evaluación del proceso

Una vez que el proceso de aprendizaje ha concluido, los resultados producidos deben ser examinados para identificar los puntos fuertes y débiles de su diseño. Esta evaluación está principalmente basada en la información acerca del grado de satisfacción obtenido por los participantes del proceso para cada uno de los objetivos de aprendizaje definidos. Si los resultados en alguno de ellos resultan ser bajos para la mayoría de los participantes, los diseñadores pueden considerar incluir material complementario, revisar el enfoque pedagógico empleado o la dificultad de las actividades de evaluación del alumno. En cualquier caso, las causas de un resultado pobre pueden también encontrarse en circunstancias externas al diseño o en un perfil del alumno incorrecto. Por tanto, es conveniente también establecer el grado de fiabilidad de los resultados actuales comparándolos con los obtenidos en pasadas ejecuciones del curso.

4.6.1.5. Integración de las adaptaciones

Una vez que los resultados del proceso han sido analizados, tiene lugar la fase de integración. De esta forma, las adaptaciones que han probado introducir algún tipo de mejora en el diseño original se convierten en parte permanente del mismo. Este proceso conlleva varios pasos: en primer lugar, los instructores examinan los resultados de las evaluaciones de las adaptaciones y seleccionan aquéllas que serán integradas de manera definitiva. A continuación, el sistema irá aplicando cada una de las adaptaciones sobre el diseño siguiendo el mismo orden de introducción original. La introducción de cada adaptación debe ser validada de forma independiente de tal manera que en caso de fallo se facilite la identificación de interdependencias entre adaptaciones. Para terminar, la definición resultante del proceso será depurada eliminando antiguas definiciones de elementos y recursos que ya no son incluidos en la estructura final.

Por otra parte, a lo largo de este proceso es posible que se detecten ciertas adaptaciones que han demostrado implementar una solución satisfactoria al problema que las originó, pero que no han sido seleccionadas para integración definitiva al responder dicho problema a circunstancias particulares del contexto de ejecución. En este caso, estas adaptaciones deben ser convenientemente marcadas indicando el contexto y propósito de la adaptación y almacenadas en la correspondiente base de datos, de tal forma que, si a lo largo de futuras ejecuciones del proceso dichas circunstancias vuelven a tener lugar, los instructores puedan disponer de una adaptación que ha demostrado ya su eficacia en la resolución del problema.

4.6.2. Implementación del método

Empleando un motor de ejecución de diseños de aprendizaje que soporte el modelo de adaptación, observación y evaluación descrito a lo largo del capítulo, es posible implementar las principales fases del método de diseño iterativo, estableciendo una correspondencia directa entre las mismas y los principales elementos del modelo. Así, la fase de monitorización de la ejecución puede ser implementada mediante el desarrollo de un *Progress Watcher* que se encargue de leer y procesar los diferentes *peeks* que instructor o diseñador vayan introduciendo. Las adaptaciones del diseño pueden describirse mediante *adaptation pokes* y la forma de evaluarlas especificada en la correspondiente sección de los mismos y, por último, la fase de evaluación puede ser principalmente soportada empleando los elementos del perfil de evaluaciones. Para poder implementar la fase de integración será necesario que el motor de ejecución almacene una copia del

diseño original previo al desarrollo de la instrucción sobre la que, una vez ésta haya finalizado, se deberán ir aplicando las adaptaciones que han sido seleccionadas en la fase de evaluación.

4.7. Arquitectura de extensión de un motor de ejecución de EMLs

Un motor de ejecución es el programa que se encarga de interpretar una UoL. Presenta las diferentes actividades y recursos a los alumnos e instructores participantes mientras que controla sus interacciones. A menudo no resulta complicado establecer una equivalencia entre los elementos de las especificaciones de los EMLs y el concepto de clase de la Programación Orientada a Objetos (OOP, *emph*Object Oriented Programming), por lo que, a la hora de implementar un motor de ejecución, suele ser natural emplear este tipo de técnicas.

Por otra parte, es necesario que el programa sea capaz de interpretar las adaptaciones descritas mediante *adaptation pokes* y aplicarlas durante la propia ejecución del proceso, es decir, de introducir distintos tipos de modificaciones sobre la definición de los elementos del EML. Por ello es necesario disponer de una estructura suficientemente flexible para permitir la introducción de dichos cambios y que, en caso de ser necesario implementar nuevos tipos de adaptaciones, no requiera ser modificada. Este escenario se corresponde con el descrito por el patrón *visitor* [30]: *Realizar la misma operación sobre los elementos de una estructura de objetos y ser capaz de definir nuevas operaciones sin cambiar las clases que implementan los elementos de la estructura.* Siguiendo los principios de diseño de este tipo de patrón, distinguiremos dos tipos de clases: las que representan los elementos objeto de las operaciones y las que implementan las propias operaciones o *visitors*. Esta separación permite añadir nuevas operaciones en forma de *visitors* sin tener que modificar las clases que representan los elementos. La interacción entre los dos tipos de clases se lleva a cabo a través de una operación *accept* implementada en los objetos de la estructura y que recibe un objeto *visitor* como argumento. Este objeto incluirá un método *visit* por cada tipo de clase sobre la que puede llevar a cabo su operación, de tal forma que los objetos de la estructura responderán a las llamadas a sus métodos *accept* invocando el método *visit* del objeto *visitor* correspondiente a su tipo.

La Fig. 4.22 muestra la aplicación de este patrón al diseño de un motor de ejecución

de EMLs. En este caso, los elementos de la estructura son los elementos del EML susceptibles de ser adaptados — *EMLElements* — y las clases *visitor* se corresponden con las subclases de *EMLModifier*, siendo la operación a realizar la propia adaptación de los elementos de la estructura, o bien cualquier otra operación que se pueda llevar a cabo sobre dichos los elementos, como por ejemplo medir el progreso del alumno en las actividades.

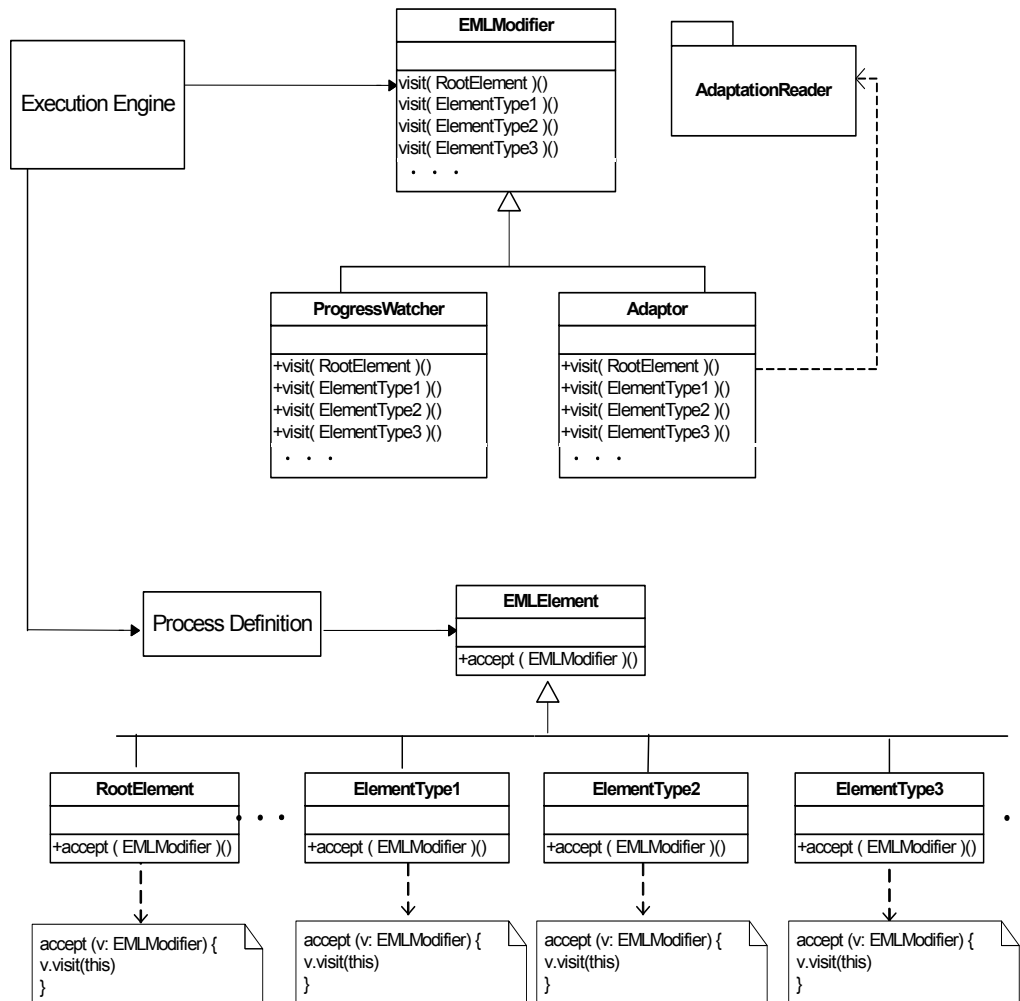


Figura 4.22: Estructura del motor de ejecución

Por cada *adaptation poke* e instancia de la UoL se creará una instancia de la clase *Adaptor*, la cual es una subclase de *EMLModifier* y encapsula todos los cambios descritos en la definición del *poke*. Con la información de la UoL cargada en memoria, será posible invocar la operación *accept* del elemento raíz en la jerarquía de elementos del EML y pasar la instancia del *Adaptor* como argumento. Esto activará el método *visit*

del *Adaptor* correspondiente a dicho elemento raíz, el cual recuperará, uno por uno, los diferentes elementos para los cuales se ha definido alguna adaptación, llamando sucesivamente a sus métodos *accept* y pasando una vez más la instancia del *Adaptor* como parámetro. La Fig. 4.23 ilustra este proceso.

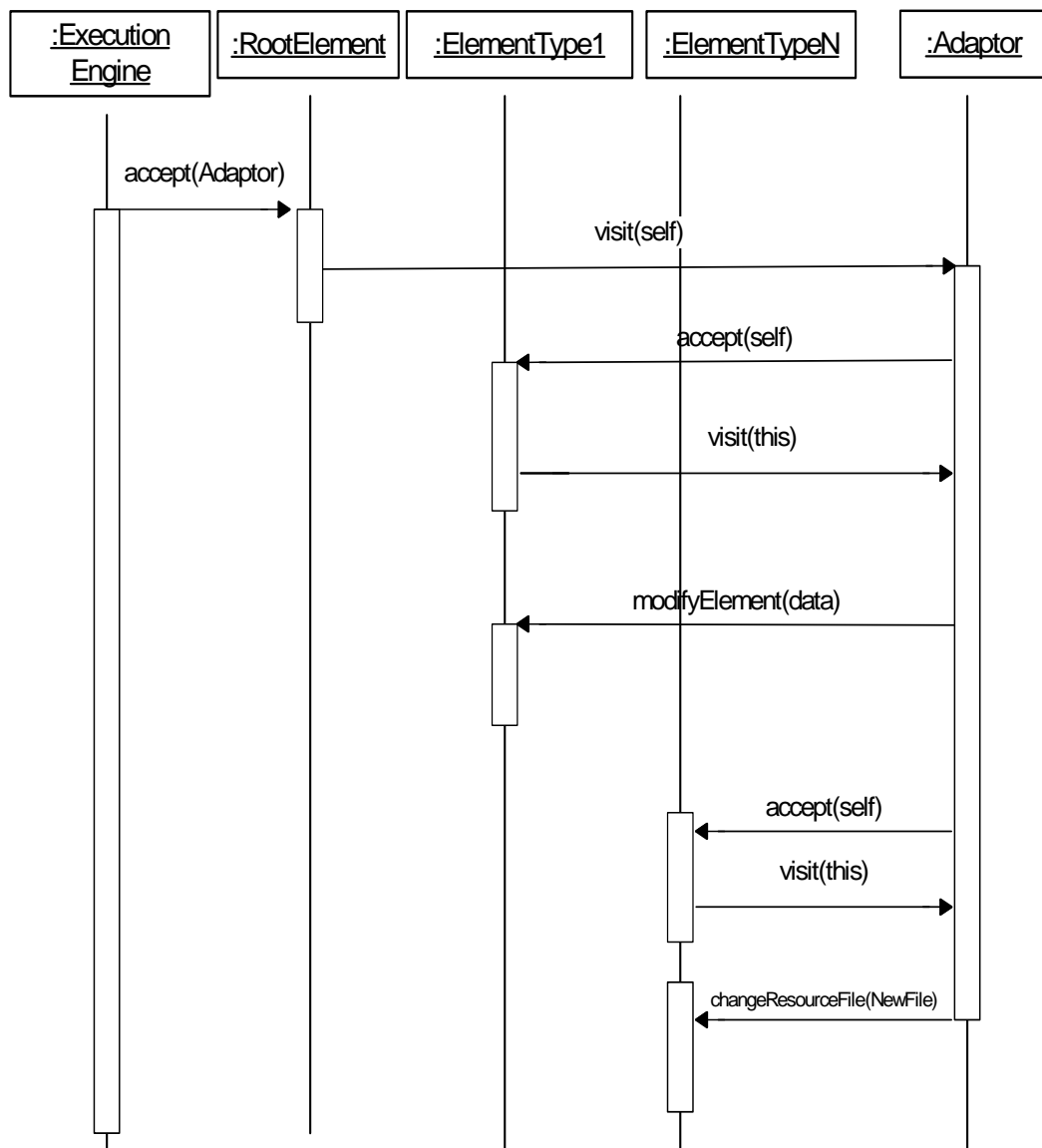


Figura 4.23: Diagrama de secuencia de la aplicación de un *adaptation poke*

Las nuevas adaptaciones se pueden aplicar fácilmente. Únicamente es necesario crear nuevos ficheros de adaptación e incluirlos en el *content package* junto con los ficheros originales, o bien aplicarlos a una instancia ya en ejecución. El *AdaptionReader* generará los objetos *Adaptor* apropiados y los transferirá al motor de ejecución para su aplicación.

La inclusión de nuevas operaciones sobre los elementos de la estructura resulta igualmente sencilla. Por ejemplo, para crear una operación que permita recuperar información acerca de diversos componentes del proceso, se puede crear una clase *ProgressWatcher* que herede de *EMLModifier* e implemente las operaciones necesarias en sus métodos *visit*. Dado que la nueva clase extiende *EMLModifier*, el nuevo tipo de objetos se puede pasar también como argumento de las operaciones *accept* de las clases de los elementos de la especificación. Éstas llamarán al método correspondiente del *ProgressWatcher* de tal manera que las nuevas operaciones se puedan llevar a cabo sin necesidad de haber introducido ningún cambio sobre la interfaz de las clases. Para éstas, es completamente transparente el tipo particular de *EMLModifier* que reciben en sus operaciones *accept*.

Esta organización es fácil de mantener y la interfaz de las clases *EMLModifier* sólo habría de ser modificada en el caso de que futuras especificaciones consideren la inclusión de nuevos tipos de elementos.

4.7.1. Implementación mediante Programación Orientada a Aspectos

La implementación del diseño previamente descrito requiere la modificación de las clases que implementan algunos de los elementos de la especificación original para incluir una nueva operación *accept*. Esta modificación puede ser salvada empleando técnicas de Programación Orientada a Aspectos, incrementando a la vez la flexibilidad y modularidad de la arquitectura del motor.

La Programación Orientada a Aspectos extiende el paradigma de orientación a objetos introduciendo el concepto de *aspecto*, el cual permite encapsular en módulos separados comportamientos transversales que afectan a varias clases. Así, los aspectos se pueden definir de forma separada de las clases y métodos originales, siendo los compiladores e intérpretes los encargados de llevar a cabo la integración de todos ellos antes de la conversión definitiva a código binario.

Empleando este enfoque, la operación *accept* se puede implementar describiéndola mediante aspectos separados y estableciendo las condiciones en las que la ejecución original será interceptada para lanzar el nuevo código. De esta forma, nuevos aspectos y condiciones de disparo de los mismos se pueden introducir o eliminar sin necesidad de alterar la estructura del motor de ejecución original.

La arquitectura del motor resulta así más flexible, siendo posible el mantenimiento

separado de los ficheros del módulo de extensión y de los ficheros del motor original, y simplificando la actualización a nuevas versiones de este último.

4.7.2. Validación de la adaptación mediante ontología

Tal y como se mencionó en la sección 4.1.4, cada vez que una adaptación es aplicada con éxito a la definición de un proceso de aprendizaje, es necesario lanzar un proceso de validación, similar al que se lleva a cabo durante el proceso de publicación, de tal forma que se verifique la conformidad de la definición del proceso adaptado con las normas y restricciones definidas en la especificación del EML utilizado para su descripción. La implementación de este proceso de validación puede resultar laboriosa al requerir de la codificación de todas las normas y restricciones descritas en la correspondiente especificación del EML y difícil de mantener en caso de que nuevas versiones de la especificación sean publicadas. Este proceso puede simplificarse si la validación es llevada a cabo empleando una ontología que capture dichas normas y restricciones. De esta forma, a partir de la definición del proceso cargada en memoria, se llevará a cabo un proceso inverso al seguido en la publicación de la UoL que resulte en la obtención de un fichero que contenga la definición del diseño del proceso adaptado en el mismo lenguaje empleado para la definición del proceso original. Este fichero deberá a su vez traducirse al lenguaje de representación de conocimiento correspondiente a la ontología empleada en la validación, de tal manera que se pueda verificar la conformidad del diseño adaptado con la especificación del EML capturado por la misma. Así, una vez terminado el proceso de validación, la adaptación del proceso será finalmente publicada o rechazada según todas las restricciones especificadas en la ontología hayan podido ser satisfechas o no.

Capítulo 5

Evaluación

Una vez definidos un modelo, método y mecanismo que posibiliten la adaptación en tiempo de ejecución de procesos de aprendizaje supervisados por instructor e implementados utilizando un EML, se aborda la validación de la solución propuesta con el fin de demostrar la consecución de los objetivos planteados en el primer capítulo de este trabajo, así como la utilidad, calidad y factibilidad de la solución y la generalidad y riqueza expresiva del modelo de adaptación.

Con este objeto se han empleado un conjunto de métodos de evaluación que incluyen tanto pruebas observacionales como experimentales:

- Con objeto de probar la factibilidad de la solución se ha desarrollado un caso de estudio consistente en la aplicación del modelo y mecanismo de adaptación a un caso concreto de EML, el IMS LD.
- Con objeto de validar la generalidad y riqueza expresiva del modelo de adaptación se ha desarrollado un conjunto de casos de uso.
- Con objeto de completar la validación de la expresividad del modelo de adaptación y probar la factibilidad del método de diseño iterativo propuesto, se ha aplicado el mismo a tres escenarios de aprendizaje reales replicados mediante UoLs.
- Con objeto de constatar la utilidad de la solución así como la calidad de la misma se ha llevado a cabo una evaluación de sus características por expertos.

A continuación se proporciona una descripción de cada una de las pruebas de evaluación realizadas y un resumen de las mismas en el que se incluye su objetivo, elementos de la solución que han participado y resultado de la evaluación.

5.1. Desarrollo de un caso de estudio: IMS LD

Con objeto de probar la factibilidad de la solución propuesta, en esta sección se aplicará el modelo y mecanismo de adaptación previamente descrito a un EML concreto, el IMS LD. Para ello será necesario llevar a cabo diversas tareas: en primer lugar se deberá definir el conjunto de elementos que podrán ser adaptados y el conjunto de características de los mismos que se permitirá modificar y/o monitorizar. Por otra parte, se necesitará desarrollar un *Learning Design Player* que sea capaz de interpretar los comandos de adaptación y de introducirlos durante la propia ejecución del proceso. Por último, será necesario utilizar una ontología que capture la semántica de la especificación y facilite la implementación de la validación del proceso adaptado.

5.1.1. Selección de características adaptables

Para poder implementar el mecanismo de adaptación descrito sobre procesos cuyo diseño ha sido llevado a cabo mediante la especificación del IMS LD, es necesario acotar en primer lugar qué elementos de la misma podrán ser modificados mediante adaptación.

5.1.1.1. Adaptaciones estáticas

Podemos dividir las adaptaciones estáticas, o adaptaciones que modifican la definición del proceso, en dos grupos, según los elementos modificados sean de nivel A o B:

- Nivel A: Tal y como se describió en las secciones previas, el mecanismo de adaptación que se describe en este documento resulta adecuado únicamente cuando el objetivo es la introducción de pequeñas variaciones en la definición de un proceso de aprendizaje. Si la modificación implica alteraciones importantes en la estructura, resulta más conveniente llevar a cabo un rediseño completo del proceso. Por este motivo, de los elementos de nivel A de la especificación, únicamente se permitirá crear o eliminar actividades, entornos y objetivos de aprendizaje. Sus definiciones originales podrán igualmente ser modificadas, sustituyendo las referencias a recursos que contengan, sus condiciones de terminación, etc. Sobre los elementos *Method*, *Play* y *Act* únicamente permitiremos modificar dichas condiciones de terminación.

- Nivel B: se permitirá la modificación de la definición de propiedades y de los nuevos valores para las condiciones de terminación y acción post-terminación de las actividades. La creación de nuevas propiedades carece de sentido al no permitir adaptaciones que describan nuevas condiciones o modifiquen la definición de las existentes.

La tabla 5.1 describe las distintas adaptaciones estáticas que pueden ser llevadas a cabo sobre los elementos del IMS LD, incluyendo, por una parte, la lista de atributos que pueden ser modificados para cada elemento del manifiesto y, por otra, los distintos tipos de componentes que pueden ser añadidos o eliminados de la estructura.

Se debe hacer notar que actualmente no se permite la adaptación de los elementos *condición* de nivel B ni de elementos *notificación* de nivel C. El principal motivo de esta restricción reside en que la especificación del IMS LD no incluye ningún atributo identificador en la definición de estos elementos, lo cual impide su referencia mediante elementos del modelo de adaptación. Actualmente se están considerando diversas opciones de ampliación del modelo que permitan salvar esta limitación, tal y como se describe en la sección “Trabajos futuros” del capítulo siguiente.

5.1.1.2. Adaptaciones dinámicas

La especificación del IMS LD incluye un conjunto de reglas que debe ser implementado por todo motor de ejecución de procesos de aprendizaje descritos mediante este lenguaje de modelado y que regirán el comportamiento en ejecución del proceso. Dichas normas definen qué elementos serán visibles para un determinado participante en un momento dado, y cuál será el flujo de ejecución. Para poder implementar dichas reglas, el IMS LD Player deberá manipular, dependiendo de su tipo, dos características de los elementos: su visibilidad y su compleción. Además, será posible que el valor asignado a un determinado *property* varíe igualmente en ejecución. Esta información ha sido tomada como punto de partida para definir la lista de características dinámicas que de cada elemento se permitirá modificar mediante adaptación, como muestra la tabla 5.2.

5.1.2. Selección de características observables

La especificación del IMS LD permite a los diseñadores implementar elementos *monitor services* dentro de una UoL. Estos elementos permiten a los participantes del proceso visualizar los valores de las propiedades de su propio dossier o del de todos los participantes que tengan asignado un rol determinado. De esta forma, dependiendo de

Tabla 5.1: Adaptaciones estáticas en el IMS LD

Modificación de Atributos		
Nivel A		
Elemento	Atributo	Valor
Learning Design	Learning Objective	id + idRef-EML ¹
Method	complete-unit-of-learning	'when-play-completed' 'time-limit'
Play	complete-play	'when-last-act-completed' 'time-limit'
Act	complete-act	'when-role-completed' 'time-limit'
Activity Structure	title	String
	number-to-select	number
Learning Activity	title	String
	description	id + idRef-EML ¹
	complete-activity	'user choice' 'time limit'
	feedback-description	id + idRef-EML ¹
Support Activity	Learning Objective	id + idRef-EML ¹
	title	String
	description	id + idRef-EML ¹
	complete-activity	'user choice' 'time limit'
Environment	feedback-description	id + idRef-EML ¹
	LearningObject	id + idRef-EML ¹
Nivel B		
Elemento	Atributo	Valor
Property	initial value	string
Learning Activity	complete-activity	'when-property-value-is-set' + idRef-EML ²
	on-completion	'change-property-value' + idRef-EML ²
Support Activity	complete-activity	'when-property-value-is-set' + idRef-EML ²
	on-completion	'change-property-value' + idRef-EML ²
Modificaciones en la estructura		
Elemento Padre	Elemento	
Learning Design	Learning Objective	
Role-Part	Activity Structure	
Role-Part	Learning Activity	
Role-Part	Support Activity	
Activity Structure	Learning Activity	
Activity Structure	Support Activity	
Activity Structure	Environment	
Learning Activity	Environment	
Learning Activity	Learning Objective	
Support Activity	Environment	
Environment	Learning Object	
Environment	Service	

¹El primer identificador será utilizado como descriptor de un nuevo *item* el cual contendrá una referencia al recurso referenciado por el segundo identificador

²El identificador referencia a un elemento *property* del diseño

Tabla 5.2: Adaptaciones dinámicas en el IMS LD

Elemento	Estado Visibilidad	Estado Compleción	Valor
UnitOfLearning	X	X	
Play	X	X	
RolePart	X	X	
Act	X	X	
ActivityStructure	X	X	
LearningActivity	X	X	
SupportActivity	X	X	
Environment	X		
Send-Mail	X		
Learning Object	X		
Monitor	X		
Property			X

la manera en que se implementan, los *monitor services* se pueden usar tanto para que los alumnos conozcan la puntuación que han alcanzado en determinadas actividades, como para que sean los tutores quienes inspeccionen dichos valores.

Por otra parte, al implementar un *monitor service*, las propiedades cuyo valor va a ser posible observar en ejecución deben ser determinadas con anterioridad durante la fase de diseño. Como complemento o alternativa a la acción de *monitor service* los diseñadores pueden considerar la implementación de un *ProgressWatcher* que permita observar un mayor rango de atributos y características del progreso, así como introducir nuevas monitorizaciones en tiempo de ejecución o bien modificar la definición de las ya existentes. Para llevar a cabo dicha implementación es necesario definir el conjunto de atributos y características del proceso cuyo valor podrá ser recuperado.

5.1.2.1. Observaciones estáticas

Dado que todo motor de ejecución de diseños especificados mediante IMS LD proporciona una implementación para un nivel dado del modelo de información descrito en la especificación [41], deberá ser posible recuperar el valor de cada uno de los atributos definidos para cada elemento descrito en dicho modelo de información para el nivel correspondiente, sin ningún tipo de restricción.

5.1.2.2. Observaciones dinámicas

La tabla 5.3 muestra la selección de elementos y características dinámicas del IMS LD cuyos valores podrán ser recuperados mediante la implementación de un *Progress-Watcher*. Esta selección ha sido llevada a cabo basándonos en la información que sobre los distintos elementos que componen un Learning Design es almacenada en el motor de ejecución CopperCore e incluye, además de las características dinámicas *valor*, *compleción* y *visibilidad*, las fechas en que tuvieron lugar el primer y último acceso sobre el elemento, así como el número total efectuado y el valor por defecto asignado.

5.1.3. Extensión de un motor de ejecución

El empleo de técnicas de programación orientada a objetos resulta adecuado para el desarrollo de un motor de ejecución de diseños de aprendizaje especificados mediante IMS LD o *Learning Design Player* dado que el propio modelo conceptual de la especificación del IMS LD fue desarrollado teniendo en cuenta su correspondencia con el diseño orientado a objetos. Por otra parte, el modelo de información de la especificación define una jerarquía entre sus elementos que tiene como nodo raíz el elemento *Learning Design* y que encaja con la arquitectura propuesta en el capítulo anterior.

Dado que el motor de ejecución CopperCore ha sido desarrollado empleado tecnología J2EE es posible diseñar, siguiendo el enfoque propuesto, un módulo de extensión que permita incorporar al mismo la capacidad de interpretar y aplicar, sobre una determinada instancia en ejecución de una UoL, las adaptaciones y monitorizaciones que el instructor o diseñador vayan introduciendo.

5.1.4. Validación del proceso adaptado

Tal y como se mencionó en la sección 4.1.4, es necesario validar la consistencia de la definición del proceso adaptado tras cada modificación, comprobando su conformidad con las normas y restricciones del EML correspondiente. En nuestro caso de estudio dicha verificación se ha llevado a cabo empleando dos mecanismos distintos de validación: por una parte se ha hecho uso de una ontología que captura las restricciones que deben ser satisfechas por la estructura del proceso adaptado, y por otra, se ha ampliado el patrón de diseño *visitor* para incluir una nueva operación de validación que garantice que el estado del proceso tras la modificación es correcto.

5.1.4.1. Validación de la descripción del proceso mediante ontología

En [4, 55] los autores describen una ontología que captura la semántica del nivel A de la especificación del IMS LD junto con las restricciones que deben verificar sus relaciones entre los conceptos, y que posteriormente fue ampliada en [5] para comprender el nivel B. Dado que dichas restricciones se expresan formalmente, esta ontología permite detectar inconsistencias en la descripción de UoL adaptadas cada vez que dichas restricciones no son verificadas. La implementación de la ontología se ha llevado a cabo en *Frame-based Logic* (F-logic) [50], y se ha usado el razonador FLORA-2 [89] para comprobar los axiomas de la ontología al introducir las instancias de conceptos.

El proceso para detectar inconsistencias se puede resumir de la siguiente manera: (1) se aplica un *adaptation poke* a una UoL modificando la definición del LD contenida en su manifiesto; (2) a continuación, la representación en XML-schema del LD adaptado es traducido a F-Logic; (3) finalmente, el razonador FLORA-2 es invocado para responder a las consultas asociadas a los axiomas que deben ser verificados.

5.1.4.2. Validación del estado en ejecución mediante codificación de restricciones

Si bien la ontología previamente mencionada permite verificar si la descripción de un proceso de aprendizaje es válida con respecto a los niveles A y B de la especificación del IMS LD, es necesario garantizar que también son satisfechas las restricciones que la especificación establece con respecto al estado en ejecución del proceso. Dado que en estos momentos no se dispone de una ontología que capture dichas restricciones, ha sido necesario emplear otro método para llevar a cabo dicha validación. Así, se ha ampliado la implementación del patrón *visitor* utilizada para llevar a cabo la adaptación y monitorización de los elementos del proceso, incluyendo una nueva operación correspondiente a la validación y representada mediante una nueva sub-clase de *EMLModifier* llamada *RuntimeValidator* (fig. 5.1). Dicha clase encapsula la descripción del conjunto de restricciones que será necesario verificar teniendo en cuenta la especificación del IMS LD y la lista de características dinámicas que de cada elemento se permitirá modificar mediante adaptación (figura 5.2). De esta forma, y tras la introducción del conjunto de cambios asociados a un determinado *poke*, se invocarán los métodos *visit* de la clase *RuntimeValidator* correspondientes a los elementos que han sido modificados, de tal forma que se disparen las comprobaciones adecuadas.

Así, tanto para una secuencia de actos, como para una secuencia de actividades

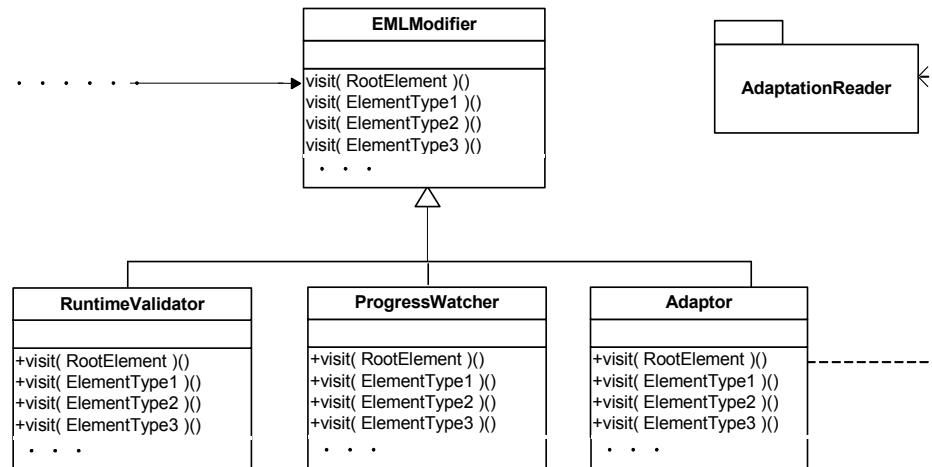


Figura 5.1: Nueva operación en el patrón visitor para implementar la validación del estado en ejecución

agrupadas en una *Activity-structure* de tipo *sequence*, se comprobará que no se encuentren elementos marcados como completos intercalados con elementos marcados incompletos y viceversa. Por otra parte el instructor deberá tener en cuenta que una vez validado y hechos efectivos los cambios del *poke* las reglas de completación definidas en la especificación serán disparadas. Así, si se marca un elemento de cierto nivel como completo, todos los elementos de nivel más bajo contenidos en él serán terminados. Por el contrario, marcar un elemento de nivel alto como incompleto sin alterar las condiciones que han provocado su terminación provocará que, una vez dichas condiciones vuelvan a ser evaluadas por el IMS LD player tras la introducción del cambio, su estado retorne a completo.

De la misma forma, las reglas de visibilidad descritas en el modelo de comportamiento de la especificación seguirán siendo comprobadas después de modificar el atributo *invisible* de un elemento, por lo que éste no será mostrado al usuario si el valor de dicho atributo entra en conflicto con el estado de un elemento superior en la jerarquía de control definida por la especificación. El instructor debe pensar en los cambios introducidos por los *poke* como situados al mismo nivel en la jerarquía que los disparados por las directivas *hide* y *show* de IMS LD.

5.2. Evaluación por casos de uso

En esta sección se presentan un conjunto de casos de uso en los que se recogen diversas situaciones que pueden tener lugar a lo largo de un proceso de aprendizaje

soportado computacionalmente y supervisado por tutor. Estos casos de uso han sido confeccionados a lo largo del proceso de desarrollo del modelo conceptual de adaptación, sirviendo como base de un proceso iterativo de refinamiento en el que la **generalidad** y **capacidad expresiva** del modelo de adaptación era gradualmente validada.

En primer lugar, el modelo ha demostrado ser lo suficientemente **general** como para permitir llevar a cabo distintos tipos de adaptaciones y observaciones en las que se modifica o recupera información tanto sobre estructura del diseño de aprendizaje como sobre su comportamiento en ejecución. Estos diversos tipos de adaptaciones han sido agrupados en ocho casos de uso distintos:

- Añadir nuevas partes al proceso - Casos 3 y 4
- Eliminar partes del proceso - Caso 8
- Sustitución de elementos - Caso 1
- Modificar características y propiedades de los elementos - Casos 4, 6 y 7
- Modificar el comportamiento de ejecución - Caso 7

Por otra parte, la tabla 5.4 refleja los distintos casos de uso clasificados según su momento de definición y tipo de activación, que son posibles de describir por medio de los elementos del modelo.

Tabla 5.4: Casos de uso: momento de introducción de las adaptaciones y observaciones

Caso de uso	Momento de definición	Momento de activación
1	Ejecución	Por el instructor
2	Ejecución	Por el instructor
3	Ejecución	Por el instructor
4	Ejecución	Por evento
5	Publicación	Por evento
6	Publicación	Temporal
7	Ejecución	Por el instructor
8	Publicación	Por el instructor

En segundo lugar, la **capacidad expresiva** del modelo de adaptación queda reflejada en la tabla 5.5¹, en la cual se recogen las distintas características de las adaptaciones

¹El caso de uso número 2 no ha sido incluido ya que no se refiere a una adaptación sino a una observación

capturadas por cada caso de uso. Así, cada columna de la tabla especifica: si el caso de uso se corresponde con una adaptación que pudo ser prevista de manera previa al comienzo de la ejecución, el momento en que su introducción se llevó a cabo, si dicha introducción fue repetida en más un ocasión, si afectó a una, varias o a todas las instancias de una ejecución, si modificó la definición del proceso o su estado en ejecución, si el cambio tuvo carácter permanente, cuál fue el grado en el que afectó a la definición de los objetivos de aprendizaje marcados para el proceso, propósito del cambio y posibilidad de evaluar su éxito.

5.2.1. Estructura de los casos de uso

Cada uno de los casos de uso incluye:

- Una narrativa en la que se describe una particular problemática que puede tener lugar a lo largo del desarrollo de un proceso de aprendizaje soportado computacionalmente y supervisado por instructor.
- Principales actores que intervienen en la situación descrita en la narrativa.
- Beneficios concretos que aportará la resolución del problema.
- Precondiciones necesarias para la resolución del problema.
- Desencadenante de la situación.
- Pasos necesarios para solventar el problema empleando los elementos del modelo.
- Posibles variaciones del caso de uso y su solución.
- Ejemplo del caso descrito mediante XML. El código correspondiente a una variación se representa con un tipo de letra distinta y en cursiva.

5.2.2. Caso 1: Sustitución de recursos

Narrativa: Durante una particular ejecución de una UoL y a la hora de que los alumnos desarrollen una de las actividades programadas, una página Web que formaba parte del conjunto de recursos necesarios para el desarrollo de la actividad resulta no estar disponible. Con el fin de que los alumnos puedan continuar con el plan previsto, el instructor decide sustituir dicho recurso por otro distinto.

Tabla 5.5: Casos de uso: características capturadas de las adaptaciones

Caso	Predictibilidad	Tiempo de introducción	Frecuencia	Ámbito (instancias)	Tipo	Perdurab.	Mod. Objetivos	Propósito	Evaluabilidad
1	No	Ejecución	1	1 o más	Definición	No	Leves	Adaptativa	No
3	No	Ejecución	1	1 o más	Definición	Si	Moderada	Correctiva	Si
4	No	Ejecución	más de 1	1 o más	Definición	Si	Moderada	Perfectiva	No
5	Si	Publicación	más de 1	Todas	Estado	Si	Moderada	Perfectiva	No
6	Si	Publicación	1	Todas	Estado	Si	Leve	Adaptativa	No
7	No	Ejecución	1	1 o más	Definición + Estado	Si	Leve	Correctiva	Si
8	Si	Publicación	1	Todas	Definición	Si	Grave	Evolutiva	No

Actores primarios: Instructor, Alumno

Actores beneficiados:

- Instructor y Alumno - No es necesario interrumpir el proceso educativo y se puede continuar con el plan previsto.

Precondiciones: El instructor tiene permisos para la modificación del diseño.

Desencadenante: Uno de los recursos no se encuentra disponible.

Pasos:

1. El instructor detecta que uno de los recursos no se encuentra disponible.
2. El instructor diseña un *poke* para sustituir el recurso original por otro que sí está disponible. La acción del *poke* no será evaluada y se considera que no va a afectar a la adquisición de los objetivos de aprendizaje.
3. El instructor aplica el *poke* a todas las instancias en ejecución de la UoL.

Ejemplo:

- Fichero de comandos:

```
<Poke id='sustitución' desc='Sustitución de un recurso'>
  <actions>
    <modification>
      <IdElement-Ref ref='LD-Actividad1' />
      <Attribute-name name='activity-description' />
      <attribute-value identifierref='id1'
        identifier='nvoRecursoLocal' />
    </modification>
  </actions>
</Poke>
```

- Fichero de manifiesto:

```
<resource identifier='nvoRecurso' type='webcontent'
  ref='paginaLocal.html'>
  <file href='paginaLocal.html' />
</resource>
```

- Fichero de contenido:

`paginaLocal.html`

5.2.3. Caso 2: Recuperación de información sobre el curso

Narrativa: Durante el desarrollo del curso los resultados en los test de evaluación que los alumnos van realizando están resultando más bajos de lo esperado. El instructor necesita averiguar cuáles son las causas de estos resultados y si residen en un diseño incorrecto del curso o bien se deben a factores externos al mismo.

Actores primarios: Alumno, instructor.

Actores destinatarios:

- Instructor - Facilita la toma de decisiones adecuadas si resulta necesario intervenir en el desarrollo del proceso.
- Alumno - Puede obtener mejor soporte del instructor.

Precondiciones:

- El instructor tiene permisos para la observación del diseño.

Desencadenante: Se ha producido un problema y el instructor necesita conocer datos sobre el desarrollo del curso para identificar su causa.

Pasos:

1. El instructor necesita obtener datos sobre el desarrollo del curso que faciliten la identificación de las causas de los pobres resultados en los test de evaluación.
2. Para poder determinar si el problema reside en el diseño o es de carácter externo, necesita conocer primero si los alumnos han llevado a cabo las actividades especificadas en el curso. Para ello han debido visitar cada una de ellas, recuperar el material recomendado y pasar cierto tiempo desarrollándolas hasta completarlas. El instructor decide diseñar una serie de *peeks* que permitan corroborar el desarrollo de estas acciones.
3. El instructor diseña un conjunto de *peeks* para conocer si los alumnos han entrado en ciertas actividades donde se trataban los conceptos luego evaluados en los test. Los *peeks* son de tipo instantáneo, ya que la observación únicamente se efectuará una vez y no incluyen condiciones de activación ya que deben aplicarse de manera inmediata.

4. Con las mismas características construye otra serie de *peek* que utilizará para recuperar el tiempo que los estudiantes pasaron en las actividades mencionadas anteriormente.

Variaciones:

- A. Aplicación a varias instancias del curso.

1a El instructor necesita obtener datos acerca del desarrollo del curso llevado a cabo por ciertos alumnos concretos.

3a y 4a El instructor diseña los *peeks* necesarios para recuperar la información deseada y los aplica únicamente a ciertas instancias de la unidad de aprendizaje correspondientes a dichos alumnos.

Ejemplo:

- Fichero de comandos:

```
<Peek id='peek-Act1Acces' desc='Num. accesos Actividad1'
                                type='instant'>
```

```
<Peek-Command>
```

```
<IdElement ref='LD-Act1' />
```

```
<Characteristic>access-count</Characteristic>
```

```
</Peek-Command>
```

```
</Peek>
```

```
<Peek id='peek-Act2Complt' desc='Actividad2 completada(s/n)'
                                type='instant'>
```

```
<Peek-Command>
```

```
<IdElement ref='LD-Act2' />
```

```
<Attribute>Completed</Attribute>
```

```
</Peek-Command>
```

```
<target-UoL>
```

```
<UoL-ref ref='User1' />
```

```
</target-UoL>
```

```
</Peek>
```

5.2.4. Caso 3: Introducción de nuevos elementos

Narrativa: Durante el desarrollo del curso el instructor detecta en los estudiantes ciertos errores en la comprensión de diversos conceptos. Con el objeto de solucionar el problema decide modificar el diseño original del curso e incluir material auxiliar que clarifique y facilite la comprensión de dichos conceptos.

Actores primarios: Alumno, instructor

Actores destinatarios:

- Instructor y Alumno - No es necesario interrumpir el proceso educativo para introducir el material y se puede continuar con el plan previsto.

Precondiciones:

- La UoL incluye componentes y actividades cuyos resultados permiten extraer conclusiones acerca del conocimiento que el alumno va alcanzando a lo largo del curso.
- El instructor tiene permisos para llevar a cabo modificaciones en el diseño del proceso.

Desencadenante: El instructor necesita modificar el diseño original del curso para incluir nuevo material y solucionar un determinado problema.

Pasos:

1. Utilizando los medios proporcionados por la UoL original, el instructor detecta ciertos problemas en la comprensión de ciertos conceptos por parte de la mayoría de los alumnos.
2. El instructor decide introducir material de refuerzo extra, de tal forma que garantice la correcta asimilación de dichos conceptos antes de concluir la sección.
3. El instructor diseña un *poke* que permita introducir el nuevo material. Dicho *poke* incluye los objetivos de aprendizaje relacionados con los conceptos que se pretende reforzar en el apartado referencias.
4. El instructor aplica el *poke* al curso en ejecución.
5. El instructor informa a los alumnos de la modificación efectuada y sugiere el repaso del tema al que pertenece antes de continuar con el programa establecido para el curso.

Variaciones:

A. Aplicación a varias instancias del curso.

- 1a El instructor detecta diversos tipos de problemas en la comprensión de ciertos conceptos por parte de algunos de los alumnos.
- 3a El instructor diseña diversos *adaptation pokes* que incluyen material de refuerzo con el fin de solventar cada uno de los problemas de comprensión detectados. En el apartado relaciones cada *poke* incluye referencias a los objetivos de aprendizaje relacionados con los conceptos que se pretende reforzar.
- 4a El instructor aplica el *poke* correspondiente a cada instancia del curso para la cual se se ha detectado cada tipo problema.
- 5a El instructor informa a los alumnos cuyas instancias del curso han sido modificadas de la adaptación efectuada y sugiere el repaso del tema al que pertenece antes de continuar con el programa establecido para el curso.

Ejemplo:

- Fichero de comandos:

```
<Poke id='poke-entorno' desc='Entorno de applets
        con representaciones gráficas'>
```

```
<actions>
```

```
<insertion>
```

```
<IdElement ref='LD-Animations' />
```

```
<Parent-Element ref='LD-Algorithms' />
```

```
</insertion>
```

```
</actions>
```

```
<target-UoL>
```

```
<UoL-ref ref='User1' />
```

```
</target-UoL>
```

```

<evaluation>
  <expression>
    <relExpression>
      <relOperator operator='greaterThan' />
      <expression>
        <peekRef idRef="peek-Puntuacion">
          <self/>
        </peekRef>
      </expression>
      <expression>
        <basicExpression>
          <langString>5</langString>
        </basicExpression>
      </expression>
    </relExpression>
  </expression>
  <activation-condition>
    <On-End />
  </activation-condition>
</evaluation>

<relations>
  <Proc_Component-ref ref='e-Insertion' />
  <Proc_Component-ref ref='e-Deletion' />
  <Proc_Component-ref ref='e-Rotation' />
</relations>

</Poke>

```

- Fichero de manifiesto:

```

<imsld:environment identifier='LD-Animations'>
  <imsld:title>Animaciones de algoritmos</imsld:title>
  <imsld:learning-object identifier='Animaciones'>
    <imsld:item identifierref='R-animations'

```

```
        identifier='id-Anim1' />
    </imsld:learning-object>
</imsld:environment>

<resource identifier='R-animations' type='webcontent'
          ref='anima.html'>
    <file href='anima.html' />
</resource>
```

- Fichero de contenido:

```
anima.html
```

5.2.5. Caso 4: Sincronización entre instancias de distintos participantes de un mismo proceso

Narrativa: Un determinado curso está dividido en una primera parte compuesta por una serie de actividades que cada participante debe desarrollar en solitario y una segunda parte compuesta por otra serie de actividades que deben ser desarrolladas en común. Una vez la instrucción comienza, el instructor detecta que el ritmo de avance a través del mismo es muy diferente dependiendo de cada alumno, y mientras algunos de ellos han completado ya el primer grupo de actividades otros aún se encuentran en las iniciales. El instructor decide entonces incluir una serie de ejercicios extra a resolver por aquellos alumnos que van más adelantados, de forma que tengan tareas para realizar hasta que los más retrasados finalicen las suyas y la segunda parte del curso pueda comenzar.

Actores primarios: Alumno, instructor

Actores destinatarios:

- Diseñador e instructor- Pueden relacionar acciones entre instancias de distintos participantes de un mismo proceso.
- Instructor y Alumno - No es necesario interrumpir el proceso educativo para introducir el material y se puede continuar con el plan previsto.
- Alumno - El curso se adapta al ritmo de aprendizaje de cada alumno, ofreciendo siempre tareas para realizar y teniendo en cuenta la situación del resto de los participantes.

Precondiciones: El instructor tiene permisos para la modificación del diseño.

Desencadenante: Algunos alumnos no tienen actividades asignadas hasta que otros alumnos terminen las suyas.

Pasos:

1. El instructor detecta que hay alumnos que se encuentran sin actividades asignadas a la espera de que otros alumnos completen las suyas.
2. El instructor diseña un *peek* de tipo continuo que recupere el estado de la última actividad a realizar en solitario por cada alumno y lo aplica a todas las instancias.
3. El instructor diseña un *poke* para la introducción de los ejercicios extra que incluye una condición de activación de tal manera que la adaptación será aplicada a las instancias de aquellos alumnos que completen la última actividad a realizar en solitario, siempre y cuando queden todavía otros alumnos por finalizarla. Además el *poke* incluye una lista de objetivos de aprendizaje que pueden verse reforzados por el desarrollo de los ejercicios extra.

Variaciones:

- 3a El instructor diseña el *poke* sin condición de activación y lo va aplicando a ciertas instancias de la UoL que selecciona de manera manual.

Ejemplo:

- Ficheros de comandos:

```
<Peek id='peek-Estado-Act5' desc='Actividad5 completada(s/n)'
                                     type='continuous'>
```

```
<Peek-Command>
  <IdElement ref='LD-Act5' />
  <Attribute>Completion</Attribute>
```

```
</Peek-Command>
</Peek>
```

```
<Poke id='poke-ActExtra' desc='Introduce actividad extra'>
  <actions>
    <insertion>
```



```
        <IdElement ref='LD-Extra' />
        <Parent-Element ref='LD-Seccion1' />
    </insertion>
</actions>

<activation-condition>
    <expression>
        <logicExpression>
            <and>
                <expression>
                    <relExpression>
                        <relOperator operator='equalTo' />
                        <expression>
                            <peekRef idRef="peek-Estado-Act5">
                                <self/>
                            </peekRef>
                        </expression>
                    </relExpression>
                </expression>
                <expression>
                    <basicExpression>
                        <langString>>true</langString>
                    </basicExpression>
                </expression>
            </relExpression>
        </logicExpression>
    </expression>
    <expression>
        <relExpression>
            <relOperator operator='equalTo' />
            <expression>
                <peekRef idRef="peek-Estado-Act5">
                    </peekRef>
                </expression>
            </relExpression>
        </expression>
        <expression>
            <basicExpression>
                <langString>>false</langString>
            </basicExpression>
        </expression>
    </expression>
</activation-condition>
```

```

        </basicExpression>
      </expression>
    </relExpression>
  </expression>
</and>
</logicExpression>
</expression>
</activation-condition>

<relations>
  <LearningObjective-ref ref='LO-classConcept' />
  <LearningObjective-ref ref='LO-objectConcept' />
</relations>

</Poke>

```

- Fichero de manifiesto:

```

<imsld:learning-activity identifier='LD-Extra'>
  <imsld:title>Ejercicios Extra</imsld:title>
  <imsld:activity-description>
    <imsld:item identifierref='R-Extra-1'
      identifier='LD-Extra-1' />
  </imsld:activity-description>
</imsld:learning-activity>

<resource identifier='R-Extra-1' type='webcontent'
  ref='ejercicios.html'>
  <file href='ejercicios.html' />
</resource>

```

- Fichero de contenido:

`ejercicios.html`

5.2.6. Caso 5: Repetición del proceso o partes del proceso

Narrativa: Se dispone de un diseño de un curso compuesto por una serie de actividades en las que se va presentando diverso material educativo y que concluyen con un test de evaluación final. En pasadas instrucciones el instructor ha comprobado cómo ciertos alumnos completan el curso con una puntuación demasiado baja y no repiten el proceso ni consultan de nuevo el material. Por ello desea modificar el diseño del curso de tal manera que aquellos alumnos que no superen un determinado umbral en la puntuación deban desarrollar de nuevo la misma secuencia de actividades. De esta forma el proceso de aprendizaje no se considerará concluido hasta que la puntuación especificada sea superada.

Actores primarios: Alumno, instructor

Actores destinatarios:

- Instructor y diseñador - Se posibilita la implementación de bucles de actividades en EMLs que no proporcionan elementos para su especificación.

Precondiciones: Se dispone de una unidad de aprendizaje con la estructura necesaria para la implementación de este tipo de bucle: secuencia de actividades y test o mecanismo final de evaluación.

Desencadenante: El instructor desea implementar un bucle de acciones.

Pasos:

1. El instructor desea implementar un bucle de acciones.
2. El instructor diseña un *peek* de tipo continuo que observe el estado *completo* de la actividad asociada al test. El *peek* se aplicará a todas las instancias y no tiene asociada ninguna condición de activación.
3. A continuación diseña otro *peek* de las mismas características que permita recuperar el valor de la propiedad en la que se almacena la puntuación del test.
4. A continuación diseña un *poke* que indica al alumno que repita de nuevo una secuencia de actividades modificando para ello, en caso necesario, su visibilidad y estados. El *poke* se aplicará a todas las instancias en ejecución de la UoL teniendo como condición de activación el que el valor recuperado por el primer

peek introducido sea *completo* y que la puntuación recuperada del test sea menor que el umbral especificado.

Ejemplo:

- Ficheros de comandos:

```
<Peek id='peek-Estado-Act5' desc='Actividad5 completada(s/n)'
      type= 'continuous'>
```

```
<Peek-Command>
  <IdElement ref='LD-Act5' />
  <Attribute>Completion</Attribute>
</Peek-Command>
</Peek>
```

```
<Peek id='peek-Score' desc='Observa puntuación'
      type='continuous'>
```

```
<Peek-Command>
  <IdElement ref='LD-Prop-Score' />
  <Attribute>Value</Attribute>
</Peek-Command>
</Peek>
```

```
<Poke id='poke-Bucle' desc='Activa bucle de actividades'>
```

```
<actions>
  <modification>
    <IdElement-Ref ref='LD-Actividad1' />
    <attribute-name name='completion' />
    <attribute-value value = 'false' />
  </modification>
  <modification>
    <IdElement-Ref ref='LD-Actividad2' />
    <attribute-name name='completion' />
    <attribute-value value = 'false' />
  </modification>
  ...
```

```
<modification>
  <IdElement-Ref ref='LD-Actividad5' />
  <attribute-name name='completion' />
  <attribute-value value = 'false' />
</modification>
</actions>

<activation-condition>
  <expression>
    <logicExpression>
      <and>
        <expression>
          <relExpression>
            <relOperator operator='equalTo' />
            <expression>
              <peekRef idRef="peek-Estado-Act5">
                <self/>
              </peekRef>
            </expression>
            <expression>
              <basicExpression>
                <langString>true</langString>
              </basicExpression>
            </expression>
          </relExpression>
        </expression>
        <expression>
          <relExpression>
            <relOperator operator='lessThan' />
            <expression>
              <peekRef idRef="peek-Score">
                </peekRef>
            </expression>
          </expression>
        </expression>
      </and>
    </logicExpression>
  </expression>
</activation-condition>
```

```

        <basicExpression>
            <langString>5</langString>
        </basicExpression>
    </expression>
</relExpression>
</expression>
</and>
</logicExpression>
</expression>
</activation-condition>

</Poke>

```

5.2.7. Caso 6: Programación de las actividades de un curso

Narrativa: Se desea diseñar una UoL para un curso que estará compuesto por una serie de actividades que deberán ser completadas por los alumnos a lo largo de diez semanas, de tal manera que cada una de ellas se volverá disponible en una fecha determinada. El objetivo es diseñar la unidad de aprendizaje de manera que sea fácilmente reutilizable, para lo cual será necesario poder activar las actividades en fechas distintas a las previstas para su primera ejecución.

Actores primarios: Diseñador, Instructor

Actores destinatarios:

- Diseñador - No requiere modificar la unidad de aprendizaje para adaptarla a nuevas ejecuciones. El mismo diseño es válido para todas ellas.

Precondiciones:

Desencadenante: Se necesita diseñar una UoL con actividades que deben volverse disponibles en fechas concretas.

Pasos:

1. Se lleva a cabo un diseño inicial del curso en el cual todas las actividades son visibles y están activas.
2. Nada más publicar el curso se introduce un *poke* que desactive y oculte todas las actividades.

3. Se prepara un conjunto de *pokes*, de manera que cada uno de ellos tiene asociada una acción para hacer visible una actividad determinada. Estos *pokes* se aplicarán a todas las instancias de la UoL y no tienen asociadas evaluaciones ni relaciones, pero sí tienen definidas condiciones de activación de tipo temporal, de tal forma que a cada uno le corresponde una fecha distinta.
4. A medida que el tiempo transcurra, y dependiendo de su fecha asociada, los distintos *pokes* se irán activando, de manera que las distintas actividades sean presentadas a los alumnos.
5. Para futuras ejecuciones de la UoL únicamente será necesario modificar las fechas especificadas en los *pokes* para adaptarlas a la nueva ejecución, no siendo necesario introducir ningún cambio en el diseño original.

Ejemplo:

- Ficheros de comandos:

```
<Poke id='poke-Desactiva' desc='Desactiva todas
                                las actividades'>
  <actions>
    <modification>
      <IdElement-Ref ref='LD-Actividad1' />
      <attribute-name name = 'visible' />
      <attribute-value value = 'false' />
    </modification>
    <modification>
      <IdElement-Ref ref='LD-Actividad2' />
      <attribute-name name = 'visible' />
      <attribute-value value = 'false' />
    </modification>
    .....
  </actions>
</Poke>
```

```
<Poke id='poke-Activa-Act1' desc='Activa actividad 1'>
  <actions>
    <modification>
      <IdElement-Ref ref='LD-Actividad1' />
      <attribute-name name = 'visible' />
      <attribute-value value = 'true' />
    </modification>
  </actions>

  <activation-condition>
    <DateExpression>
      <Time>01/01/2007 00:00:00</Time>
    </DateExpression>
  </activation-condition>
</Poke>
```

```
<Poke id='poke-Activa-Act1' desc='Activa actividad 1'>
  <actions>
    <modification>
      <IdElement-Ref ref='LD-Actividad1' />
      <attribute-name name = 'visible' />
      <attribute-value value = 'true' />
    </modification>
  </actions>

  <activation-condition>
    <DateExpression>
      <Time>01/02/2007 00:00:00</Time>
    </DateExpression>
  </activation-condition>
</Poke>
```


5.2.8. Caso 7: Variación de características de elementos del curso

Narrativa: En un determinado curso los diseñadores han establecido un determinado tiempo límite para completar cierta actividad. Sin embargo, durante el desarrollo del mismo, los instructores detectan que ninguno de los participantes del proceso ha sido capaz de completar dicha actividad en el tiempo estipulado. La estimación inicial no ha sido por tanto la adecuada para el perfil de los alumnos y los instructores necesitan modificar el tiempo inicialmente asignado para que éstos puedan completar la actividad.

Actores primarios: Instructor, Alumno

Actores destinatarios:

- Instructor y Alumno - No es necesario interrumpir el proceso educativo para llevar cabo la depuración y se puede continuar con el plan previsto.

Precondiciones: El instructor tiene permisos para llevar a cabo modificaciones en el diseño del proceso.

Desencadenante: Se necesita modificar el tiempo asignado para el desarrollo de una actividad.

Pasos:

1. Los instructores detectan la necesidad de modificar el tiempo asignado para el desarrollo de cierta actividad.
2. Los instructores diseñan un *poke* cuyas acciones modifican por un lado el estado de la actividad, pasando de “completa” a “incompleta”, y por otro el tiempo originalmente asignado para que los alumnos la terminen. El *poke* será aplicado a todas las instancias de la UoL y no incluirá condición de activación ni objetivos de aprendizaje relacionados. Se considerará que la adaptación ha sido un éxito para una instancia si al completarse el nuevo tiempo asignado al test, la variable que almacena la puntuación obtenida ha sido cargada con un valor. Esta condición será especificada en la sección *evaluation* del *poke*, de tal forma que, si al finalizar el proceso el número de pokes con evaluación positiva supera un determinado umbral, se considerará que el nuevo tiempo asignado a la actividad es correcto.

Ejemplo:

- Ficheros de comandos:

```
<Peek id='peek-Estado-ActFinal' desc='Actividad
      Final completada (s/n)' type= 'continuous'>
  <Peek-Command>
    <IdElement ref='LD-Final' />
    <Attribute>Completion</Attribute>
  </Peek-Command>
</Peek>
```

```
<Peek id='peek-Score' desc='Observa puntuación'
      type= 'continuous'>
  <Peek-Command>
    <IdElement ref='LD-Prop-Score' />
    <Attribute>Value</Attribute>
  </Peek-Command>
</Peek>
```

```
<Poke id='poke-evalTime' desc='Ajuste tiempo
      de la evaluación'>
  <actions>
    <modification>
      <IdElement-Ref ref='LD-Final' />
      <attribute-name name='completion' />
      <attribute-value value='false' />
    </modification>
    <modification>
      <IdElement-Ref ref='LD-Final' />
      <attribute-name name='complete-activity' />
      <attribute-value ref='time-limit'
        value='PT45M' />
    </modification>
  </actions>
```

```
<evaluation>
  <activation-condition>
    <expression>
      <relExpression>
        <relOperator operator='equalTo' />
        <expression>
          <peekRef idRef="peek-Estado-ActFinal'">
            <self/>
          </peekRef>
        </expression>
        <expression>
          <basicExpression>
            <langString>true</langString>
          </basicExpression>
        </expression>
      </relExpression>
    </expression>
  </activation-condition>

  <expression>
    <relExpression>
      <relOperator operator='notEqualTo' />
      <expression>
        <peekRef idRef="peek-Score">
          <self/>
        </peekRef>
      </expression>
      <expression>
        <basicExpression>
          <langString>null</langString>
        </basicExpression>
      </expression>
    </relExpression>
  </expression>
</evaluation>
```

</evaluation>

</Poke>

5.2.9. Caso 8: Creación de versiones

Narrativa: Se dispone de una UoL que se emplea en la instrucción de una determinada materia impartida como parte de un determinado curso. Se desea reutilizar su diseño para emplearla dentro de cierto seminario, pero únicamente interesa utilizar aquellas actividades de tipo teórico, eliminando del plan de curso las prácticas y las evaluaciones.

Actores primarios: Diseñador, Instructor

Actores destinatarios:

- Diseñador e Instructor - Se puede generar una nueva versión del diseño original sin que sea necesario el mantenimiento de dos UoLs distintas.

Precondiciones: Se dispone de una UoL original.

Desencadenante: Surge la necesidad de crear una nueva versión de la UoL introduciendo ligeros cambios en el diseño.

Pasos:

1. Surge la necesidad de crear una nueva versión de una UoL eliminando las actividades prácticas y las evaluaciones.
2. El diseñador genera un *poke* que suprima las secciones que no son necesarias para el curso.
3. El instructor publica la UoL original junto con el *poke* que adapta el diseño.
4. Los alumnos actúan sobre la nueva versión de la UoL de igual forma que si el diseño original no hubiese incluido las secciones prácticas y evaluaciones.

Ejemplo:

- Ficheros de comandos:

```
<Poke id='poke-version2' desc='Versión UoL seminario'>
  <actions>
    <deletion>
      <IdElement-Ref ref='LD-Pract1' />
    </deletion>
    <deletion>
      <IdElement-Ref ref='LD-Pract2' />
    </deletion>
    <deletion>
      <IdElement-Ref ref='LD-Eval1' />
    </deletion>
    <deletion>
      <IdElement-Ref ref='LD-Eval2' />
    </deletion>
    ....
  </actions>
</Poke>
```

5.3. Aplicación a escenarios de aprendizaje reales

La factibilidad del método de diseño iterativo fue probada mediante su aplicación al desarrollo de tres experiencias educativas distintas que se correspondían con tres modalidades de educación diferentes, no-presencial, semi-presencial y presencial, y en las que, si bien el uso y grado de soporte computacional del proceso variaba, compartían la presencia de uno o más instructores a cargo de la supervisión del proceso. Esta prueba sirvió a su vez para completar la validación de la expresividad del modelo de adaptación y la factibilidad del mecanismo de implementación.

Teniendo en cuenta las problemáticas descritas en el capítulo tercero y para garantizar el éxito del proceso educativo, las pruebas de evaluación fueron efectuadas sobre réplicas en forma de UoL de los procesos de aprendizaje. Así, por cada proceso, se implementó una versión sobre Moodle, que sería utilizada por los participantes para llevar a cabo las distintas actividades propuestas, y otra mediante la especificación del IMD LD, que sería empleada para desarrollar las distintas fases del método de diseño iterativo y sobre la que se aplicarían las mismas adaptaciones y modificaciones que se

fuesen introduciendo en la versión en Moodle.

El desarrollo de esta prueba de evaluación persigue comprobar tanto la adecuación del método propuesto al desarrollo de un proceso de aprendizaje real, como la capacidad del modelo para describir mediante sus elementos las distintas adaptaciones que sea necesario introducir. Así, a lo largo de esta sección se detallarán las distintas adaptaciones que surgieron en los tres escenarios de evaluación y que en total capturan, aproximadamente, un 72 % de las características de las adaptaciones recogidas en las tablas 5.4 y 5.5 de la primera sección del capítulo. Se espera que este porcentaje sea incrementado en futuras aplicaciones del método en nuevos procesos de aprendizaje, a medida que adaptaciones de características y requisitos más inusuales tengan lugar.

A continuación se detalla el desarrollo de cada uno de los tres procesos de aprendizaje empleados en la evaluación. Por cada uno se describe, en primer lugar, el propio escenario de evaluación, especificando las características del curso y su estructura, y en segundo lugar, el desarrollo de cada una de las fases del método de diseño iterativo aplicado para el caso particular. La sección concluye con algunas conclusiones extraídas tras el análisis de las tres pruebas.

5.3.1. ESA6: Curso de formación continua en el empleo

El primero de los procesos de evaluación de la solución tuvo lugar dentro del marco del proyecto ESA6. Dicho proyecto, desarrollado en colaboración entre el Laboratorio DEI de la Universidad Carlos III de Madrid, y la Dirección General de Protección Civil y Emergencias, tenía como objetivo la creación de una herramienta que permitiese la automatización del proceso de recogida de informes, y la generación de estadísticas de las actuaciones de los Servicios de Extinción de Incendios y Salvamento. Como parte del mismo, y de forma previa a la implantación definitiva del sistema, era necesario impartir una serie de cursos de formación donde se diera a conocer el funcionamiento del sistema a los usuarios, y que a la vez permitiesen recopilar sus opiniones y sugerencias sobre los diversos aspectos de la solución.

Dada la dispersión geográfica y el gran número de usuarios que era necesario formar en la herramienta, se optó por una modalidad de curso no-presencial, de tal manera que cada alumno pudiese seguir el proceso y completar las actividades propuestas desde su puesto habitual de trabajo y en el horario más conveniente. El desarrollo del curso sería supervisado por un instructor que además proporcionaría soporte a los alumnos a través de diversos medios de comunicación como foros, chats, correo electrónico, etc.

La experiencia tuvo lugar entre el 5 y el 30 de Marzo del 2007 y el número total de participantes en el proceso fue de 132 alumnos.

5.3.1.1. Estructura del curso

El curso estaba formado por cuatro módulos distintos. El primero de ellos servía como introducción a la herramienta, describiendo sus características generales, objetivos, requisitos de software, etc, y los otros tres cubrían diferentes funcionalidades de la misma: gestión de usuarios, gestión de informes y consulta y generación de estadísticas. Si bien existían distintos perfiles de usuario, cada uno de los cuales hacía uso de una funcionalidad específica de la herramienta, se consideró apropiado que todos los usuarios completasen todos los módulos de tal forma que adquiriesen una visión general del funcionamiento del sistema. Cada módulo se encontraba dividido en una parte teórica, que describía el modo de uso de una funcionalidad a través de cierto material de lectura; una parte práctica, donde se proponía una serie de ejercicios para ejercitar el conocimiento adquirido; y un test de autoevaluación que servía para que el propio usuario obtuviese una medida de la comprensión alcanzada sobre los conceptos explicados en el módulo. Cada test tenía asignado un determinado tiempo límite para su realización, si bien podían ser repetidos un número indefinido de veces. El aspecto final del curso puede observarse en la captura de pantalla mostrada en la figura 5.2.

Además, una vez los cuatro módulos fuesen completados, el usuario debía realizar un test final de evaluación de toda la materia y se solicitaba rellenase un cuestionario con objeto de recoger su opinión tanto sobre la herramienta como sobre el propio curso de formación.

5.3.1.2. Desarrollo del proceso

Esta sección detalla el desarrollo de las distintas fases del método de diseño iterativo dentro del escenario previamente descrito.

FASE 1: Diseño inicial

En primer lugar se desarrolló la fase de diseño inicial del proceso, cuyo resultado fue la obtención de una primera versión de la UoL del curso junto con un *perfil de evaluaciones* que sería utilizado para estimar el grado de éxito del desarrollo del mismo.

Diseño de la UoL

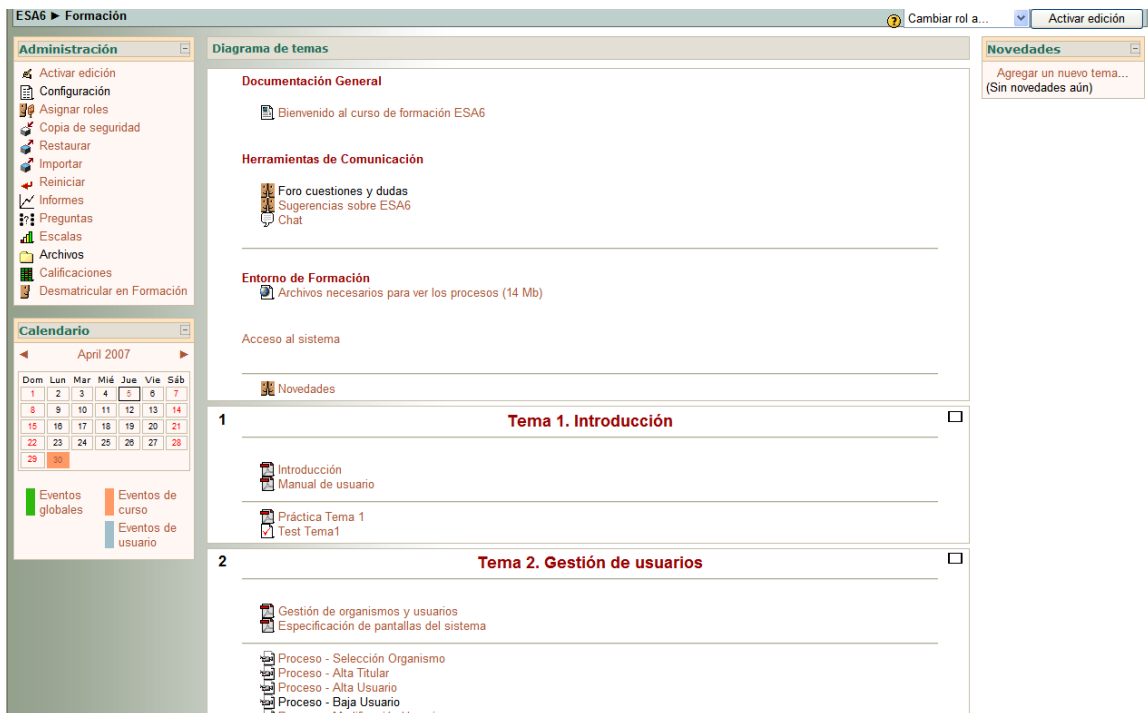


Figura 5.2: Imagen de la versión del proceso implementada en Moodle

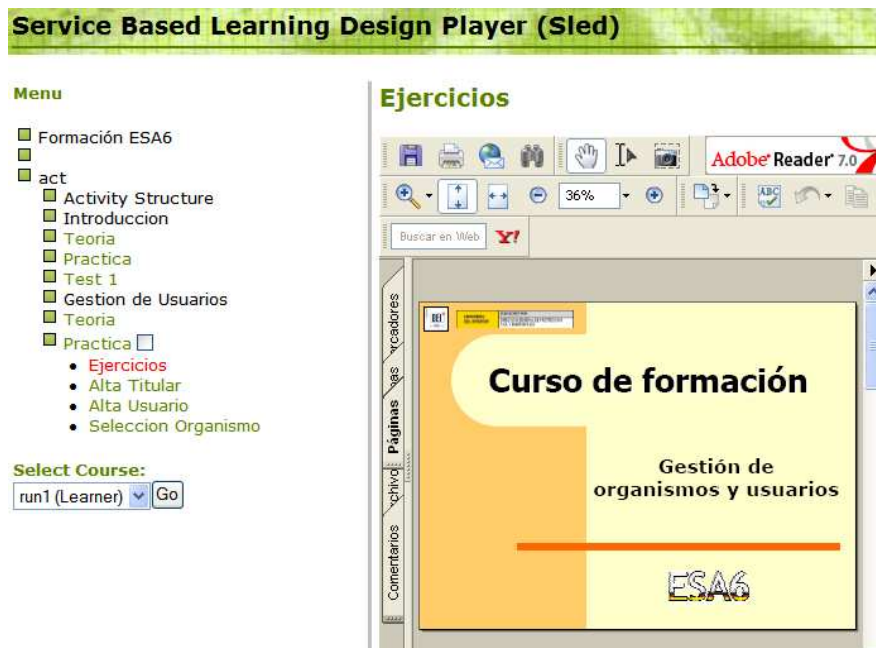


Figura 5.3: Imagen de la replica del proceso implementada mediante UoL

La implementación de la UoL se llevó a cabo utilizando la especificación del IMS LD y considerando dos roles distintos entre los participantes: *learner* o alumno y *tutor*. La visión que los primeros tenían del curso sería la de un único elemento *play*, que a su vez contendría un único acto o elemento *act* donde se organizarían los distintos módulos del curso organizados temáticamente, tal y como aparecían en la versión en Moodle. Dichos módulos fueron representados mediante elementos *activity-structure* que agrupaban tres actividades distintas: teoría, práctica y test. Haciendo uso de las posibilidades de secuenciación que ofrece el lenguaje, se obligaba al alumno a completar cada actividad y cada módulo antes de poder pasar al siguiente. A cada test se asignó el mismo tiempo límite de realización que en el curso real. Por su parte, el tutor o instructor únicamente tenía asignada como actividad el proveer soporte a los alumnos en el desarrollo de las distintas actividades, tarea que era soportada mediante un servicio de foro sobre el que todos los participantes del proceso tenían acceso.

La figura 5.3 muestra el aspecto de una ejecución de la UoL sobre el IMS LD Player Sled [86].

Definición del perfil de evaluaciones

A la hora de diseñar el perfil de evaluaciones para el curso de formación ESA6, se decidió considerar como único objetivo de aprendizaje la asimilación por parte del alumno de los conceptos básicos presentados en el curso. Dicho objetivo se consideraría alcanzado siempre y cuando la puntuación del test final de evaluación del curso fuese superior a seis. Para llevar a cabo la estimación de este objetivo, se definió un componente de evaluación simple o *plain-component* de tipo interno, que recuperaría el resultado del test por medio de la apropiada observación o *peek*. Además, se definieron cuatro componentes de evaluación de proceso o *proc-components* que representarían la actuación del alumno en cada tema del curso estimándola por medio de su último resultado en los respectivos test de auto-evaluación. Si bien estos componentes no serían empleados directamente en la valoración del objetivo de aprendizaje marcado para el curso, sí se encuentran relacionados con el mismo y cambios en su definición podrían afectar a la consecución del objetivo. Por otra parte, su inclusión podía ser de utilidad a la hora de definir posibles evaluaciones de futuras adaptaciones.

A continuación se incluye la descripción en XML del perfil de evaluaciones utilizado durante el proceso de aprendizaje:

```
<evaluationsProfile id='UoL-ESA6'>
```

```

<procComponents>
  <procComponent id='pc-Intro' desc='Actividad Intro'
    datatype='number' initialValue='0' >
    <idElement-Ref ref='Introducción' />
    <peek-ref id-ref='peek-autoscoreIntro'>
      <self />
    </peek-ref>
  </procComponent>
  <procComponent id='pc-Tema2' desc='Actividad Tema2'
    datatype='number' initialValue='0' >
    <idElement-Ref ref='Tema2' />
    <peek-ref id-ref='peek-autoscore1'>
      <self />
    </peek-ref>
  </procComponent>
  <procComponent id='pc-Tema3' desc='Actividad Tema3'
    .....
</procComponents>

<plain-Components>
  <internal-component id='i-TestFinal' desc='Resultado Test Final'
    datatype='number' initialValue='0'>
    <peek-ref id-ref='peek-TestFinal'>
      <self />
    </peek-ref>
  </internal-component>
</plain-Components>

<learning-objectives>
  <learning-objective id='objetivo1' desc='Conocer conceptos
    básicos ESA6' datatype='number' initialValue='0'>
    <evaluation>
      <relExpression>
        <relOperator operator='greaterThanOrEqual' />

```

```

        <BasicExpression>
            <internal-component-ref id-ref='i-TestFinal' />
        </BasicExpression>
        <BasicExpression>
            <langString>6</langString>
        </BasicExpression>
    </relExpression>
    <activation-condition>
        <on-end />
    </activation-condition>
</evaluation>
<relations>
    <proc_Component-ref ref='pc-Intro' />
    <proc_Component-ref ref='pc-Tema2' />
    <proc_Component-ref ref='pc-Tema3' />
    <proc_Component-ref ref='pc-Tema4' />
</relations>
</learning-objective>
</learning-objectives>
</evaluationsProfile>

```

FASES 2 y 3: Monitorización, desarrollo e introducción de adaptaciones

A lo largo del proceso fue necesario realizar distintas adaptaciones y monitorizaciones sobre la implantación del proceso soportada sobre Moodle. Las primeras se llevaron a cabo modificando directamente la estructura o contenidos utilizados en el curso, mientras que para las segundas se empleó el módulo auxiliar GISMO¹. Paralelamente, se trató de describir cada adaptación y observación realizada por medio de los elementos del modelo propuesto, de tal forma que se pudiese llevar a cabo una acción análoga sobre la versión del proceso especificada en la UoL. A continuación se detallan las distintas adaptaciones y observaciones que tuvieron lugar a lo largo del proceso, incluyendo el código del *poke* asociado de las más representativas.

¹GISMO es una herramienta gráfica para la monitorización y seguimiento del desarrollo de un curso. A partir de información extraída de la base de datos de Moodle genera diversos diagramas que facilitan el análisis y estudio de la actividad de los participantes del proceso. <http://gismo.sourceforge.net/>

A.1 Añadir un *learning object* a un entorno [Introducción de material complementario]:

A través de los foros y otros medios de comunicación del curso los alumnos plantearon diversas dudas acerca de algunos de los conceptos incluidos en el primer tema, por lo que se decidió incluir cierto material complementario con información acerca de navegadores y arquitecturas Web. Esta adaptación se tradujo, en el caso de la UoL, en la introducción de un nuevo *learning object* en el entorno de la actividad teórica del tema.

A.2 Añadir un nuevo entorno a una actividad [Introducción de material complementario]:

Debido a que algunos de los usuarios sufrieron ciertos problemas de conexión con la herramienta donde poner en práctica los ejercicios, se decidió incluir en el curso capturas de vídeo donde se mostraban los pasos que se deberían seguir para llevar a cabo la resolución de los mismos. En el caso de la UoL, esta adaptación se llevó a cabo introduciendo un nuevo entorno por cada actividad práctica, compuesto por un *learning object* por cada una de las capturas de vídeo correspondientes al tema.

A.3 Añadir un *learning object* a un entorno en una posición determinada [Introducción de material complementario]:

A petición de los usuarios que disponían de conexiones a baja velocidad, se incluyó por cada tema un archivo en formato comprimido en el que se agrupaban, para facilitar su descarga, todas las capturas de vídeo correspondientes a dicho tema. En el caso de la UoL, esta adaptación supuso la introducción en el nuevo entorno de un nuevo *learning object* asociado con dicho archivo. Dicho *learning object* debía aparecer el primero de todos los correspondientes al entorno. Con el objeto de llevar a cabo este cambio se definió el siguiente *poke*:

- Acción adaptativa:

```
<Poke id='poke-entornoCapturas2' desc='Introducir nuevo
                                fichero que agrupa capturas de video'>
<actions>
  <insertion>
    <IdElement-Ref ref='lo-Tema2_capZip' />
```

```

        <ParentElement-Ref ref='env-Tema2_cap' pos='1' />
    </insertion>
</actions>

</Poke>

```

- Descripción de nuevos elementos:

```

<imsld:learning-object identifier="lo-Tema2_capZip"
    isvisible="true" type="knowledge-object">
    <imsld:title>Capturas de video agrupadas</imsld:title>
    <imsld:item identifier="I-env-tema2_capZip"
        identifierref="res-env-tema02-capZip"
        isvisible="true" />
</imsld:learning-object>

<resource identifier="res-env-tema02-capZip"
    type="webcontent"
    href="tema02_todasCapturas.zip">
    <file href="tema02_todasCapturas.zip" />
</resource>

```

- Nuevo recurso:

```
tema02_todasCapturas.zip
```

A.4 Modificación de atributo de un elemento [*Sustitución de material*]:

A través de los foros y preguntas planteadas por los usuarios se detectó un error de comprensión en una de las preguntas del test de autoevaluación del tercer tema. Debido a ello se decidió sustituir el recurso original asociado al test por uno nuevo conteniendo dicha pregunta replanteada de forma distinta. Esta adaptación se tradujo, en el caso de la UoL, en la modificación de los atributos de la actividad correspondiente al test, concretamente en la modificación de la referencia al recurso a utilizar en el mismo. Con el objeto de llevar a cabo este cambio se definió el siguiente *poke*:

- Acción adaptativa:

```

<Poke id='poke-sustitucionRecurso' desc='Sustituir test'>

  <actions>
    <modification>
      <IdElement-Ref ref='I-test2' />
      <attribute-name>identifierref</attribute-name>
      <attribute-value>res-test2-nvo</attribute-value>
    </modification>
  </actions>

  <evaluation>
    <relExpression>
      <relOperator operator='greaterThanOrEqual' />
      <expression>
        <procComponent-Ref idRef="pc-Tema3" />
      </expression>
      <expression>
        <basicExpression>
          <langString>6</langString>
        </basicExpression>
      </expression>
    </relExpression>
  </evaluation>

</Poke>

```

- Descripción de nuevos elementos:

```

<resource identifier="res-test2-nvo"
  type="imsqti_item_xmlv2p0" href="test2-nvo.xml">
  <file href="test2-nuevo.xml" />
</resource>

```

- Nuevo recurso:

Fichero: test2-nvo.xml

O.1 Recuperación exitosa de valor de un atributo en ejecución [Observación]

A medida que algunos participantes iban terminando el proceso se observó que la puntuación de algunos test de evaluación era muy baja o nula. Con el fin de obtener más información acerca de las posibles causas de su bajo rendimiento se decidió recuperar información acerca del número de veces que habían accedido a cada recurso. La definición exacta del *peek* asociado a esta observación es la siguiente:

```
<Peek id='peek-CuentaAccesosInfPract' desc='Obtener número de
      accesos de un usuario a actividad'
      type='number' initialValue='unknown'
      peekType='instant'>
```

```
<Peek-Command>
  <IdElement ref='Tema3-pract' />
  <Attribute>Number-access</Attribute>
</Peek-Command>
```

```
<targetUoL>
  <UoL-ref ref='user3' />
</targetUoL>
```

```
</Peek>
```

O.2 Recuperación fallida de valor de un atributo en ejecución [Observación]

Con objeto de clarificar las causas del bajo rendimiento de algunos alumnos en los cuestionarios de evaluación se decidió observar el tiempo que cada uno de ellos había dedicado al desarrollo de cada actividad propuesta. Si bien fue posible obtener esta información sobre la implementación del proceso en Moodle, no fue posible recuperar su equivalente de la versión UoL debido a que el motor de ejecución utilizado para soportar el proceso, CopperCore, no almacena datos relativos a la duración de los accesos sobre sus elementos.

En caso de haber sido posible disponible disponer de dicha información el código del *peek* sería el que se describe a continuación. Obsérvese que, en este caso, se

ha sustituido la referencia a una instancia objetiva concreta por una condición de activación. De esta manera la observación será aplicada sobre todas las instancias para las cuales un *peek* previamente definido para recuperar el resultado del test, “peek-autoscore3”, haya obtenido puntuaciones menores de 5.

```

<Peek id='peek-TiempoActividad3' desc='Obtener tiempo
        dedicado a actividad3'
        type='number' initialValue='unknown'
        peekType='instant'>

  <Peek-Command>
    <IdElement ref='Tema3-pract' />
    <Attribute>Time</Attribute>
  </Peek-Command>

  <activation-condition>
    <expression>
      <relExpression>
        <relOperator operator='lessThan' />
        <basicExpression>
          <peek-ref id-ref='peek-autoscore3'>
            <self />
          </peek-ref>
        </basicExpression>
        <basicExpression>
          <langString>5</langString/>
        </basicExpression>
      </relExpression>
    </expression>
  </activation-condition>

</Peek>

```

A.5 Eliminar actividades [*Supresión de material*]

El análisis llevado a cabo sobre la información recuperada reveló que ciertos

usuarios, debido a falta de tiempo, únicamente completaban las actividades de contenido teórico y no llevaban a cabo ejercicios prácticos ni test de evaluación alguno. Por ello, y con el fin de no desfigurar los resultados de la evaluación del curso, se decidió presentar a los usuarios que así lo indicasen, una versión reducida del proceso que únicamente incluía contenidos teóricos y que no incorporaba el test de evaluación final. La definición exacta del *poke* asociado a este cambio es la siguiente:

- Acción adaptativa:

```
<Poke id='poke-VersionTeorica' desc='Crear una versión  
teórica del proceso'>
```

```
<actions>  
  <deletion>  
    <IdElement-Ref ref='test1' />  
  </deletion>  
  <deletion>  
    <IdElement-Ref ref='test2' />  
  </deletion>  
  <deletion>  
    <IdElement-Ref ref='test3' />  
  </deletion>  
  <deletion>  
    <IdElement-Ref ref='test4' />  
  </deletion>  
  <deletion>  
    <IdElement-Ref ref='intro-pract' />  
  </deletion>  
  <deletion>  
    <IdElement-Ref ref='tema2-pract' />  
  </deletion>  
  <deletion>  
    <IdElement-Ref ref='tema3-pract' />  
  </deletion>
```

```
        <deletion>
            <IdElement-Ref ref='tema4-pract' />
        </deletion>
    </actions>

    <targetUoL>
        <UoL-ref ref='user3' />
    </targetUoL>

</Poke>
```

FASES 5, 6 y 7: Evaluación de adaptaciones y resultados e integración

Una vez superada la fecha límite establecida para la finalización del proceso tuvo lugar la fase de evaluación de resultados. Durante esa fase se analizaron las causas, objetivos y consecuencias de las cinco adaptaciones que se aplicaron durante el desarrollo del proceso. Cuatro de esas cinco adaptaciones fueron incorporadas a petición de los propios participantes del proceso y tenían carácter correctivo o incorporaban una mejora sobre la definición original del proceso. Para una de ellas, la que implicaba la sustitución de cierto material didáctico, se definió además una fórmula de evaluación de dicha mejora que obtuvo un resultado satisfactorio para un 95% de los participantes. Asimismo, durante el análisis, se determinó que el origen de las adaptaciones no residía en el contexto actual de ejecución del proceso sino que por el contrario tenían claro carácter general, por lo cual las cuatro adaptaciones fueron seleccionadas para la fase de integración. La quinta adaptación, por su parte, permitiría generar una versión distinta del curso meramente teórica, por lo que fue convenientemente anotada y almacenada para permitir su aplicación en futuras ocasiones.

Durante la fase de integración se partió de la versión original de la UoL para ir aplicando una por una y en el mismo orden de introducción cada una de las adaptaciones seleccionadas. Para finalizar, se eliminaron aquellos contenidos y definiciones de recursos incluidos en la UoL pero que tras los cambios no eran referenciados por ningún elemento del proceso.

5.3.2. Máster: Curso de educación superior de postgrado

El segundo proceso de evaluación se llevo a cabo dentro del Master de Ingeniería de la Web desarrollado en colaboración entre la Universidad Carlos III de Madrid y la Universidad Francisco Gavidia de El Salvador. El master, con una duración total de 540 horas de formación, es impartido en modalidad a distancia, si bien cuenta con varias semanas de instrucción presencial durante las cuales, entre otras, se llevan a cabo las actividades de evaluación de los alumnos en las distintas asignaturas que componen el Master. De entre ellas, la asignatura “Implementación de sistemas Web” fue concretamente la utilizada para llevar a cabo el proceso de evaluación de la solución propuesta en el presente documento. Dicha asignatura fue impartida entre el 25 de febrero y el 25 de marzo de 2007, fechas entre las cuales los 26 alumnos matriculados deberían completar un conjunto de actividades teóricas y prácticas organizadas por semanas. Para ello, contaban con la asistencia de un tutor experto en la materia que estaría a cargo además de la dinamización del grupo. La plataforma elegida para el desarrollo del master fue Moodle, y la comunicación con el tutor se llevaría a cabo a través del correo electrónico y diversos foros creados para cada asignatura.

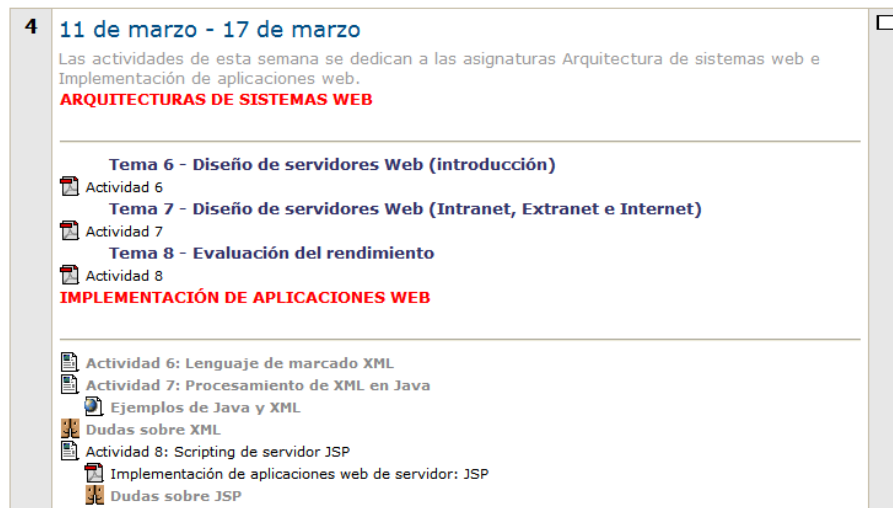


Figura 5.4: Captura de pantalla de la descripción de varias actividades para asignaturas del “Master de Ingeniería de la Web”

La figura 5.4 muestra una captura de pantalla de las actividades propuestas en una semana del curso para las asignaturas “Arquitecturas de sistemas Web” e “Implementación de sistemas Web”.

5.3.2.1. Estructura del curso

Cada asignatura del Master estaba compuesta por un conjunto de actividades de carácter teórico, acompañadas de diversos ejercicios donde poner en práctica el conocimiento adquirido y distintas pruebas de evaluación. Además, se proporcionaba una introducción videográfica a la asignatura y una guía docente donde se presentaba los objetivos e hitos a alcanzar, los requisitos previos necesarios y una planificación recomendada para el desarrollo de las actividades. También podían incluir diversas lecturas para complementar y ampliar los conceptos abordados, y pruebas de auto-evaluación que permitían al alumno medir el grado de conocimiento adquirido.

En el caso de la asignatura “Implementación de sistemas Web” se pretendía que el alumno obtuviese una visión general de las tecnologías fundamentales usadas en la capa de presentación de las aplicaciones Web a través de nueve actividades distintas descompuestas en diversas tareas teóricas y ejercicios. Cada actividad tenía asociada un foro distinto donde los alumnos planteaban sus dudas y discutían los conceptos abordados. La evaluación final del alumno se llevaría a cabo mediante un examen presencial y la entrega de varios ejercicios.

5.3.2.2. Desarrollo del proceso

Esta sección detalla el desarrollo de las distintas fases del método de diseño iterativo dentro del escenario de la asignatura “Implementación de sistemas Web”.

FASE 1: Diseño inicial

En primer lugar se desarrolló la fase de diseño inicial del proceso, cuyo resultado fue la obtención de una primera versión de la UoL del curso junto con un *perfil de evaluaciones* que sería utilizado para estimar el grado de éxito del desarrollo del mismo.

Diseño de la UoL

El diseño de la UoL para la asignatura “Implementación de sistemas Web” estaba compuesto por un único elemento *play* que a su vez contendría un único elemento *act*. Se consideraron dos roles distintos entre los participantes del proceso: el rol del alumno o *learner* y el del *tutor*. Al alumno se asignó la realización de una *activity-structure* compuesta a su vez de nueve actividades que se correspondían con las nueve descritas en la guía docente de la asignatura, y en cuyo entorno se encontraban asociados los recursos correspondientes a la guía docente, introducción videográfica y el foro general

de la asignatura. Por otra parte, cada una de las nueve actividades simples contenía una descripción de las tareas a realizar y un entorno asociado donde se proporcionaban los recursos necesarios para su desarrollo, así como lecturas complementarias y un foro donde discutir las dudas y problemas que pudieran surgir durante su realización. No se asignó ni tiempo límite para la realización de las actividades ni se incluyeron restricciones de navegación, de tal manera que quedaba a elección del propio alumno el seguir la planificación sugerida en la guía didáctica. Por su parte, el rol del tutor tenía asignado como única actividad el proporcionar soporte a los alumnos a lo largo de todo el proceso.

Definición del perfil de evaluaciones

Los objetivos marcados para el proceso de aprendizaje fueron cuatro:

- Conocer lenguajes de marcado y presentación recomendados por el W3C
- Aplicar las tecnologías involucradas en el desarrollo de aplicaciones web clásicas
- Capacitar al alumno para que pueda desarrollar aplicaciones Web dinámicas e interactivas utilizando tanto lenguajes de cliente como de servidor
- Aplicar herramientas de ayuda al desarrollo

Dichos objetivos serían evaluados a través de un examen que se desarrollaría una vez finalizado el curso de forma presencial y de una serie de entregas de ejercicios a lo largo del desarrollo del curso. Dichos ejercicios servirían, además, para supervisar el progreso de los alumnos y detectar posibles problemas.

El perfil de evaluaciones consideró por tanto los cuatro objetivos marcados. El primero y el segundo se encontraban íntimamente relacionados: la aplicación de la tecnología implica la comprensión de los lenguajes y dicha comprensión era necesaria para poder llevar a cabo la construcción de aplicaciones. Ambos objetivos serían estimados por medio de las entregas efectuadas por los alumnos a lo largo del curso, si bien los pesos asignados a la puntuación de cada entrega variarían de un objetivo a otro: mientras que en el primero se valoran más el resultado de los ejercicios de carácter menos práctico, en el segundo se tendrían más en cuenta los de aspecto más técnico. Por otra parte, el tercer objetivo que hacía referencia al desarrollo de la capacidad del alumno para el desarrollo de aplicaciones sería estimado por medio de un ejercicio escrito presencial en el que el alumno debería tomar y explicar diversas decisiones en cuanto al

diseño de ciertas aplicaciones Web. Por último, el cuarto objetivo hacía referencia al conocimiento desarrollado por el alumno en el uso de herramientas de ayuda al desarrollo. Para considerar este objetivo como alcanzado bastaba con que el alumno hubiese sido capaz de realizar alguna entrega y por tanto hacer uso de dichas herramientas.

Dado que tanto la puntuación obtenida por los alumnos en cada entrega, como la obtenida en el examen presencial, sería calculada por un tutor experto en la materia, se incluyeron en el perfil de evaluaciones cuatro componentes simples de tipo *tutor* que representasen cada valor. En caso de que dichos resultados hubiesen sido almacenados en un dispositivo externo de evaluaciones al que el motor de ejecución dispusiese acceso, igualmente hubiesen podido ser representados por elementos *external*, especificando la llamada a efectuar a dicho dispositivo para recuperar cada valor.

Por otra parte se definieron nueve componentes de proceso que representarían cada una de las actividades propuestas y se establecieron sus relaciones con cada uno de los objetivos de aprendizaje. La actuación del alumno para cada componente era evaluada empleando los resultados de las entregas, aunque hubo de tenerse en cuenta que en algunos casos una misma entrega estaba relacionada con varios componentes.

A continuación se incluye la descripción en XML del perfil de evaluaciones utilizado durante el proceso de aprendizaje:

```
<evaluationsProfile id='Master-ImplementacionWeb'>
  <procComponents>

    <procComponent id='pc-Act1' desc='Actividad 1: Netbeans'
      datatype='number' initialValue='0' >
      <idElement-Ref ref='Actividad 1' />
      <expression>
        <basicExpression>
          <langString>NULL</langString>
        </basicExpression>
        <logicExpression>
          <or>
            <relExpression>
              <relOperator operator='notEqualTo' />
              <basicExpression>
                <tutorComponent-Ref id-ref='t-EntregaAct3' />
              </basicExpression>
            </or>
          </logicExpression>
        </expression>
      </procComponent>
    </procComponents>
  </evaluationsProfile>
```

```

        </basicExpression>
        <basicExpression>
            <langString>-1</langString>
        </basicExpression>
    </relExpression>
    <logicExpression>
        <or>
            <relExpression>
                <relOperator operator='notEqualTo' />
                <basicExpression>
                    <tutorComponent-Ref id-ref='t-EntregaAct4' />
                </basicExpression>
                <basicExpression>
                    <langString>-1</langString>
                </basicExpression>
            </relExpression>
            .....
        </logicExpression>
    </or>
</logicExpression>

</expression>
</procComponent>

<procComponent id='pc-Act2' desc='Actividad 2: Lenguaje de
    marcas HTML' datatype='number'
    initialValue='0' >
    <idElement-Ref ref='Actividad 2' />
    <expression>
        <basicExpression>
            <tutorComponent-Ref id-ref='t-EntregaAct3' />
        </basicExpression>
    </expression>
</procComponent>

```

```
<procComponent id='pc-Act3' desc='Actividad 3: Hojas de
    estilo CSS' datatype='number'
    initialValue='0' >
  <idElement-Ref ref='Actividad 3' />
  <expresion>
    <basicExpresion>
      <tutorComponent-Ref id-ref='t-EntregaAct3' />
    </basicExpresion>
  </expresion>
</procComponent>
```

```
<procComponent id='pc-Act4' desc='Actividad 4: Lenguaje de
    scripting Javascript' datatype='number'
    initialValue='0' >
  <idElement-Ref ref='Actividad 4' />
  <expresion>
    <basicExpresion>
      <tutorComponent-Ref id-ref='t-EntregaAct4' />
    </basicExpresion>
  </expresion>
</procComponent>
```

```
<procComponent id='pc-Act5' desc='Actividad 5: Scripting de
    servidor CGI y PHP' datatype='number'
    initialValue='0' >
  <idElement-Ref ref='Actividad 5' />
  <expresion>
    <basicExpresion>
      <tutorComponent-Ref id-ref='t-EntregaAct5' />
    </basicExpresion>
  </expresion>
</procComponent>
```



```
<procComponent id='pc-Act6' desc='Actividad 6: Lenguaje de
    marcado XML' datatype='number'
    initialValue='0' >
    .....
</procComponents>

<plain-Components>
    <tutor-component id='t-EntregaAct3' desc='Entrega actividad 3'
        datatype='number' initialValue='-1'
        texto= 'Introduzca el resultado de la entrega
            correspondiente a la actividad 3' />
    <tutor-component id='t-EntregaAct4' desc='Entrega actividad 4'
        datatype='number' initialValue='-1'
        texto= 'Introduzca el resultado de la entrega
            correspondiente a la actividad 4' />
    <tutor-component id='t-EntregaAct5' desc='Entrega actividad 5'
        datatype='number' initialValue='-1'
        texto= 'Introduzca el resultado de la entrega
            correspondiente a la actividad 5' />
    <tutor-component id='t-EntregaAct9' desc='Entrega actividad 9'
        datatype='number' initialValue='-1'
        texto= 'Introduzca el resultado de la entrega
            correspondiente a la actividad 9' />
    <tutor-component id='i-EvalFinal' desc='Resultado examen final'
        datatype='number' initialValue='-1'
        texto= 'Introduzca el resultado de la
            evaluación final' />
</plain-Components>

<learning-objectives>
    <learning-objective id='L01' desc='Conocer lenguajes
        de marcado y presentación'
        datatype='number' initialValue='0'>
        <evaluation>
```

```

<expression>
  <mathExpression>
    <term>
      <factor>
        <basicExpression>
          <tutorComponent-ref ref='t-EntregaAct3' />
        </basicExpression>
      </factor>
      <mult>
        <basicExpression>
          <langString>0.4</langString>
        </basicExpression>
      </mult>
    </term>
    <sum>
      <term>
        <mathExpression>
          <term>
            <factor>
              <basicExpression>
                <tutorComponent-ref
                  ref='t-EntregaAct3' />
              </basicExpression>
            </factor>
            <mult>
              <basicExpression>
                <langString>0.4</langString>
              </basicExpression>
            </mult>
          </term>
          <sum>
            .....
          </sum>
        </mathExpression>
      </term>
    </sum>
  </mathExpression>

```

```

        <term>
        </sum>
    </mathExpression>
</expression>
</evaluation>
<relations>
    <procComponent-ref ref='pc-Act2' />
    <procComponent-ref ref='pc-Act3' />
    <procComponent-ref ref='pc-Act4' />
    <procComponent-ref ref='pc-Act5' />
    <procComponent-ref ref='pc-Act6' />
    <procComponent-ref ref='pc-Act7' />
</relations>
</learning-objective>

<learning-objective id='L02' desc='Aplicar tecnologías
    involucradas' datatype='number' initialValue='0'>
    <evaluation>
        <expression>
            <mathExpression>
                <term>
                    <factor>
                        <basicExpression>
                            <tutorComponent-ref ref='t-EntregaAct3' />
                        </basicExpression>
                    </factor>
                    <mult>
                        <basicExpression>
                            <langString>0.1</langString>
                        </basicExpression>
                    </mult>
                </term>
            </mathExpression>
        </expression>
    </evaluation>
</learning-objective>

```

```

    <mathExpression>
      <term>
        <factor>
          <basicExpression>
            <tutorComponent-ref
              ref='t-EntregaAct3' />
          </basicExpression>
        </factor>
        <mult>
          <basicExpression>
            <langString>0.1</langString>
          </basicExpression>
        </mult>
      </term>
    </sum>
    .....
  </sum>
  <mathExpression><
    <term>
      </sum>
    </mathExpression>
  </expression>
</evaluation>
<relations>
  <learningObjective-ref ref='L01' />
  <procComponent-ref ref='pc-Act9' />
</relations>
</learning-objective>

<learning-objective id='L03' desc='Capacitar desarrollo
  aplicaciones dinámicas e interactivas'
  datatype='number' initialValue='0'>
  <evaluation>
    <expression>

```

```

        <basicExpresion>
            <tutorComponent-ref ref='i-Examen' />
        </basicExpresion>
    </expresion>
</evaluation>
<relations>
    <learningObjective-ref ref='L01' />
    <learningObjective-ref ref='L02' />
</relations>
</learning-objective>

<learning-objective id='L04' desc='Conocer
                    herramientas de ayuda al desarrollo'
                    datatype='number' initialValue='0'>
    <evaluation>
        <expresion>
            <basicExpresion>
                <procComponent-ref ref='pc-Act1' />
            </basicExpresion>
        </expresion>
    </evaluation>
    <relations>
        <procComponent-ref ref='pc-Act1' />
    </relations>
</learning-objective>

</learning-objectives>
</evaluationsProfile>

```

FASES 2 y 3: Monitorización, desarrollo e introducción de adaptaciones

Al igual que en el escenario anterior, las adaptaciones y monitorizaciones realizadas sobre la plataforma Moodle, bien usando directamente los elementos provistos por la misma o bien utilizando el módulo auxiliar GISMO, fueron traducidas a su correspondiente representación mediante elementos del modelo propuesto en el presente docu-

mento. Esta sección describe cada una de ellas incluyendo el código correspondiente a los *pokes* y *peeks* que generaron.

A.6 Modificación de atributo e introducción de material alternativo [*Sustitución de material*]:

Para la realización de la actividad 6, correspondiente al lenguaje de marcado XML, se proporcionaba al alumno varios enlaces a diversos tutoriales en la red que resultaron no encontrarse disponibles durante el desarrollo del curso. Como resultado, se hubo de publicar material alternativo y modificar la descripción de la actividad. Para llevar a cabo esta adaptación sobre la versión del proceso implementada en la UoL, fue necesario introducir nuevos ficheros que correspondían por una parte a los nuevos contenidos y por otra a la nueva descripción de la actividad. Los ficheros fueron añadidos dentro del entorno correspondiente a la actividad 7, íntimamente relacionada con la actividad 6. Por otra parte fue necesario modificar los atributos de esta última para que referenciasen al recurso correspondiente a la nueva descripción.

La adaptación efectuada entraría dentro del grupo de los *pokes de reajuste*, es decir, se trata de una rectificación llevada a cabo debido a causas externas al propio desarrollo del proceso por lo que no se incluirá una evaluación que estime su éxito. En cualquier caso, el uso de un recurso distinto al planificado a la hora de desarrollar la actividad puede influir en la eficacia de la misma, más aún si tenemos en cuenta la premura con la cual, dada la urgencia del problema, se preparó el material sustitutivo. Por tanto, se definieron relaciones con los componentes del proceso afectados por la adaptación, concretamente los que representan las actividades 6 y la 7, los cuales a su vez se encuentran relacionadas con los objetivos 1, 2 y 3. Dichas relaciones serán tenidas en cuenta al analizar los resultados durante la fase de evaluación del proceso.

La definición exacta del *poke* asociado a esta adaptación es la siguiente:

- Acción adaptativa:

```
<Poke id='poke-recursosNoDisponibles'
      desc='Modifica descripción y recursos actividad 6'>

  <actions>
    <modification>
      <IdElement-Ref ref='Actividad 6' />
```

```

        <attribute-name>identifierref</attribute-name>
        <attribute-value>res-descAct6-nvo</attribute-value>
    </modification>

    <insertion>
        <IdElement-Ref ref='lo-Tema7-ejemplos' />
        <ParentElement-Ref ref='env-Actividad7' pos='1' />
    </insertion>
</actions>

<relations>
    <procComponent-ref id-ref='pc-Act6' />
    <procComponent-ref id-ref='pc-Act7' />
</relations>

</Poke>

```

- Descripción de nuevos elementos:

```

<imsld:learning-object identifier="lo-Tema7-ejemplos"
    isvisible="true" type="knowledge-object">
    <imsld:title>Ejemplos Java y XML</imsld:title>
    <imsld:item identifier="I-env-tema7_ejemplos"
        identifierref="res-ejemplosJavaXML"
        isvisible="true" />
</imsld:learning-object>

<resource identifier="res-descAct6-nvo-I"
    type="webcontent"
    href="desc-Act6-II.html">
    <file href="desc-Act6-II.html" />
</resource>

<resource identifier="res-ejemplosJavaXML"

```

```
        type="webcontent"
        href="javaxml.jar">
    <file href="javaxml.jar" />
</resource>
```

- Nuevos recursos:

Fichero con la nueva descripción de la actividad 6:

desc-Act6-II.html

Fichero con los nuevos recursos a utilizar:

javaxml.jar

A.7 Modificación de atributo [*Sustitución de material*]:

Para garantizar la efectividad del cambio introducido el tutor monitorizó de cerca el desarrollo de la actividad comprobando que no se produjesen problemas con los nuevos recursos introducidos. Al hacerlo detectó que, si bien los alumnos no habían parado de interactuar con la plataforma, no avanzaban ni completaban la actividad a pesar de que el tiempo estimado para su realización había sido sobrepasado ampliamente. Tras ponerse en contacto con ellos se identificó un problema en la interpretación de uno de los enunciados que implicaba la realización de una tarea mucho más laboriosa de la originalmente propuesta. Fue por tanto necesario modificar de nuevo la descripción de la actividad para subsanar el malentendido. Esta adaptación supuso la introducción de un nuevo recurso en la UoL correspondiente a la nueva descripción de la actividad y la modificación del atributo que la referencia.

La adaptación efectuada entraría dentro del grupo de los *pokes correctivos*, es decir, trata de dar solución a un problema derivado del diseño y definición del proceso. Para evaluar si la adaptación reparó realmente el problema se podía continuar monitorizando el tiempo invertido en el desarrollo de la actividad comparándolo con el estimado. En cualquier caso, a la hora de realizar el diseño del *poke* no se consideró la definición de una evaluación debido a la imposibilidad de recuperar dicha información del *LD player* utilizado, CopperCore. En cualquier caso se consideró que el nuevo enunciado no daba lugar alguno a ambigüedades.

La definición exacta del *poke* asociado a esta adaptación es la siguiente:

- Acción adaptativa:

```

<Poke id='poke-enunciadoConfuso'
      desc='Modifica descripción actividad 6'>

  <actions>
    <modification>
      <IdElement-Ref ref='Actividad 6' />
      <attribute-name>identifierref</attribute-name>
      <attribute-value>res-descAct6-nvo-II</attribute-value>
    </modification>
  </actions>

  <relations>
    <procComponent-ref id-ref='pc-Act6' />
    <procComponent-ref id-ref='pc-Act7' />
  </relations>

</Poke>

```

- Descripción de nuevos elementos:

```

<resource identifier="res-descAct6-nvo-II"
          type="webcontent"
          href="desc-Act6-III.html">
  <file href="desc-Act6-III.html" />
</resource>

```

- Nuevos recursos:

Fichero con la nueva descripción de la actividad 6:
desc-Act6-III.html

O.3 Recuperación fallida de valor de un atributo en ejecución [*Observación*]:

Al igual que en el escenario de evaluación anterior, no fue posible llevar a cabo la observación del tiempo dedicado a la actividad sobre la versión del proceso

implementada en la UoL ya que el motor de ejecución, CopperCore, no almacena dicha información sobre el proceso.

A.8 **Añadir servicio según condición basada en información de naturaleza externa** [*Añadir un servicio*]:

A la hora de llevar a cabo el desarrollo de los ejercicios planteados los alumnos debían utilizar instalaciones efectuadas en asignaturas previas del Master. Dichas instalaciones se llevaron a cabo en las máquinas de cada alumno haciendo uso de distintos sistemas operativos, concretamente Linux y Windows. Si bien hasta el momento no habían surgido problemas derivados del uso de una u otra herramienta, a lo largo del desarrollo de esta asignatura fue necesario configurar ciertos aspectos de las instalaciones efectuadas, como por ejemplo PHP, que ocasionaron un mayor número de problemas a los usuarios de Windows. Dada la gran cantidad de tiempo ocupada en la resolución de problemas de configuración y el número de mensajes en foros referentes a este tema, se añadió un servicio de Chat con el fin de agilizar la comunicación entre alumnos e instructores. Al traducir esta adaptación a la versión del proceso en UoL sería posible especificar una condición de activación basada en la elección del sistema operativo efectuada, información que podría ser almacenada en el “portafolio” del alumno. Así, los usuarios que hubiesen escogido el sistema operativo Windows encontrarían disponible a la hora de seguir la asignatura un servicio de Chat donde plantear sus dudas acerca de la configuración de los aspectos necesarios para la ejecución de los ejercicios planteados.

En cualquier caso, la especificación de IMS LD no permite especificar actualmente servicios de comunicación síncronos de tipo Chat. La única manera de llevar a cabo una adaptación análoga a la realizada en Moodle es hacer uso de un servicio creado y gestionado de manera externa y al que se haga referencia mediante un enlace. Dicho recurso se añadió como parte del entorno de actividad principal del proceso que agrupa al resto. Dado que el objetivo del Chat era la resolución de dudas relacionadas con la instalación del software, y no el abordar cuestiones relativas al temario, no se especificó ninguna evaluación para la adaptación que hiciese uso de la información recuperada sobre el desarrollo del proceso.

La definición exacta del *poke* asociado a esta adaptación es la siguiente:

- Acción adaptativa:

```

<Poke id='poke-añadirChat'
      desc='Añade un servicio de Chat según S.O. utilizado'>

  <actions>
    <insertion>
      <IdElement-Ref ref='lo-chat' />
      <ParentElement-Ref ref='env-ActividadPpal' />
    </insertion>
  </actions>

  <activation-condition>
    <expression>
      <relExpression>
        <relOperator operator='notEqualTo' />
        <basicExpression>
          <externalComponent-ref ref='e-tipoSO' />
        </basicExpression>
        <basicExpression>
          <langString>Linux</langString/>
        </basicExpression>
      </relExpression>
    </expression>
  </activation-condition>

</Poke>

```

- Descripción de nuevos elementos:

En el perfil de evaluaciones:

```

<external-component id='e-tipoSO' desc='Tipo S.O. alumno'
                    datatype='text' initialValue='Desconocido'
                    comando='Obtener-tipo-de-SO-alumno'>
  <activation-condition>
    <on-start />

```

```

    </activation-condition>
  </external-component/>

```

En el manifiesto:

```

<imsld:learning-object identifier="lo-chat"
    isvisible="true" type="knowledge-object">
  <imsld:title>Imagen del software</imsld:title>
  <imsld:item identifier="I-env-chat"
    identifieref="res-link-Chat"
    isvisible="true" />
</imsld:learning-object>

<resource identifier="res-link-Chat"
    type="webcontent"
    href="link-Chat.html">
  <file href="link-Chat.html" />
</resource>

</resource>

```

FASES 5, 6 y 7: Evaluación de adaptaciones y resultados e integración

Una vez concluido el proceso tuvo lugar la evaluación del éxito del diseño. En primer lugar se tuvieron en cuenta los resultados obtenidos por los alumnos observándose que, del total de alumnos matriculados, un 90 % superó las pruebas de evaluación propuestas. Se estimó que el margen de suspenso entraba dentro de lo razonable y que por tanto la dificultad de las pruebas de evaluación podía considerarse correctamente calibrada. A continuación se analizaron las dudas y cuestiones planteadas por los participantes a lo largo del desarrollo del curso, advirtiéndose que un gran número de ellas se encontraban relacionadas con la instalación del software necesario para el desarrollo de las prácticas. Por ello, se decidió que para futuros cursos, sería conveniente proveer a los alumnos con un archivo-imagen de la instalación necesaria para poder desarrollar las actividades propuestas. Para llevar a cabo esta adaptación se diseñó un último *poke* que no incluía

ninguna evaluación, puesto que sería automáticamente seleccionado para su integración, y que suponía la introducción, como parte del entorno de la actividad principal, de un nuevo *learning-object* con el fichero de la imagen para su descarga. Además, también supuso la modificación de la descripción de dicha actividad para explicar a los alumnos el proceso de instalación de la imagen.

A.9 Añadir un learning object a un entorno [*Introducir un recurso*]:

- Acción adaptativa:

```
<Poke id='poke-añadirImagenSoftware'
      desc='Añade un recurso conteniendo la imagen del software'>

  <actions>
    <insertion>
      <IdElement-Ref ref='lo-imagen' />
      <ParentElement-Ref ref='env-ActividadPpal' />
    </insertion>
  </actions>

</Poke>
```

- Descripción de nuevos elementos:

```
<imsld:learning-object identifier="lo-imagen"
                       isvisible="true" type="knowledge-object">
  <imsld:title>Imagen software</imsld:title>
  <imsld:item identifier="I-env-imagen"
              identifierref="res-imagen"
              isvisible="true" />
</imsld:learning-object>

<resource identifier="res-imagen"
           type="webcontent">
```

```
        href="imagenSoft.rar">
    <file href="imagenSoft.rar" />
</resource>

</resource>
```

A continuación, se analizaron las adaptaciones introducidas a lo largo del desarrollo del proceso. El éxito de la primera adaptación (A.6) no fue estimado ya que, como se explicó, no se definió evaluación alguna para la misma. En cualquier caso se estudiaron los resultados de las actividades 6 y 7, las afectadas por el cambio, y se consideraron satisfactorios. Aún así, el propio instructor mostró su preferencia por la utilización de los recursos originales, por lo cual la adaptación no fue seleccionada para su integración final. La adaptación fue convenientemente anotada y almacenada de tal forma que, si en futuras ejecuciones del curso se volviese a producir la situación, el instructor contase con una adaptación cuya eficacia ha sido ya demostrada.

En cuanto a la siguiente adaptación (A.7), dado que suponía una solución clara a un problema detectado en la definición del proceso, fue directamente seleccionada para integración. Además, durante su análisis, se observó que llevaba definida una relación con el componente de proceso que representaba la actividad 6, para el cual, a su vez, se había definido otra adaptación (A.6). Así pues, se estudiaron posibles vínculos entre las adaptaciones y se observó que la A.7 introducía una modificación sobre un recurso utilizado en la A.6. Como resultado ambas adaptaciones fueron combinadas de tal forma que la A.6, la guardada como alternativa cuando los recursos de la actividad 6 no estuviesen disponibles, hiciese uso del recurso introducido por la A.7, y ésta, la seleccionada para integración, proporcionase una descripción de la actividad que no incluyese los cambios introducidos por la primera.

En tercer lugar, y para terminar con la evaluación, se analizó la última de las adaptaciones introducidas (A.8), la inclusión del Chat. Esta adaptación fue desestimada para integración dado que surgió como respuesta a la necesidad de responder las dudas planteadas acerca de la instalación del software, y a que este problema sería salvado en futuras ejecuciones gracias al software introducido en la adaptación A.9.

Por último, y para terminar con el ciclo del proceso iterativo de diseño, tuvo lugar la fase de integración. Durante la misma, las dos adaptaciones seleccionadas para formar parte de la nueva versión del proceso (A.7 y A.9) fueron aplicadas, resultando en la introducción de un nuevo *learning-object* en el entorno de la actividad principal, así

como en la modificación de la descripción de esta última y de la actividad sexta. En caso de haber sido seleccionada para integración la adaptación A.8, la correspondiente al chat, hubiese sido necesario introducir propiedades y condiciones que capturasen la condición de activación de la misma. Tras la introducción de las modificaciones, la nueva definición del proceso fue, como de costumbre, validada para garantizar su conformidad con la especificación del IMS LD, y para terminar, se eliminaron aquellos recursos y definiciones de elementos antiguos que, tras los cambios, no serían presentados al usuario, concretamente el recurso originalmente utilizado para la descripción de la actividad 6, así como el fichero empleado en el mismo.

5.3.3. EAO: Curso de educación superior de grado

El tercer escenario de evaluación de la solución propuesta tuvo lugar dentro del marco de la asignatura Educación Asistida por Ordenador (EAO). Dicha asignatura, ofertada como optativa o de libre elección por la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, fue impartida en modo presencial durante el segundo cuatrimestre del curso 2005/2006. Su objetivo era conseguir que el estudiante fuese capaz, al finalizar el curso, de usar e implementar soluciones de tele-educación en diferentes entornos, tanto académicos como empresariales. La plataforma Moodle fue utilizada como complemento y soporte a la actividades desarrolladas en las clases magistrales y en las prácticas en el laboratorio, sirviendo tanto para poner a disposición de los alumnos el material necesario para el desarrollo de las mismas, como herramienta de comunicación entre alumnos e instructores. El número total de alumnos matriculados fue de 40.

5.3.3.1. Estructura del curso

La duración total del curso fue de catorce semanas. Durante las nueve primeras se abordaron los principales estándares de la industria de eLearning, explicando mediante clases magistrales los fundamentos teóricos de cada uno de ellos y poniéndolos en práctica a través de diferentes ejemplos y ejercicios. Las cinco últimas se dedicaron al desarrollo en grupo de proyectos para la implementación de herramientas que hiciesen uso de alguna de las funcionalidades vistas. La nota final del alumno en la asignatura se calcularía a partir de la nota obtenida en un test sobre sus conocimientos teóricos, la alcanzada en diversas entregas de prácticas y la del proyecto.

A la hora de realizar el diseño del curso se definió, en primer lugar, una programa-

ción inicial de los contenidos según las relaciones y requisitos existentes entre ellos. A continuación se configuró un curso en Moodle organizado por semanas, de tal forma que al comienzo de cada semana los instructores fueran publicando en él los recursos didácticos relacionados con el contenido teórico visto en clase, así como los enunciados y recursos necesarios para el desarrollo de las prácticas. La publicación de contenidos y prácticas semanalmente permitía introducir cierta flexibilidad en el desarrollo del programa, variando el número de entregas inicialmente previstas o desarrollando distintas propuestas de ejercicios y proyectos según la evolución del curso y los distintos perfiles de los alumnos matriculados.

El aspecto final de la implementación del curso sobre la plataforma Moodle puede observarse en la captura de pantalla mostrada en la figura 5.5.

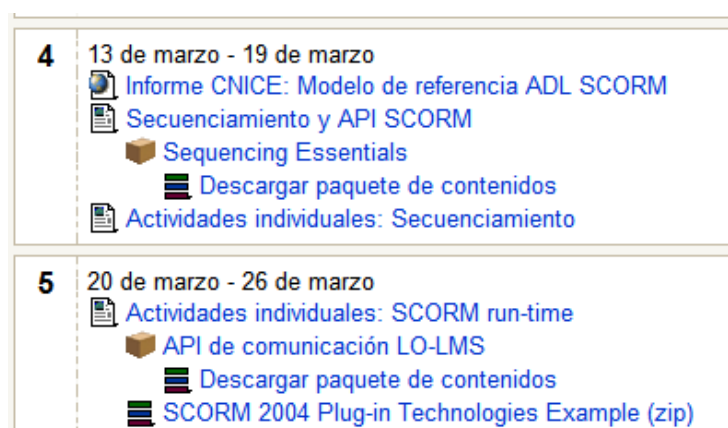


Figura 5.5: Imagen de la versión del proceso implementada en Moodle

5.3.3.2. Desarrollo del proceso

Esta sección detalla el desarrollo de las distintas fases del método de diseño iterativo dentro del escenario de la asignatura de “Enseñanza Asistida por Ordenador”.

FASE 1: Diseño inicial

En primer lugar se desarrolló la fase de diseño inicial del proceso, cuyo resultado fue la obtención de una primera versión de la UoL del curso junto con un *perfil de evaluaciones* que sería utilizado para estimar el grado de éxito del desarrollo del mismo.

Diseño de la UoL

Una vez más la implementación del proceso mediante IMS LD consideró dos roles distintos, *learner* y *tutor*, un único *play*, un único elemento *act* y una *activity-structure* principal que incluía el foro de la asignatura, la planificación inicial y presentación

de la misma, y donde se organizarían las actividades y materiales correspondientes a cada semana del curso. Así, al comienzo de cada semana, se publicaría mediante la correspondiente adaptación una nueva *activity-structure* que incluiría dichos materiales y que habría sido desarrollada por los instructores la semana anterior teniendo en cuenta la evolución del curso y las necesidades observadas.

Definición del perfil de evaluaciones

A la hora de diseñar el perfil de evaluaciones para la asignatura de “Enseñanza Asistida por Ordenador”, se decidió considerar dos objetivos de aprendizaje distintos:

- Asimilar los conceptos teóricos relacionados con los principales estándares de eLearning.
- Desarrollar la capacidad de aplicar dichos estándares en el desarrollo de herramientas de soporte a la educación.

De cara a la hora de establecer la nota final del alumno en la asignatura estos objetivos tendrían un peso de un 30 % y un 70 % respectivamente.

Teniendo en cuenta esto, se definió un perfil de evaluaciones que incluyó ambos objetivos de aprendizaje. Para estimar el primero, únicamente se utilizaría la nota obtenida en el examen teórico, la cual proporcionaba una valoración conjunta del conocimiento alcanzado por el alumno sobre todos los estándares vistos en el curso. En cuanto al conocimiento práctico, se considerarían las notas obtenidas en las entregas realizadas en clase y la nota final del proyecto, en una relación de un 40 y un 60 % por ciento respectivamente.

Se definieron tres componentes de proceso, uno que representase el conocimiento teórico, otro el desarrollo de los ejercicios prácticos y otro el desarrollo del proyecto. Igualmente se definieron dos componentes simples de tipo *tutor*, uno que recogiese la nota obtenida por el examen teórico y que se asociaría la componente teórico, y otro que recogiese la nota del proyecto y que se asociaría al componente que representaba al proyecto. A medida que el curso se desarrollase y se fuesen publicando los ejercicios prácticos a entregar, se añadirían nuevas definiciones de elementos *tutor* que permitiesen recoger la puntuación obtenida en cada una de ellas, a la vez que se modificaría fórmula de evaluación del componente de proceso que las representa en conjunto para que las tuviese en cuenta.

A continuación se incluye la descripción en XML del perfil de evaluaciones utilizado:

```

<evaluationsProfile id='UoL-EAO'>

  <procComponents>
    <procComponent id='pc-Teoría' desc='Contenidos teóricos del curso'
      datatype='number' initialValue='0' >
      <expresion>
        <basicExpresion>
          <tutorComponent-Ref id-ref='t-examen' />
        </basicExpresion>
      </expresion>
    </procComponent>
    <procComponent id='pc-Prácticas' desc='Ejercicios prácticos'
      datatype='number' initialValue='0' >
    </procComponent>
    <procComponent id='pc-Proyecto' desc='Desarrollo del proyecto'
      datatype='number' initialValue='0' >
      <expresion>
        <basicExpresion>
          <tutorComponent-Ref id-ref='t-Proyecto' />
        </basicExpresion>
      </expresion>
    </procComponent>
  </procComponents>

  <plain-Components>
    <tutor-component id='t-examen' desc='Resultado examen teórico'
      datatype='number' initialValue='-1'
      texto= 'Introduzca el resultado del examen teórico'/>
    <tutor-component id='t-Proyecto' desc='Nota proyecto'
      datatype='number' initialValue='-1'
      texto= 'Introduzca la calificación
        obtenida en el desarrollo del proyecto'/>
  </plain-Components>

```

```

<learning-objectives>
  <learning-objective id='lo1-teoría' desc='Conocimientos teóricos'
                    datatype='number' initialValue='0'>
    <evaluation>
      <basicExpression>
        <tutor-component-ref id-ref='t-examen' />
      </basicExpression>
    </evaluation>
    <relations>
      <procComponent-ref ref='pc-Teórico' />
    </relations>
  </learning-objective>

  <learning-objective id='lo2-práctica' desc='Aplicación práctica
                    estándares' datatype='number' initialValue='0'>
    <evaluation>
      <expression>
        <mathExpression>
          <term>
            <factor>
              <basicExpression>
                <procComponent-ref ref='pc-prácticas' />
              </basicExpression>
            </factor>
            <mult>
              <basicExpression>
                <langString>0.4</langString>
              </basicExpression>
            </mult>
          </term>
          <sum>
            <term>
              <mathExpression>
                <term>

```

```

        <factor>
            <basicExpresion>
                <tutorComponent-ref
                    ref='pc-Proyecto' />
            </basicExpresion>
        </factor>
        <mult>
            <basicExpresion>
                <langString>0.6</langString>
            </basicExpresion>
        </mult>
    </term>
</mathExpresion>
</term>
</sum>
</mathExpresion>
</expresion>
</evaluation>

<relations>
    <procComponent-ref ref='pc-teórico' />
    <procComponent-ref ref='pc-Prácticas' />
    <procComponent-ref ref='pc-Proyecto' />
</relations>
</learning-objective>
</learning-objectives>

</evaluationsProfile>

```

FASES 2 y 3: Monitorización, desarrollo e introducción de adaptaciones

Podemos distinguir dos grupos distintos de adaptaciones de entre las introducidas a lo largo del curso de EAO. El primero incluiría aquéllas utilizadas para publicar las actividades teóricas y prácticas correspondientes a cada semana, y el segundo aquéllas introducidas con el propósito de llevar a cabo algún tipo de corrección o reajuste. Los

pokes correspondientes al segundo grupo de adaptaciones son similares a los vistos en los dos escenarios previos de evaluación, es decir, tienen un propósito correctivo o perfectivo, mientras que los correspondientes al primer grupo poseen ciertas características comunes ya que, de hecho, no deben ser considerados como adaptaciones del diseño del proceso sino como parte del mismo. Así, todas fueron acciones de inserción, ya que añadían nuevos elementos al proceso; ninguna incluyó evaluación del éxito del cambio, puesto que no fueron cambios sino concreciones del diseño; e introdujeron fragmentos de manifiesto grandes en comparación con el resto de adaptaciones. Debido a los nuevos elementos introducidos algunas requirieron, además, la modificación del perfil de evaluaciones.

Debido a las características especiales de este escenario de evaluación se produjeron un número elevado de adaptaciones sobre el diseño inicial del proceso, si bien, tal y como se ha mencionado arriba, la mayoría de ellas presentaba la misma estructura y características. Por ello, en esta sección, se proveerá la descripción de una única adaptación, representativa sin embargo, del resto.

A.10 **Introducción de nuevas actividades** [*Introducción de material*]:

Esta adaptación corresponde al grupo de las que definían las actividades a realizar a lo largo de una semana determinada, concretamente a una de las primeras semanas del curso. El estándar abordado era SCORM, y la adaptación suponía la definición de una nueva *activity-structure* que a su vez incluía dos actividades: la primera agrupaba el contenido teórico, y la segunda el ejercicio práctico a realizar. Ambas actividades presentaban la misma estructura: una descripción de tareas a desarrollar provista mediante un fichero *html*, y un *entorno* donde eran incluidos los recursos necesarios para la realización de cada tarea.

La adaptación suponía la definición de una nueva entrega de ejercicios que sería tomada en cuenta en el cómputo de la nota final de la asignatura. Por ello, el perfil de evaluaciones fue modificado introduciendo, por una parte, un nuevo elemento *tutor* que recogiese la nota obtenida, y cambiando, por otra, la definición del componente de proceso que representaba todas las entregas para que tuviese en cuenta la puntuación en el nuevo ejercicio.

Por último, dado que el *poke* suponía la introducción de nuevos contenidos teóricos cuya asimilación afectaría al éxito y capacidad del alumno para llevar a cabo el proyecto, se definieron relaciones con los componentes de proceso que representaban a los

elementos del proceso asociadas con dichas capacidades. Además, para liberar al instructor de la tarea de introducir el cambio en el momento exacto, se especificó la fecha en que la modificación debía producirse por medio de una condición de activación.

La definición exacta del *poke* asociado a esta adaptación es la siguiente:

- Acción adaptativa:

```
<Poke id='poke-Semana3' desc='Definición actividades tercera
                               semana'>

  <actions>
    <insertion>
      <IdElement-Ref ref='AS-Semana3' />
      <ParentElement-Ref ref='AS-Principal' pos='3' />
    </insertion>
  </actions>

  <activation-condition>
    <dateExpression>
      <Time>13/03/2007 09:00:00<Time>
    </dateExpression>
  </activation-condition>

  <relations>
    <procComponent-ref id-ref='pc-Teoría' />
    <procComponent-ref id-ref='pc-Prácticas' />
    <procComponent-ref id-ref='pc-Proyecto' />
  </relations>

</Poke>
```

- Descripción de nuevos elementos:

En el perfil de evaluaciones:

```
<tutor-component id='t-ejer-SCORM_Model' desc='Introduzca la
                               calificación obtenida en el ejercicio secuenciamiento
```

```

en SCORM' datatype='number' initialValue='0'
texto= 'Introduzca la calificación obtenida en
        el ejercicio de secuenciamiento en SCORM' />

<procComponent id='pc-Prácticas' desc='Ejercicios prácticos'
                datatype='number' initialValue='0' >
  <expression>
    <basicExpression>
      <mathExpression>
        <term>
          <factor>
            <basicExpression>
              <tutorComponent-ref ref='t-ejer-SCORM_Model' />
            </basicExpression>
          </factor>
          <mult>
            <basicExpression>
              <langString>0.5</langString>
            </basicExpression>
          </mult>
        </term>
      </mathExpression>
    </basicExpression>
  </expression>
  <sum>
    <term>
      <mathExpression>
        <term>
          <factor>
            <basicExpression>
              <tutorComponent-ref ref='t-ejer-SCORM_Sec' />
            </basicExpression>
          </factor>
          <mult>
            <basicExpression>
              <langString>0.5</langString>
            </basicExpression>
          </mult>
        </term>
      </mathExpression>
    </term>
  </sum>
</procComponent>

```

```

                </mult>
            </term>
        </mathExpresion>
    </term>
</sum>
</mathExpresion>
</basicExpresion>
</expresion>
</procComponent>

```

En el manifiesto:

```

<imsld:learning-activity identifier="ACT-Teoría-Semana3"
                        isvisible="true">
    <imsld:title>SCORM: API y Secuenciamiento</imsld:title>
    <imsld:environment-ref ref="ENV-Teoría-Semana3" />
    <imsld:activity-description>
        <imsld:title>SCORM</imsld:title>
        <imsld:item identifier="item-desc-SCORM-sec"
                    identifierref="res-act-SCORM-sec" isvisible="true" />
    </imsld:activity-description>
</imsld:learning-activity>

<imsld:learning-activity identifier="ACT-Práctica-Semana3"
                        isvisible="true">
    <imsld:title> Ejercicio: SCORM Secuenciamiento</imsld:title>
    <imsld:environment-ref ref="ENV-Práctica-Semana3" />
    <imsld:activity-description>
        <imsld:title>
            Actividades individuales: Secuenciamiento
        </imsld:title>
        <imsld:item identifier="item-desc-Ejercicio-SCORM-sec"
                    identifierref="res-act-ejercicio-SCORM-sec"
                    isvisible="true" />
    </imsld:activity-description>

```



```
</imsld:learning-activity>

<imsld:activity-structure identifier="AS-Semana3">
  <imsld:title>Semana 3</imsld:title>
  <imsld:environment-ref ref="ENV-Semana3" />
  <imsld:learning-activity-ref ref="ACT-Teoría-Semana3" />
  <imsld:learning-activity-ref ref="ACT-Práctica-Semana3" />
</imsld:activity-structure>

<imsld:environment identifier="ENV-Teoría-Semana3">
  <imsld:title>Teoría SCORM</imsld:title>

  <imsld:learning-object identifier="lo-SCORM-Modelo"
    isvisible="true" type="knowledge-object">
    <imsld:title>
      Informe CNICE: Modelo de referencia ADL SCORM
    </imsld:title>
    <imsld:item identifier="item-SCORM-Modelo"
      identifierref="res-SCORM-Modelo" isvisible="true" />
  </imsld:learning-object>

  <imsld:learning-object identifier="lo-SCORM-Sec"
    isvisible="true" type="knowledge-object">
    <imsld:title>
      Secuenciamiento y API SCORM
    </imsld:title>
    <imsld:item identifier="item-SCORM-API"
      identifierref="res-SCORM-API" isvisible="true" />
  </imsld:learning-object>
  .....
</imsld:environment>
.....
```

```

<resource identifier="res-act-SCORM-sec" type="webcontent"
           href="desc-SCORM-sec.html">
  <file href="desc-SCORM-sec.html" />
</resource>

<resource identifier="res-act-ejercicio-SCORM" type="webcontent"
           href="desc-ejercicio-SCORM-sec.html">
  <file href="desc-ejercicio-SCORM-sec.html" />
</resource>

<resource identifier="res-SCORM-Modelo"
           type="webcontent"
           href="CNICE-SCORM.pdf">
.....

```

- Nuevos recursos:

```

desc-SCORM-sec.html
desc-ejercicio-SCORM-sec.html
CNICE-SCORM.pdf
SCORM-Modelo.pdf
.....

```

FASES 5, 6 y 7: Evaluación de adaptaciones y resultados e integración

Una vez hubo concluido el curso se evaluaron los resultados del proceso así como las distintas adaptaciones introducidas. En cuanto al proceso, cabe destacar que todos los alumnos superaron tanto la parte teórica como la parte práctica de la asignatura por lo que los resultados se consideraron satisfactorios. La flexibilidad en la planificación permitió disminuir el número de entregas inicialmente previstas, así como hacer propuestas de proyectos acordes con el perfil de los alumnos, cuyo nivel de conocimiento de programación, en algunos casos, era muy bajo.

A la hora de analizar las adaptaciones introducidas es necesario distinguir entre los dos tipos distintos de cambios diferenciados al comienzo de la sección: las que suponen la concreción del diseño del curso y las que suponen la introducción de un cambio. En

cuanto a las primeras, tal y como se comentó anteriormente, no se definió ninguna evaluación que estimase la bondad de su acción. El éxito de las mismas fue determinado por el propio éxito del diseño del proceso, y todas forman parte del diseño final integrado. Por otra parte, dado que el propio diseño fue concretado durante la ejecución del proceso, fue posible ir proporcionando respuesta a los distintos inconvenientes que fueron surgiendo, eliminándose, a la vez, la necesidad de introducir adaptaciones del segundo tipo que introdujesen correcciones o modificaciones. Así pues, el número de las adaptaciones correctivas fue muy reducido y no acarrearón cambios de consideración que fuese necesario evaluar. En cualquier caso, la captura implícita de estos cambios y correcciones dentro de las adaptaciones de diseño impide obtener información acerca de la naturaleza del cambio y determinar si su introducción obedece a circunstancias propias del contexto de ejecución o no.

5.3.4. Resumen de la aplicación a escenarios de aprendizaje reales

Las tablas 5.6 y 5.7 muestran un resumen de las características de las adaptaciones y observaciones introducidas a lo largo del desarrollo de las distintas fases del método de diseño iterativo en los tres escenarios previamente descritos. Así, en cuanto a las adaptaciones (tabla 5.6), cada columna especifica la causa de la adaptación, si proporcionaba una respuesta a una situación concreta dentro del contexto de ejecución del proceso o si por el contrario tenía carácter general, si fue aplicada a todas las instancias en ejecución o sólo a determinados usuarios y si se establecieron condiciones para su activación o fórmulas de evaluación de su éxito. En cuanto a las observaciones (tabla 5.7), se especifica si fue posible completar la operación con éxito, qué dato se observó, si la observación fue realizada de forma continua o únicamente en un instante determinado, si fue aplicada a todas las instancias en ejecución o sólo a determinados usuarios y si se establecieron condiciones para su activación.

Tal y como se ha comentado al comienzo de la sección, la aplicación del método de diseño iterativo en los tres escenarios descritos ha servido tanto para validar su factibilidad, como para completar la validación de la factibilidad del mecanismo de implementación y de la expresividad del modelo. Las relaciones entre las dos pruebas realizadas para completar esta última validación y los sub-objetivos definidos para el modelo de adaptación pueden observarse en la tabla 5.8.

A continuación se presentan las conclusiones que el análisis de la información obte-

Tabla 5.6: Relación de las distintas adaptaciones introducidas en los escenarios de la evaluación empírica

Esc.	Nº	Adaptación	Causa	Carácter	Instancias objetivo	Condición	Evaluación
1	A.1	Añadir material complementario	Perfil alumno / Solicitud	General	Todos	No	No
1	A.2	Añadir material complementario	Problemas de conexión	General	Todos	No	No
1	A.3	Añadir material complementario	Solicitud alumnos	General	Todos	No	No
1	A.4	Sustituir material	Problema material	General	Todos	No	Sí
1	A.5	Suprimir material	Decisión instructor	Contextual	Algunas	No	No
2	A.6	Sustituir material	Recurso no disponible	Contextual	Todos	No	No
2	A.7	Sustituir material	Problema material	General	Todos	No	No
2	A.8	Añadir servicio	Mejorar comunicación	Contextual	Algunas	Sí	No
2	A.9	Añadir recursos	Decisión instructor	General	Todos	No	No
3	A.10	Concretar parte del diseño	Planificada	General	Todos	Sí	No

Tabla 5.7: Relación de las distintas observaciones introducidas en los escenarios de la evaluación empírica

Esc.	Nº	Observación	Éxito	Dato	Acción	Instancias objetivo	Condición
1	O.0 ¹	Observación asociada a un componente de evaluación interno	S	Propiedad	Continua	Todas	No
1	O.1	Recuperación exitosa de valor de un atributo en ejecución	S	Número accesos	Instantánea	Algunos	No
1	O.2	Recuperación fallida de valor de un atributo en ejecución	N	Tiempo	Instantánea	Algunos	Sí
2	O.3	Recuperación fallida de valor de un atributo en ejecución	N	Tiempo	Instantánea	Algunos	No

¹Esta observación fue introducida con objeto de cargar el valor correspondiente al componente interno del perfil de evaluaciones *i-TestFinal*, el cual representaba el resultado obtenido por el alumno en el test de evaluación final del curso.

Tabla 5.8: Validación de la expresividad del modelo de adaptación

Objetivo	Descripción	Casos de uso	Aplicación escenarios
O1.1	Modelo adaptación capaz de modificar la definición del proceso y su comportamiento en ejecución.	1, 3, 4, 5, 6, 7 y 8	Todas las adaptaciones han podido ser expresadas mediante elementos del modelo
O1.2	Modelo adaptación que permita especificar el momento de la adaptación	4, 5 y 6	A.10 Adaptación con condición asociada a una fecha
O1.3	Modelo que soporte la monitorización del desarrollo del proceso	2, 4, 5 y 7	O.0 Observar atributo O.1 Observar característica
O1.4	Modelo que facilite el análisis del éxito de las adaptaciones	3 y 7	A.4 Adaptación con evaluación asociada
O1.5	Modelo que facilite el análisis del éxito del proceso		Los perfiles de evaluación han permitido expresar las evaluaciones requeridas en los 3 escenarios

Tabla 5.9: Relación entre objetivos y acciones de la evaluación empírica

Objetivo	Descripción	Evaluación Empírica
01. Modelo		
01.1	Modelo adaptación capaz de modificar la definición del proceso y su comportamiento en ejecución.	Todas las adaptaciones han podido ser expresadas mediante elementos del modelo
01.2	Modelo adaptación que permita especificar el momento de la adaptación.	A.10 Adaptación con condición asociada a una fecha
01.3	Modelo que soporte la monitorización del desarrollo del proceso	O.0 Observar atributo O.1 Observar característica
01.4	Modelo que facilite el análisis del éxito de las adaptaciones	A.4 Adaptación con evaluación asociada
01.5	Modelo que facilite el análisis del éxito del proceso	Los perfiles de evaluación han permitido expresar las evaluaciones requeridas en los 3 escenarios
02. Método		
02.1	Método que ordene el desarrollo de las tareas del instructor soportadas mediante elementos del modelo de adaptación	Aplicación método en los tres escenarios
02.2	Método que recoja de forma explícita el cambio introducido sobre la definición original del proceso	Adaptaciones realizadas en los tres escenarios
02.3	Método que permita realizar las actividades de adaptación y monitorización del proceso	Adaptaciones y monitorizaciones realizadas en los tres escenarios
02.4	Método que permita realizar las actividades de evaluación de los cambios introducidos y éxito del proceso	A.4 Adaptación con evaluación asociada Evaluación desarrolladas en los tres escenarios
03. Mecanismo implementación		
03.1	Mecanismo implementación modelo no intrusivo	Mantenimiento separado de diseño de proceso y de adaptaciones A.5, A.6 y A.8
03.2	Mecanismo implementación modelo consistente	Validación de las adaptaciones
03.3	Mecanismo implementación modelo flexible	

nida tras el desarrollo de los tres procesos de aprendizaje ha permitido extraer relacionándolas cuando corresponde, tal y como muestra la tabla 5.9, con los objetivos de la tesis :

- La necesidad de disponer de un mecanismo que permita al instructor la modificación del diseño original del proceso de aprendizaje ha sido puesta de manifiesto en los tres escenarios. Dicha necesidad ha sido originada bien por problemas en la planificación o en el material proporcionado (A.4 y A.7), bien por causas externas al proceso (A.2 y A.6) o bien por iniciativa de los propios participantes del mismo que han sugerido alguna mejora (A.1, A.3, A.5, A.8 y A.9). En cualquier caso el modelo de adaptación propuesto ha sido suficientemente expresivo como para permitir la descripción de todas ellas **[O1.1]**, al igual que el método ha permitido su aplicación **[O2.3]**.
- Debido al escaso número de adaptaciones de carácter correctivo introducidas, sólo se ha encontrado una adaptación para la que se ha definido fórmula de evaluación de su éxito (A.4). Tanto el modelo como el método han permitido especificar dicha fórmula **[O1.4]** y llevar a cabo la correspondiente evaluación **[O2.4]**.
- Ha sido posible programar una adaptación para su activación en una fecha concreta (A.10) **[O1.2]** o condicionar su aplicación a la recuperación de un determinado valor en uno de los componentes del perfil de evaluaciones (A.8).
- El mecanismo de validación propuesto ha servido para garantizar la validez del proceso tras la introducción de cada una de las adaptaciones descritas **[O3.2]**.
- Igualmente, el modelo ha demostrado proporcionar suficientes elementos para permitir el desarrollo de las tareas de observación (O.1) y monitorización (O.0) del proceso según el método **[O1.3, O2.3]**. Las limitaciones detectadas en el desarrollo de dichas tareas (O.2 y O.3) se han debido a las características del *IMS LD player* utilizado para implementar el proceso. Esto es también aplicable al caso de las adaptaciones, donde las limitaciones del propio EML han obligado al uso de servicios externos para la implementación de alguna de ellas (A.5).
- La evaluación del éxito del proceso a través de los elementos del modelo y siguiendo el método propuesto también ha sido posible en los tres escenarios **[O1.5, O2.4]**. Dicha evaluación se ha llevado a cabo mediante la especificación de los

objetivos de aprendizaje marcados para cada proceso, las fórmulas de evaluación de los mismos, así como la forma de recuperar los valores de los componentes de dichas evaluaciones. Cabe destacar que, dado que dos de los escenarios de enseñanza se desarrollaron en formato semi-presencial y que las pruebas de evaluación de los alumnos en estos casos no se encontraban automatizadas, se ha requerido de la intervención directa del instructor para la recuperación de la información necesaria para la evaluación de los objetivos. En cualquier caso, hubiese sido posible lograr una mayor transparencia en el desarrollo del proceso de evaluación si dicha información hubiese sido almacenada en un dispositivo al que el motor de ejecución tuviese acceso directo.

- El desarrollo de las tareas del instructor soportadas mediante elementos del modelo y organizadas según las distintas fases del método de diseño iterativo ha facilitado el análisis de las acciones llevadas a cabo, simplificando tanto la identificación tanto de relaciones entre las mismas (A.6 y A.7), como de mejoras introducidas de carácter general (A.1, A.2, A.3, A.4, A.7 y A.9) o problemas propios del actual contexto de ejecución (A.5, A.6 y A.8) [O2.1, O2.2]. Cabe destacar, como ventajas adicionales del empleo de la solución sobre la versión del proceso en UoL con respecto al mismo proceso implementado sobre la plataforma Moodle:
 - Facilidad para crear visiones personalizadas del proceso por participante, aún durante la propia ejecución (A.8).
 - El desacoplamiento entre la descripción de las adaptaciones y la descripción original del proceso permite aplicar todas las modificaciones necesarias para adaptar el desarrollo del curso a un contexto concreto salvaguardando al mismo tiempo la versión original (A.5, A.6 y A.8) [O3.1].
- El modelo y método propuestos permiten llevar a cabo una planificación abierta del proceso de aprendizaje que es concretada durante su propio desarrollo mediante las apropiadas adaptaciones (3^{er} escenario). En cualquier caso, este tipo de planificación debe ser tenida en cuenta a la hora de preparar el perfil de evaluaciones, de forma que se evite o reduzca la necesidad de actualizar sus componentes a medida que nuevos elementos de proceso sean añadidos. Asimismo, también se debe considerar que las adaptaciones que concretan el diseño del proceso pueden

capturar implícitamente correcciones sobre otras partes del diseño o responder a circunstancias del actual contexto de ejecución.

5.4. Evaluación experta

Para constatar la utilidad de la solución así como la calidad de la misma se llevó a cabo una última prueba de evaluación que tuvo por objeto medir el grado de satisfacción de un conjunto de evaluadores con una serie de características que se desearía que la solución exhibiese. La mayor parte de los evaluadores seleccionados para el proceso fueron expertos en tecnología educativa, familiarizados, por tanto, con el diseño de material educativo y con el uso y posibilidades que los EMLs ofrecen. Sin embargo, y con el fin de recoger la opinión de todos los participantes de un proceso de aprendizaje, se contó también con la colaboración de algunos instructores y varios estudiantes. Así, del total de 15 evaluadores, 11 de ellos son expertos en la materia, 5 diseñan habitualmente contenido educativo, 6 son instructores y 2 estudiantes.

A cada evaluador se hizo llegar un documento que presentaba un resumen de la solución propuesta en el capítulo 4, junto con los resultados obtenidos durante la aplicación del método diseño iterativo al primer escenario de evaluación descrito en la sección anterior. Una vez completada la lectura se solicitaba cumplimentar un cuestionario que tenía como fin percibir el grado de satisfacción del evaluador con las diversas características del modelo, método y mecanismo de adaptación propuesto. El cuestionario incluía tanto preguntas abiertas como preguntas con una escala de cinco puntos (1=nada y 5=mucho). Tanto el cuestionario como los resultados obtenidos se encuentran recogidos en los anexos C y D. A continuación presentamos algunas conclusiones extraídas del análisis de las opiniones de los expertos en materia educativa encuestados.

La primera pregunta del cuestionario no trataba propiamente ningún aspecto de la solución sino que perseguía, dados los diferentes perfiles existentes, obtener una medida del grado de confortabilidad del evaluador con la notación utilizada para describir los elementos del modelo de adaptación. Así, a la vista de los resultados, se puede afirmar que, si bien los encuestados aportaron sugerencias al respecto, todos ellos comprendieron y encontraron apropiada la notación, por lo que se puede suponer ésta no influyó en su capacidad de interpretar y comprender la funcionalidad de los elementos del modelo.

El objetivo de las cuestiones 2, 5, 6 y 7 era recoger el grado de satisfacción del evaluador con respecto a ciertas cualidades que sería deseable que el modelo propuesto

exhibiera, centrándose, cada una de ellas, en una funcionalidad concreta del modelo. Así, la pregunta 2 hacía referencia a la capacidad de describir adaptaciones, la 5 a la capacidad para definir monitorizaciones, y la 6 y 7 a la de describir evaluaciones del éxito de la adaptación y del proceso respectivamente. La figura 5.6 refleja los resultados obtenidos, de donde se puede extraer la conclusión de que la opinión media de los evaluadores es que el modelo exhibe “bastante” todas las cualidades encuestadas. De entre ellas, las características más valoradas son la utilidad y necesidad, y la menos la eficiencia, especialmente en el caso de la evaluación del éxito de la adaptación y el proceso.

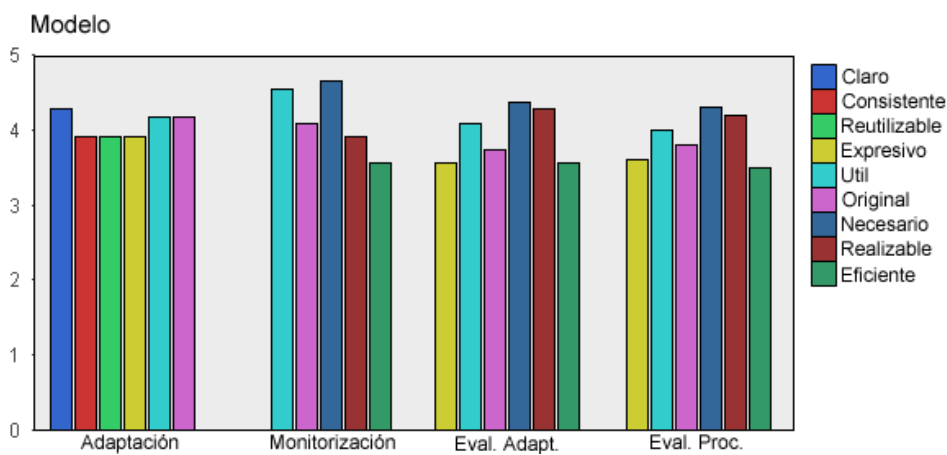


Figura 5.6: Análisis sobre las características del modelo propuesto

Dado que la aportación principal de esta tesis es la introducción de adaptaciones sobre el diseño de un proceso en tiempo de ejecución, se incluyeron dos preguntas (3 y 4) específicas sobre el tema. La primera de ellas pretendía recoger la opinión del evaluador acerca de diferentes aspectos sobre el caso concreto de la adaptación en tiempo de ejecución como, por ejemplo, su utilidad y su sencillez. La segunda evaluaba cuán útil consideraba el poder describir distintas características de las adaptaciones como ciertas condiciones de activación o el momento de introducción, por ejemplo. En la figura 5.7 se pueden observar los resultados obtenidos, de donde se puede deducir que la amplia mayoría considera muy útil y necesaria la modificación del diseño del proceso en ejecución. En cuanto a las características, la más valorada fue la posibilidad de activar la adaptación según los eventos de participantes, y la menos la posibilidad de establecer frecuencia de introducción, si bien, todas fueron valoradas como altamente útiles.

Por otra parte las cualidades del método propuesto en el presente documento fue-

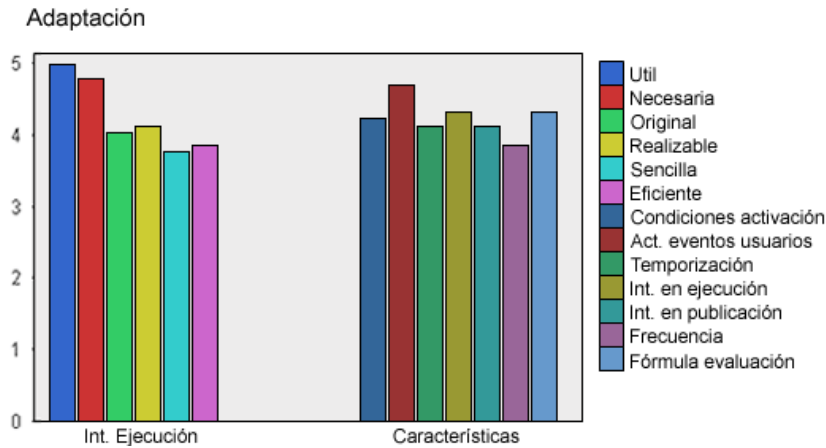


Figura 5.7: Análisis sobre los requisitos de la adaptaciones del diseño

ron evaluadas a través de las preguntas 8, 9 y 10. Las dos primeras se centraban en la automatización de la evaluación de las adaptaciones y éxito del proceso mientras que la tercera recogía la opinión acerca del método general de diseño iterativo de procesos educativos propuesto. Los resultados obtenidos se recogen en la figura 5.8 donde se observa que, si bien la mayoría de las cualidades encuestadas obtienen una puntuación notable, la eficiencia, sobre todo en el caso del mecanismo de evaluación de adaptaciones, obtiene una valoración ligeramente inferior que el resto. Estos resultados parecen estar relacionados con los obtenidos para las mismas características del modelo y pueden deberse a la dificultad para discernir qué es exactamente consecuencia de una adaptación y qué no lo es, si bien, como algún evaluador señala, sí puede servir como un primer indicador del efecto que la adaptación ha tenido sobre el proceso.

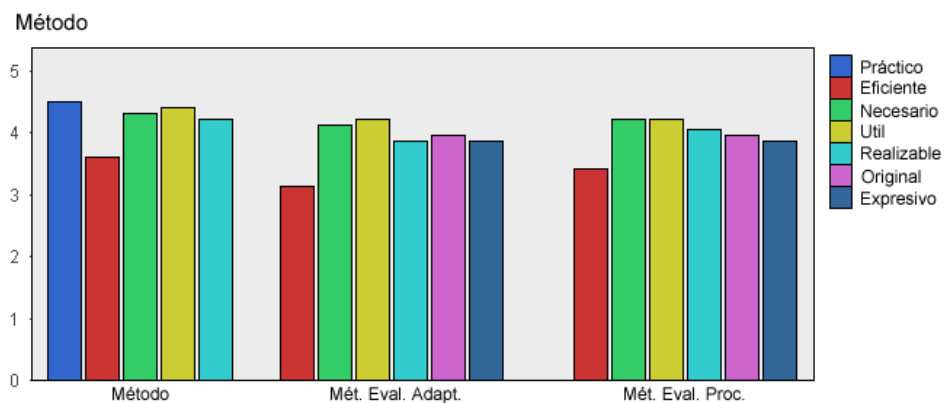


Figura 5.8: Análisis sobre las características del método propuesto

La opinión de los evaluadores acerca de la arquitectura de extensión de motores de ejecución y el mecanismo de validación basado en el uso de ontologías fue recogida

mediante las preguntas 11 y 12. Estas valoraciones se recogen en la figura 5.9 de donde se puede concluir que la característica de la arquitectura con la que los evaluadores se encuentran más satisfechos es su extensibilidad. Las distintas cualidades del mecanismo de validación son consideradas exhibidas en un grado “bastante” alto.

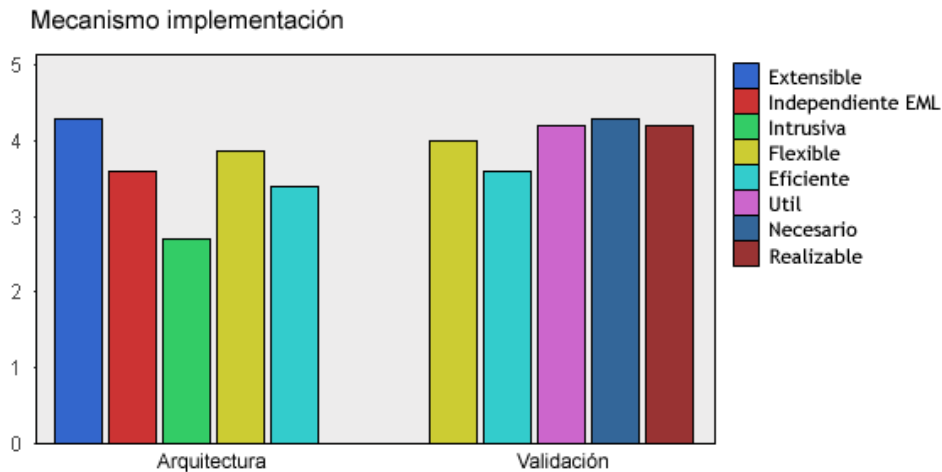


Figura 5.9: Análisis sobre las características del mecanismo de implementación propuesto

Por último, el cuestionario finalizaba con un par de preguntas abiertas para que el evaluador comentase cuáles eran según su criterio las principales ventajas e inconvenientes de la solución, así como expresase su disposición o no a utilizar un sistema que la implementase. De entre las ventajas, la más señalada fue que la solución permite romper la rigidez de las especificaciones y herramientas, y aporta la necesaria flexibilidad a la puesta en práctica de un diseño de un proceso de aprendizaje. También fue destacada la utilidad que, frente a otras opciones de implementación en LMSs que no lo permiten, supone el poder modificar el diseño del proceso en ejecución y preservar, al mismo tiempo, su definición original separada. Todos los evaluadores se mostraron favorables a usar un sistema que implemente la solución y casi no se reseñaron inconvenientes a la solución, si bien varios indicaron que podría ser necesario introducir ciertas modificaciones sobre la misma para adecuarla a especificaciones distintas del IMS LD. Este respecto ya fue tenido en cuenta por el autor durante el diseño del modelo de adaptación, cuando se optó por utilizar como guía para el desarrollo de la solución dicha especificación. Esta decisión se basó en el hecho de que, a día de hoy, el IMS LD es la especificación más completa y de uso más extendido, con gran diferencia respecto al resto.

Para terminar con esta sección señalar que del análisis de los resultados conside-

rando únicamente las respuestas de los participantes de procesos educativos, esto es diseñadores, instructores y estudiantes sean o no expertos en tecnología educativa, se obtienen conclusiones similares, si bien, tal y como muestra la figura 5.10, son aún más valoradas tanto las características de las adaptaciones encuestadas, como la necesidad y utilidad de la introducción de adaptaciones en ejecución.

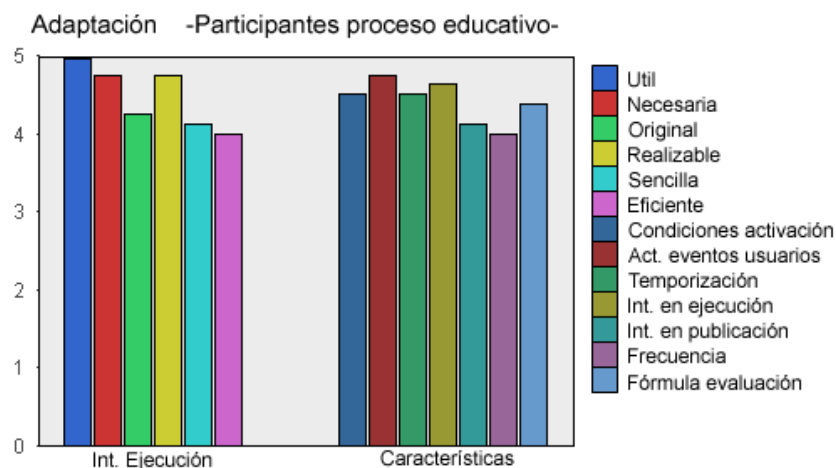


Figura 5.10: Análisis sobre los requisitos de la adaptaciones del diseño por los participantes de un proceso educativo

5.5. Resumen

En este capítulo se ha descrito el conjunto de evaluaciones llevadas a cabo para determinar tanto la generalidad y expresividad de la solución propuesta, como su factibilidad, calidad y utilidad.

A continuación, se muestra un resumen de dichas pruebas de evaluación junto con su objetivo, los elementos que han participado y el resultado de realizar dicha evaluación.

▪ Desarrollo de un caso de estudio: IMS LD

- **Objetivo:** Probar la factibilidad de la solución propuesta a través de su aplicación a un EML concreto, el IMS LD.
- **Sujeto de evaluación:** El modelo y mecanismo de adaptación.
- **Resultado:** Se ha demostrado que el modelo es aplicable a la especificación seleccionada para el caso de estudio. La selección de esta especificación concreta para el proceso de validación se ha llevado a cabo empleando criterios

de compleción y aceptación de las propuestas. Igualmente, se ha demostrado que es posible diseñar un módulo de extensión que permita ampliar las capacidades de un IMS LD Player existente para que pueda interpretar los elementos del modelo y llevar a cabo la operación correspondiente.

▪ Desarrollo de un conjunto de casos de uso

- **Objetivo:** Validar la generalidad y la expresividad del modelo de adaptación propuesto.
- **Sujeto de evaluación:** El modelo de adaptación.
- **Resultado:** El modelo aporta los elementos necesarios para permitir la descripción de los distintos tipos adaptaciones y monitorizaciones con las características especificadas por los sub-objetivos recogidos en el capítulo 1 para el modelo de adaptación.

▪ Aplicación de la solución a tres escenarios de aprendizaje

- **Objetivo:** Probar la factibilidad del método de diseño iterativo, así como completar la validación de la factibilidad del mecanismo y de la expresividad del modelo.
- **Sujeto de evaluación:** El modelo, método y mecanismo de adaptación.
- **Resultado:** Ha sido posible implementar las distintas fases del método de diseño iterativo sobre cada uno de los tres escenarios de aprendizaje, capturando los cambios introducidos a lo largo de su desarrollo y permitiendo llevar a cabo las distintas actividades de adaptación, observación y evaluación requeridas. Igualmente, el modelo ha probado aportar suficientes elementos para permitir la descripción de todas las adaptaciones y observaciones ocurridas, así como definir el conjunto de evaluaciones necesarias para probar la satisfacción de los objetivos de aprendizaje. Mediante el mecanismo de implementación propuesto ha sido posible validar la definición del proceso resultante tras la introducción de cada una de las adaptaciones, así como permitir mantener separada la definición original del proceso de la definición de las adaptaciones.

▪ Evaluación Experta

- **Objetivo:** Probar la utilidad de la solución propuesta, así como la calidad de la misma según un conjunto determinado de características que se desearía exhibiera.
- **Sujeto de evaluación:** El modelo, método y mecanismo de adaptación.
- **Resultado:** Los evaluadores determinaron su satisfacción con la solución propuesta, resaltando especialmente la utilidad de la misma y la necesidad del desarrollo de propuestas que, al igual que ésta, permitan romper la rigidez de las especificaciones de EMLs y aporten flexibilidad en su aplicación.

Las actividades de evaluación realizadas y sus relaciones con cada objetivo específico, según fueron enunciados en el capítulo 1, se encuentran resumidas en la tabla 5.10. Es necesario hacer una mención especial a la evaluación del objetivo [O3.3]. A día de hoy no se han producido cambios en la especificación del IMS LD que hayan permitido evaluar de forma empírica la flexibilidad del mecanismo de implementación del modelo por lo que únicamente ha sido posible contar con la valoración positiva de los evaluadores a ese respecto. Además, el patrón de diseño aplicado (*Visitor*) en la arquitectura propuesta sugiere que la simple adición de nuevos métodos *visit* en la clase *Adaptor* sea suficiente para modificar el conjunto de elementos seleccionados como adaptables dentro esa especificación, siempre y cuando los nuevos elementos de la especificación mantengan la estructura jerárquica propia de la definición del manifiesto.

Una vez presentado tanto el modelo propuesto para la adaptación de diseños de procesos de aprendizaje especificados por EML y supervisados por instructor, como las tareas llevadas a cabo para su evaluación y los resultados obtenidos en el desarrollo de las mismas, sólo resta extraer y analizar las conclusiones obtenidas a lo largo del proceso de investigación y realizar una síntesis del trabajo realizado.

Tabla 5.10: Relación entre objetivos y pruebas de evaluación

Objetivo	Descripción	Evaluación
O1. Modelo		
O1.1	Modelo adaptación capaz de modificar la definición del proceso y su comportamiento en ejecución.	G ¹ , E ² , F ³ y C ⁴
O1.2	Modelo adaptación que permita especificar el momento de la adaptación.	E, F y C
O1.3	Modelo que soporte la monitorización del desarrollo del proceso	E, F y C
O1.4	Modelo que facilite el análisis del éxito de las adaptaciones	E, F y C
O1.5	Modelo que facilite el análisis del éxito del proceso	E, F y C
O2. Método		
O2.1	Método que ordene el desarrollo de las tareas del instructor soportadas mediante elementos del modelo de adaptación	F y C
O2.2	Método que recoja de forma explícita el cambio introducido sobre la definición original del proceso	F y C
O2.3	Método que permita realizar las actividades de adaptación y monitorización del proceso	F y C
O2.4	Método que permita realizar las actividades de evaluación de los cambios introducidos y éxito del proceso	F y C
O3. Mecanismo implementación		
O3.1	Mecanismo de implementación del modelo no intrusivo	F y C
O3.2	Mecanismo de implementación del modelo consistente	F y C
O3.3	Mecanismo de implementación del modelo flexible	C

¹G: Generalidad²E: Expresividad³F: Factibilidad⁴C: Calidad

Capítulo 6

Conclusiones

En este capítulo se presentará las principales conclusiones alcanzadas y las aportaciones originales que han surgido del proceso de investigación. Después se detallarán posibles carencias y limitaciones de aplicación del trabajo. Finalmente, se presentan las líneas de investigación que quedan abiertas en el campo y otras que pueden constituir la continuación del esfuerzo realizado.

6.1. Conclusiones

Los EMLs resultan adecuados para la especificación de las tareas y actividades a desarrollar en procesos de aprendizaje en los que no existe la figura del instructor. Sin embargo, tal y como se detalló en el capítulo 3, su uso para la descripción de procesos en los que esa figura se encuentra presente, no resulta del todo apropiada: su capacidad de actuación se ve limitada y no es capaz de desarrollar las mismas tareas o acciones que llevaría a cabo en plataformas virtuales no basadas en el uso de estándares, o en procesos de aprendizaje no soportados computacionalmente. Por otra parte, la actualización del diseño de un proceso de aprendizaje con información proveniente de su aplicación práctica es un reflejo del modo de trabajo real de los diseñadores. La utilidad que representa el posibilitar la implementación de este procedimiento y el garantizar el control del instructor sobre el desarrollo de procesos especificados mediante EMLs, ha sido constatada por la práctica unánimidad en la opinión recogida de múltiples expertos en el área de la tecnología educativa consultados, tanto en el proceso de evaluación de la tesis, como en los diversos congresos y talleres donde el trabajo ha sido presentado [92, 91, 95]. Dicha utilidad resulta especialmente relevante en el caso de dominios complejos donde existen muchos parámetros que no pueden ser conocidos previamente a la ejecución.

El proceso de evaluación descrito en este documento ha probado tanto la adecuación del método propuesto al desarrollo de procesos de aprendizaje reales, como la capacidad expresiva del modelo para permitir la descripción de las distintas adaptaciones del diseño del proceso. Dicha capacidad expresiva ha sido constatada en tres escenarios de aprendizaje distintos, a lo largo de cuyo desarrollo fue posible traducir las adaptaciones que sobre la descripción del proceso implementado en la plataforma utilizada en un entorno real fueron introducidas, a adaptaciones sobre una versión del proceso en forma de UoL. Por tanto, la aplicación práctica de la solución queda supeditada a la disponibilidad de herramientas que, además de proveer una implementación para la misma proporcionen los adecuados interfaces, tanto para posibilitar la rápida autoría de las adaptaciones, como para facilitar la gestión de la información recogida acerca de la evolución del proceso. Dichas herramientas podrán ser desarrolladas siguiendo la arquitectura propuesta para la extensión de motores de ejecución de procesos de aprendizaje especificados mediante EMLs, una vez que se disponga de versiones comerciales de los mismos que ofrezcan la fiabilidad necesaria para su uso en entornos de aprendizaje reales.

A continuación se presentarán una serie de conclusiones particulares que sobre el modelo, método y mecanismo propuesto se han obtenido a lo largo del desarrollo de este trabajo y a la vista de los datos obtenidos de su evaluación.

6.1.1. Conclusiones sobre el modelo

El objetivo general de esta tesis es incrementar las posibilidades de adaptación de los diseños de procesos de enseñanza supervisados por instructor especificados mediante EML. Con tal objeto, se planteó como objetivo inicial el desarrollo de un modelo de adaptación que permitiese extender el modelo de información de un EML permitiendo a los instructores de procesos de aprendizaje de dichas características la aplicación de adaptaciones análogas a las que llevarían a cabo en un entorno de enseñanza tradicional. Este objetivo surge, tal y como se explicó en el capítulo 3, al tratar de implementar este tipo de procesos y detectar las limitaciones en el soporte ofrecido actualmente al instructor que supervisa su desarrollo.

Con este propósito se construyó un modelo de adaptación que, partiendo de la especificación de un EML ya existente, permitiese su extensión y posibilitase la descripción de modificaciones tanto sobre la definición de los diseños de procesos desarrollados empleando dicha especificación, como sobre su comportamiento en ejecución. Dado que

para detectar la necesidad de adaptar y conocer el tipo y características de la adaptación requerida, es necesario que el instructor pueda evaluar y monitorizar el desarrollo del proceso, así como conocer el grado de éxito alcanzado por las adaptaciones introducidas, el modelo de adaptación fue ampliado para incluir los elementos necesarios para proporcionar soporte al desarrollo de estas funciones.

Se puede concluir que, además de constituir una base para permitir al tutor la modificación de la propia definición del proceso durante su ejecución, el modelo desarrollado proporciona una oportunidad para poner en práctica los principios del contextualismo funcional, de manera que se puedan definir y evaluar la consecución de los objetivos de aprendizaje marcados, así como el éxito de las modificaciones introducidas sobre los elementos que componen el proceso. Siguiendo los preceptos del constructivismo, tanto los objetivos como las modificaciones sobre la definición del proceso podrán ser definidas a lo largo del desarrollo del mismo.

6.1.2. Conclusiones sobre el método de diseño iterativo

El segundo de los objetivos generales planteados consistía en organizar el desarrollo de las tareas de observación, evaluación y adaptación del instructor, para lo cual se definió un método de diseño iterativo de procesos de aprendizaje que permite capturar dentro del propio diseño la experiencia obtenida de sus ejecuciones previas.

El proceso de evaluación descrito en este documento ha probado la correspondencia del método con el modo de trabajo de los instructores. Dicha correspondencia ha sido constatada mediante su aplicación en tres escenarios de aprendizaje distintos, y contrastada con la opinión de los evaluadores, de entre los cuales un 93 % consideraron el método propuesto como bastante o muy necesario, práctico y realizable.

6.1.3. Conclusiones sobre el mecanismo de implementación

Finalmente, y según lo propuesto como tercer objetivo general de la tesis, se diseñó un mecanismo para la implantación del modelo que permite mantener la lógica de las adaptaciones separada de la información original de la UoL, de tal forma que puedan definirse y aplicarse en tiempo de ejecución adaptaciones no previstas con anterioridad al comienzo de la misma.

Este mecanismo permite además servir de base para implementar, siguiendo un enfoque constructivista, entornos donde los diseños de los procesos de aprendizaje son concretados durante la propia ejecución del proceso. Dicho enfoque resultaba difícil de

ser llevado a la práctica mediante UoLs, dado que éstas proporcionan descripciones cerradas del diseño, encapsulando conjuntamente tanto la lógica de adaptación del proceso como los recursos utilizados en su desarrollo. Sin embargo, haciendo uso de este mecanismo el instructor puede ir precisando el proceso durante su desarrollo, personalizándolo según las propias necesidades de aprendizaje de los participantes, y guiando a los alumnos hacia los objetivos docentes marcados.

Por otra parte, la alta consideración de los expertos, tanto sobre la flexibilidad y capacidad de extensión del mecanismo de implementación como sobre la flexibilidad del mecanismo de validación, sugiere que las implementaciones desarrolladas siguiendo la arquitectura propuesta serán fácilmente adaptables a cambios en la especificación del EML correspondiente o a modificaciones en el conjunto de elementos o características del proceso a adaptar.

6.1.4. Conclusiones adicionales

Para finalizar, se puede concluir que los resultados obtenidos en la elaboración de esta tesis promueven el desarrollo de herramientas que amplíen el ámbito de aplicación de procesos especificados mediante EMLs, aumentando sus prestaciones para posibilitar su uso en nuevos entornos de aprendizaje o extendiendo el alcance de su propósito.

6.2. Aportaciones

La principal aportación de esta tesis ha sido la definición de un modelo de adaptación que permite extender la especificación de un EML existente para permitir al instructor de un proceso de aprendizaje la adaptación del propio diseño del proceso en tiempo de ejecución.

Las aportaciones concretas de esta tesis son las siguientes:

- Desarrollo de un modelo que posibilita la definición y aplicación de adaptaciones sobre el diseño de un proceso de aprendizaje y permita mantener separada la lógica de la adaptación de la información original de la UoL.
- Definición de un instrumento que permite la recuperación de información acerca del estado del proceso, sus componentes o las actividades desarrolladas por sus participantes.

- Elaboración de un mecanismo que permite la especificación tanto de los objetivos de aprendizaje definidos para el proceso, como de la forma de medir el grado de satisfacción de los mismos por los distintos participantes, así como el éxito de las adaptaciones introducidas.
- Definición de un método de diseño de procesos de aprendizaje que organiza el desarrollo de las diferentes tareas del instructor del proceso soportadas mediante elementos del modelo.

Como aportaciones adicionales significativas que surgen como resultado del proceso de investigación caben destacar:

- Un mecanismo de validación de diseños de aprendizaje basado en ontologías: Como respuesta a la necesidad de garantizar la consistencia de la UoL, una vez la adaptación ha sido introducida, fue necesario desarrollar este mecanismo, el cual constituye una aplicación directa del trabajo previo realizado por el Departamento de Electrónica y Computación de la Universidad de Santiago de Compostela sobre ontologías para la descripción de IMS LD [4, 55, 5] y que, además de simplificar el proceso de implementación de la validación de diseños de aprendizaje, ofrece un marco de explotación de las ontologías para fines prácticos.
- Arquitectura de extensión de motores de ejecución: Permite aumentar las capacidades adaptativas del motor original y ha sido concebida haciendo uso del patrón de diseño *Visitor* y un enfoque basado en Programación Orientada a Aspectos. La mayor ventaja de esta arquitectura reside en su fácil ampliación y en que permite mantener separados los elementos de la extensión de los del motor original.
- El mantenimiento externo de la lógica de adaptación del proceso puede ser utilizado para superar limitaciones del EML empleado en la descripción del proceso. Así, tal y como se demostró en el capítulo 5, es posible implementar ciertas secuencias de control, como por ejemplo iteraciones, en EMLs que no proporcionan elementos para su definición, o sincronizar el desarrollo de actividades entre ejecuciones de instancias de UoLs correspondientes a distintos participantes de un proceso colaborativo.

Finalmente, también cabe destacar algunos usos del mecanismo de adaptación y del método de diseño definidos en el presente documento distintos de los inicialmente previstos y previamente descritos:

- Incrementar la reutilización de UoLs protegidos por *copyrights*: El uso de UoLs cuyo diseño se encuentra protegido por derechos de copia restringe sus posibilidades de aplicación en contextos distintos al original. Tal y como expuso Rob Koper durante la presentación de este trabajo en el taller conjunto UNFOLD-PROLEARN sobre Learning Design celebrado en Valkenburg (Holanda) [92], el mecanismo de adaptación descrito permite al instructor modificar el comportamiento en ejecución de la unidad de aprendizaje sin modificar su diseño original y sin violar por tanto las leyes de restricción de uso. De esta manera se salvaguarda la inversión incrementando las posibilidades de reutilización.
- Adaptación de la ejecución por parte del alumno: Si bien este trabajo está fundamentalmente dirigido al diseñador que crea la unidad de aprendizaje y al instructor que controla su ejecución, el mecanismo propuesto puede ser igualmente empleado en procesos educativos donde no existe la figura del tutor. En este caso, proporcionando al alumno una determinada clave de seguridad se permitiría que fuera él quien controlara la ejecución, de tal manera que pudiera repetir ciertas actividades, concluir aquellas que no resulten de su interés y, de manera más general, adaptar la ejecución del proceso a sus propias necesidades siguiendo, una vez más, los principios pedagógicos del constructivismo.
- Método para autoría progresiva de UoLs: El método de diseño iterativo de procesos de aprendizaje puede ser utilizado como alternativa a los procedimientos clásicos de construcción de UoLs. De esta forma, el diseñador puede ir añadiendo mayor complejidad a un diseño básico inicial, simulando distintas ejecuciones y añadiendo las correspondientes adaptaciones.

6.3. Carencias y limitaciones

Tal y como se mencionó en el capítulo 2, varias son las especificaciones que satisfacen la definición de EML según el CEN [74], presentando cada una de ellas un enfoque distinto a la hora de proporcionar medios para la descripción del contenido y modo de procesamiento de las unidades de aprendizaje. Sin embargo, en la práctica, su uso y aplicación queda prácticamente restringido al ámbito de la investigación o al entorno de una herramienta concreta. De entre todas ellas, es sin duda el IMS LD el lenguaje de modelado más completo y de más aceptación en la actualidad [33, 4]. Por este motivo, a la hora de realizar esta propuesta, se optó por utilizar la especificación del IMS LD

como guía para el desarrollo y validación de la solución de tal forma que su aplicación sobre procesos descritos mediante la misma fuese lo más sencilla posible.

Por tanto, la principal limitación del modelo de adaptación propuesto reside principalmente en la necesidad de adaptarlo a las características propias de la especificación del EML concreto, en caso de tratarse de una propuesta distinta al IMS LD. Dependiendo de la propia especificación y de la manera en que el modelo de información sea descrito, el grado necesario de adecuación de la solución variará. Así, resultará sencillo trasladar el modelo a cualquier especificación que, de manera análoga a la del modelo de información del IMS LD, organice de forma jerárquica los elementos utilizados para describir los procesos de aprendizaje. Por ejemplo, para aplicarlo al caso de PALO se debe tener en cuenta que no existe un único modelo de información genérico sino que se dispone de un conjunto de plantillas instruccionales, pero por otra parte, dado que cada plantilla tiene asociada una determinada DTD, sus elementos se encontrarán organizados siguiendo una estructura arborescente. Teniendo en cuenta esto, la aplicación del modelo de adaptación para este caso concreto de EML deberá simplemente considerar que la selección de características adaptables y observables se deberá realizar para cada una de las plantillas a utilizar.

En cualquier caso, tal y como se observó durante la aplicación de la solución en los escenarios reales de uso descritos en el capítulo 5, la capacidad de adaptación y observación ofrecida por un sistema que implemente la solución descrita en este documento estará limitada tanto por la especificación del EML utilizada para la descripción del proceso, como por el motor de ejecución empleado para interpretar la UoL. Estas limitaciones son debidas, en primer lugar, a que la descripción del proceso adaptado debe seguir siendo válida con respecto a la especificación del EML, y en segundo, a que la información que será posible recuperar acerca de los elementos de una instancia del proceso estará determinada por la información que el propio motor almacena acerca de cada uno de ellos.

Por último, mencionar que el mecanismo de adaptación y validación descrito en el documento permitirían introducir modificaciones sobre las características de cualquier elemento de un determinado diseño de aprendizaje, así como llevar a cabo cualquier tipo de cambio en su estructura. Sin embargo, tal y como se explicó en el capítulo 4, es posible que la introducción de cambios importantes en la estructura o definición del diseño implique serias modificaciones en la estrategia pedagógica implementada en el mismo. En estos casos, un enfoque más adecuado consideraría llevar a cabo un

proceso de rediseño completo del proceso en el que se estudie y analice el impacto de la modificación. Por el contrario, cuando únicamente se trata de introducir pequeñas variaciones en el diseño original, como la sustitución de recursos o la introducción de una actividad concreta, que no requieren un estudio exhaustivo del impacto del cambio, el mecanismo de adaptación aquí descrito ofrece una alternativa que permite salvar el esfuerzo asociado al rediseño completo de la UoL.

6.4. Líneas de trabajo futuras

Tanto a lo largo del desarrollo de esta tesis, como tras la finalización de la misma, se han observado posibles líneas de trabajo futuras que incluyen, por un lado, extensiones concretas al modelo de adaptación presentado y su aplicación práctica, y por otro lado, perspectivas de trabajo más generales relacionadas con la misma. En esta sección se describen las más relevantes.

6.4.1. Extensiones al trabajo realizado

En esta sección se describen algunas ampliaciones al modelo de adaptación cuya realización se está abordando o se planea abordar a corto plazo:

- Incrementar el conjunto de operadores aritméticos, relacionales y lógicos proporcionados para la descripción de las fórmulas de evaluación de los objetivos de aprendizaje y del éxito de las adaptaciones, así como para la definición de condiciones de activación. De esta forma será posible crear condiciones basadas en cambios de valores de los componentes del perfil de evaluaciones, utilizar operadores de pertenencia, describir fórmulas matemáticas más complejas, etc.
- La versión actual del modelo no permite el desarrollo de operaciones de adaptación u observación sobre elementos para los que la especificación del EML no incluya en su definición un atributo que los identifique unívocamente. Tal es el caso de los elementos *condition* y *notification* del IMS LD. Con objeto de poder implementar operaciones sobre elementos de este tipo, así como definir operaciones y observaciones sobre varios de ellos al mismo tiempo, se puede desarrollar una extensión del modelo que permita, mediante el uso de expresiones *XPath* [88], llevar a cabo una selección de los elementos sobre los que operar. De esta forma será posible actuar, por ejemplo, sobre todos los elementos que son visibles al usuario en un momento dado.

- Con objeto de permitir al instructor la especificación de evaluaciones que midan el éxito alcanzado por la acción conjunta de varias adaptaciones, así como hacer posible la definición de distintos tipos de adaptaciones cada una de las cuales posea distintos atributos y características, se puede desarrollar una extensión del modelo conceptual propuesto que incluya nuevos elementos para permitir la representación de este tipo de adaptaciones basada en la aplicación del patron de diseño *composite* [30] a la definición del elemento *poke* del modelo conceptual.
- Tal y como se explicó en el capítulo 4, la modificación de los contenidos de la UoL o de ciertas partes del proceso puede suponer la introducción de incoherencias con otras partes del proceso que deriven en problemas de comprensión en el alumno o en inconsistencias con los objetivos de aprendizaje marcados. Para evitar el problema se puede utilizar un mecanismo de validación análogo al utilizado para validar la consistencia del proceso con respecto a la especificación del EML, empleando en este caso una ontología que capture las relaciones entre las diversas partes, contenidos y objetivos de aprendizaje marcados para la UoL.

6.4.2. Perspectivas de trabajos relacionados

Con independencia de las extensiones mencionadas en la sección anterior sobre el trabajo ya realizado existen varias líneas de investigación posibles que complementarían la propuesta presentada.

6.4.2.1. Adaptación automática al contexto de procesos de aprendizaje

Tal y como se ha comentado en el capítulo 4, durante la fase de integración del método de diseño iterativo de procesos de aprendizaje se identifican aquellas adaptaciones que, después del análisis efectuado en la fase de evaluación, han demostrado introducir alguna mejora en el diseño del proceso. Sin embargo, en ocasiones, si bien algunas de las adaptaciones introducidas han demostrado implementar una solución satisfactoria al problema que las origina, dado que éste responde a circunstancias concretas del contexto de ejecución, no serán seleccionadas para su integración definitiva dentro del proceso. Estas adaptaciones, convenientemente marcadas, pueden ser almacenadas en un repositorio de tal forma que el instructor pueda hacer uso de ellas en futuras ejecuciones, en caso de detectarse una situación similar.

Asimismo, la introducción de estas adaptaciones puede ser automatizada permitiendo la adaptación del diseño del proceso de aprendizaje original a su contexto actual

de ejecución de forma transparente. Para ello, el motor de ejecución deberá ser dotado de la capacidad de detectar determinadas situaciones de contexto, acceder a la base de datos de adaptaciones, seleccionar la adaptación más adecuada a los requisitos actuales y aplicarla.

6.4.2.2. Desarrollo de un método generativo de Ingeniería Instrucciona

El mecanismo de adaptación descrito en el presente documento puede ser utilizado para desarrollar un método generativo de Ingeniería Instrucciona (IE) [73] que permita incrementar el grado de automatización en la creación y adaptación de programas de desarrollo de competencias (CDPs). Así, a partir de modelos donde queden recogidas formalizaciones de las características que el diseño puede exhibir junto con reglas de composición, conjuntos de componentes de procesos de aprendizaje y un enfoque de transformación basado en el uso de meta-modelos, los *adaptation pokes* pueden ser utilizados para soportar el ensamblado de los componentes y servicios de aprendizaje finales que satisfagan una determinada necesidad de aprendizaje [25].

6.4.2.3. Uso de diferentes EMLs para la descripción del proceso

A la hora de llevar a cabo el diseño de un proceso de aprendizaje es necesario emplear un EML que sea suficientemente expresivo para poder capturar todas las características del proceso que deseamos implementar [14]. En ocasiones, sin embargo, puede ser difícil encontrar un EML que satisfaga todos los requisitos de expresión del proceso o bien, por motivos de compatibilidad, los autores pueden verse forzados a usar un determinado EML que puede no ser el más adecuado.

En estos casos, sería deseable ofrecer al diseñador la posibilidad de salvar las carencias expresivas de un determinado EML empleando, allá donde fuese necesario, aquel EML que más se adecúe a sus necesidades. Así, sería posible llevar a cabo la descripción de las adaptaciones de un determinado proceso de aprendizaje empleando el EML más adecuado a las características de la adaptación, la cual, dependiendo de las capacidades del motor de ejecución, podría ser aplicada de varias maneras. En primer lugar, si el motor de ejecución fuese capaz de interpretar y procesar primitivas de los dos EMLs combinadas entre sí, la adaptación podría ser directamente introducida en la definición del proceso. Por otra parte, si el motor únicamente fuese capaz de interpretar primitivas del EML correspondiente a la definición original del proceso, las primitivas de la adaptación deberían ser traducidas previamente a su aplicación.

Apéndice A

Traducción a XML Schema

El modelo de información previamente descrito ha sido presentado de una manera general y abstracta con objeto de permitir distintos tipos de implementación.

El objetivo de este capítulo es ofrecer una representación de ese mismo modelo empleando la notación propia del XML *schema*, es decir, mediante un conjunto de elementos XML. El motivo por el cual se ha escogido el lenguaje de marcado XML para llevar a cabo esta representación se debe a que es este mismo lenguaje el empleado por el IMS Learning Consortium para ofrecer una descripción funcional e implementable de la mayoría de sus estándares y, en concreto, el que se ha seguido en el caso particular del IMS Learning Design.

El presente capítulo está compuesto de una primera sección en la que se ofrece una breve introducción al lenguaje XML y XML Schema y una segunda en la cual mediante gráficos, descripciones en lenguaje natural y ejemplos se detallan los distintos elementos que componen la traducción a XML Schema del modelo de información propuesto en el capítulo 4.

A.1. XML y XML Schema

XML es el acrónimo de *eXtensible Markup Language* o lenguaje de marcado extensible y proviene del SGML (*Standard General Markup Language*) o lenguaje estándar de marcado generalizado. Este lenguaje fue normalizado por la ISO en 1986 y ofrece un sistema para definir lenguajes para dar formato a documentos, es decir, permite especificar un conjunto de reglas de etiquetado que un tipo de documento debe satisfacer. Así, el HTML es un lenguaje de formato de documentos definido a partir del SGML que se creó con el fin de describir documentos para la WWW. Dado que el HTML no ha llegado a cumplir los requisitos que la creciente sociedad de Internet ha

ido demandando, el W3C comenzó en 1998 el desarrollo del XML, un nuevo estándar con una funcionalidad similar a la del SGML pero más sencillo. XML no es por tanto un lenguaje particular sino una manera de definir lenguajes de marcado.

Los documentos XML están formados por uno o más *elementos*, cuyos límites suelen ir delimitados por etiquetas de comienzo y, en caso de tener contenido, etiquetas de fin. Cada *elemento* viene identificado por un nombre, puede contener otros *elementos* o tipos de datos y puede tener definidos valores para un determinado conjunto de *atributos* o propiedades.

Se dice que un documento XML es válido si su estructura se corresponde con la definida en un documento externo que puede ser expresado como DTD o *XML Schema*. Cada *Schema* define la forma y el contenido de los documentos XML que se validan contra él. Para poder describir instancias del modelo de información propuesto en este capítulo mediante XML es necesario traducir dicho modelo a un conjunto de XML *Schemas* que definan los elementos del mismo y capturen las restricciones y relaciones existente entre ellos.

A.2. Descripción del modelo de información

A.2.1. Elementos relacionados con los procesos de evaluación

En esta sección se describe la representación en XML de los elementos del modelo relacionados con los procesos de evaluación.

A.2.1.1. Elemento <Evaluations-Profile>

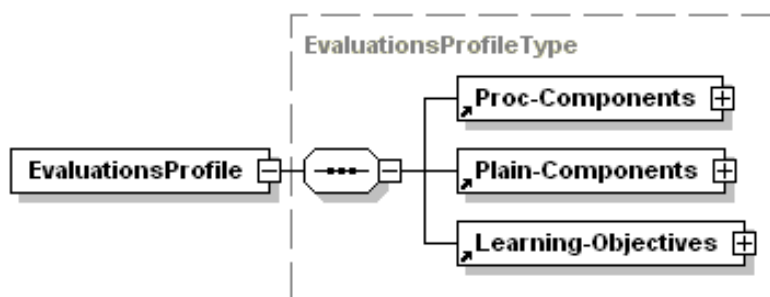


Figura A.1: Elemento Evaluations-Profile

Descripción: Describe un modelo para la evaluación de un determinado proceso de aprendizaje.

Multiplicidad: Si bien lo habitual sería definir un único perfil de evaluación para cada UoL se pueden definir tantos perfiles de evaluaciones distintos como instancias de una UoL se encuentren en ejecución

Atributos:

- **Identificador (obligatorio):** Identificador único. Tipo de datos = ID
- **Instancia (opcional):** Identifica la instancia de la UoL a la cual se aplica el perfil de evaluación. Este atributo puede ser dejado intencionalmente en blanco durante la etapa de diseño de tal forma que sea el motor de ejecución el que genere tantas instancias del perfil, como ejecuciones existan de la UoL cargando aquí el correspondiente identificador de cada una. Tipo de datos = IDRef-EML.

Elementos:

- Proc-Components
- Plain-Components
- Learning-Objectives

A.2.1.1.1. <Proc-Components>

- **Descripción:** Elemento que contiene las descripciones de componentes de proceso que afectan a la adquisición de los objetivos de aprendizaje.
- **Multiplicidad:** El elemento <Proc-Components> es obligatorio y aparece una única vez.
- **Atributos:** Ninguno
- **Elementos:**
 - Proc-Component
- **Ejemplo:**

```

<Proc-Components>

  <Proc-Component id='PC-1' datatype='number' description='Test
    Inicial 1' initvalue='0'>
    <IdElement-Ref ref='LD-Test1' />
    <Peek-Ref id-ref='Peek1'>
      <self/>
    </Peek-Ref>
  </Proc-Component>

</Proc-Components>

```

A.2.1.1.2. <Plain-Components>

Descripción: Contiene definiciones de datos cuyos valor es utilizado en la valoración de la actuación de un alumno para un componente de proceso o objetivo de aprendizaje.

Multiplicidad: El elemento <Plain-Components> es obligatorio y aparece una única vez.

Atributos: Ninguno

Elementos:

- Internal-Info
- External-Info
- Tutor-Info

Ejemplo:

```

<Plain-Component>

  <Internal-Component id='I-I1' description='Actividad-1 visitada
    (s/n)' datatype='boolean' initvalue='false'>
    <Peek-Ref id-Ref='Peek4'>
      <self/>
    </Peek-Ref>

```

```

</Internal-Component>

<External-Component id='I-E1' description='Perfil del alumno'
                    datatype='String' command='...'>
  <Activation-condition>
    <DateExpression>
      <Time>00/00/00 00:00:00</Time>
      <Referred-From><Start/></Referred-From>
    </DateExpression>
  </Activation-condition>
</External-Component>

<Tutor-Component id='I-T1' description='Valoración participación
                    alumno' datatype='String'
                    text='Valore el grado de participación del alumno en los
                    foros del curso (Mucho / Normal / Poco) '>
</Tutor-Component>

</Plain-Component>

```

A.2.1.1.3. <Learning-Objectives>

Descripción: Contiene la descripción de los objetivos de aprendizaje especificados para el proceso.

Multiplicidad: El elemento <Learning-Objectives> es obligatorio y aparece una única vez.

Atributos: Ninguno

Elementos:

- Learning-Objective

Ejemplo:

```

<Learning-Objectives>
  <Learning-Objective id='LO-1' description='Asimilar conceptos
                    básicos' datatype='number' initialValue='0'>
    <Expression>

```


- **Identificador (obligatorio):** Identificador único dentro del modelo. Tipo de datos = ID.
- **Descripción (opcional):** Descripción en lenguaje natural del objetivo y naturaleza del componente de proceso. Tipo de datos = String.
- **Tipo de datos (obligatorio):** Tipo de datos del valor devuelto por la valoración. Tipo de datos = Token.
- **Valor inicial (opcional):** Valor inicial asignado a la valoración del componente del proceso. Tipo de datos = String.

Elementos:

- IdElement-Ref
- Peek-Ref
- Expression

Ejemplo:

```
<Proc-Component id='PC-1' description='Test Inicial 1'
                datatype='number'  initvalue='0'>
  <IdElement-Ref ref='LD-Test1' />
  <Peek-Ref id-ref='Peek1'>
    <self/>
  </Peek-Ref>
</Proc-Component>
```

A.2.1.3. Elemento `<Plain-Components>`

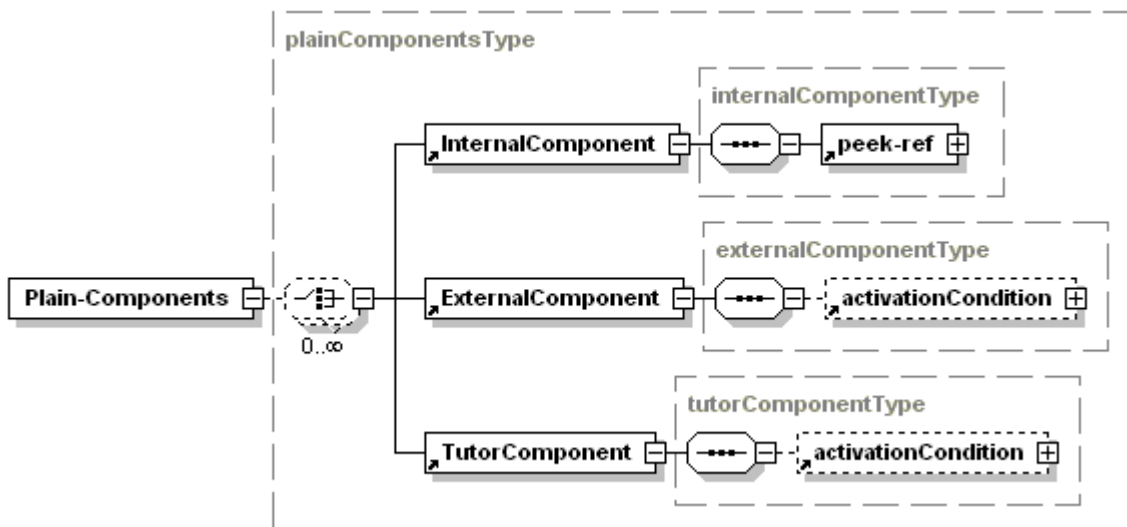


Figura A.3: Elemento Plain-Components

A.2.1.3.1. `<Internal-Component>`

Descripción: Representa un dato acerca el desarrollo del proceso que será empleado en las expresiones utilizadas para evaluar los objetivos de aprendizaje y componentes de proceso.

Multiplicidad: El elemento `<Internal-Component>` es opcional y puede aparecer n veces.

Atributos:

- **Identificador (obligatorio):** Identificador único del elemento. Tipo de datos = ID.
- **Descripción (opcional):** Descripción en lenguaje natural del elemento. Tipo de datos = String.
- **Tipo de datos (obligatorio):** Tipo de datos de la información recuperada. Tipo de datos = Token.
- **Valor inicial (opcional):** Valor inicial asignado al elemento. Tipo de datos = String.

Elementos:

- Peek-Ref

Ejemplo:

```
<Internal-Component id='I-I1' description='Actividad-1 visitada
                    (s/n)' datatype='boolean' initvalue='false'>
  <Peek-Ref id-Ref='Peek4'>
    <self/>
  </Peek-Ref>
</Internal-Component>
```

A.2.1.3.2. <External-Component>

Descripción: Representa un dato de naturaleza externa al desarrollo del proceso de aprendizaje pero necesario para especificar las evaluaciones de los objetivos de aprendizaje y componentes de proceso.

Multiplicidad: El elemento <External-Component> es opcional y puede aparecer n veces.

Atributos:

- **Identificador (obligatorio):** Identificador único del elemento. Tipo de datos = ID.
- **Descripción (opcional):** Descripción en lenguaje natural del elemento. Tipo de datos = String.
- **Tipo de datos (obligatorio):** Tipo de datos de la información recuperada. Tipo de datos = Token.
- **Valor inicial (opcional):** Valor inicial asignado al elemento. Tipo de datos = String.
- **Comando (obligatorio):** Expresión que analizada por el motor de ejecución permite recuperar de un dispositivo externo el valor a asignar al elemento. Tipo de datos = String.

Elementos:

- Activation-Condition

Ejemplo:

```

<External-Component id='I-E1' description='Perfil del alumno'
                    datatype='String' command='...'>
  <Activation-condition>
    <DateExpression>
      <Time>00/00/00 00:00:00</Time>
      <Referred-From><Start/></Referred-From>
    </DateExpression>
  </Activation-condition>
</External-Component>

```

A.2.1.3.3. <Tutor-Component>

Descripción: Representa un dato necesario para especificar las evaluaciones de los objetivos de aprendizaje y componentes de proceso y cuyo valor será introducido directamente por el instructor del proceso.

Multiplicidad: El elemento <Tutor-Component> es opcional y puede aparecer n veces.

Atributos:

- **Identificador (obligatorio):** Identificador único del elemento. Tipo de datos = ID.
- **Descripción (opcional):** Descripción en lenguaje natural del elemento. Tipo de datos = String.
- **Tipo de datos (obligatorio):** Tipo de datos de la información recuperada. Tipo de datos = Token.
- **Valor inicial (opcional):** Valor inicial asignado al elemento. Tipo de datos = String.
- **Texto (obligatorio):** Texto que será mostrado al tutor a la hora de requerir la introducción del valor. Tipo de datos = String.

Elementos:

- Activation-Condition

Ejemplo:

```

<Tutor-Component id='I-T1' description='Valoración participación
    alumno' datatype='String'
    text='Valore el grado de participación del alumno en los
    foros del curso (Mucho / Normal / Poco) '>
  <Activation-Condition>
    <Expression>
      <RelExpression>
        <RelOperator operator='equalTo' />
        <Expression>
          <BasicExpression>
            <Peek-Ref id='P-FinProceso'>
              <Self/>
            </Peek-Ref>
          </BasicExpression>
        </Expression>
      </RelExpression>
      <Expression>
        <BasicExpression>
          <LangString>true</lanstring>
        </BasicExpression>
      </Expression>
    </Expression>
  </Activation-condition>
</Tutor-Component>

```

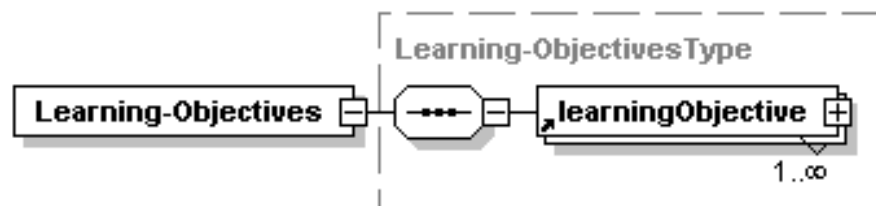
A.2.1.4. Elemento <Learning-Objectives>

Figura A.4: Elemento Learning-Objectives

A.2.1.4.1. <Learning-Objective>

Descripción: Representa un objetivo de aprendizaje que se pretende los participantes hayan alcanzado al concluir el proceso.

Multiplicidad: El elemento <Learning-Objectives> es obligatorio y al menos debe aparecer una vez.

Atributos:

- **Identificador (obligatorio):** Identificador único del elemento. Tipo de datos = ID.
- **Descripción (opcional):** Descripción en lenguaje natural del objetivo de aprendizaje. Tipo de datos = String.
- **Tipo de datos (obligatorio):** Tipo de datos de la evaluación del objetivo de aprendizaje. Tipo de datos = Token.
- **Valor inicial (opcional):** Valor inicial de la evaluación. Tipo de datos = String.

Elementos:

- Evaluation
- Relations

Ejemplo:

```
<Learning-Objective id='L0-1' description='Asimilar conceptos
    básicos' datatype='number' initialValue='0'>
  <Evaluation>
    <Activation-Condition>
      <Expression>
        <RelExpression>
          <RelOperator operator='equalTo' />
          <Expression>
            <BasicExpression>
              <Peek-Ref id='Act1-completion'>
                <Self/>
              </Peek-Ref>
            </BasicExpression>
          </Expression>
        </RelExpression>
      </Expression>
    </Activation-Condition>
  </Evaluation>
</Learning-Objective>
```

```
        </BasicExpression>
    </Expression>
    <Expression>
        <BasicExpression>
            <LangString>true</langString>
        </BasicExpression>
    </Expression>
</RelExpression>
</Expression>
</Activation-Condition>
<Expression>
    <Math-Expression>
        <Term>
            <Factor>
                <BasicExpression>
                    <ProcComponent-Ref idRef='PC-2' />
                </BasicExpression>
            </Factor>
        </Term>
    </Math-Expression>
</Expression>
</Evaluation>
<Relations>
    <ProcComponent-Ref idRef='PC-1' />
    <ProcComponent-Ref idRef='PC-2' />
</Relations>
</Learning-Objective>
```

A.2.1.5. Elemento <Learning-Objective>

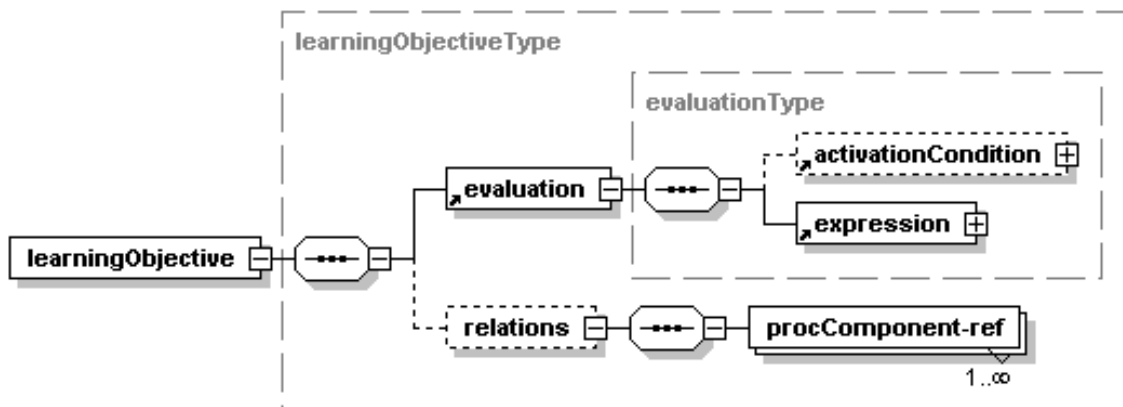


Figura A.5: Elemento Learning-Objective

A.2.1.5.1. <Evaluation>

Descripción: Define una fórmula para evaluar la consecución de un objetivo de aprendizaje o el éxito de una adaptación, y especifica el momento en el tiempo en el cual será llevada a cabo.

Multiplicidad: El elemento <Evaluation> es obligatorio y debe aparecer una única vez.

Atributos:

- **Identificador (obligatorio):** Identificador único del elemento.

Elementos:

- Activation-Condition
- Expression

Ejemplo:

```
<Evaluation>
  <Expression>
    <Math-Expression>
      <Term>
        <Factor>
          <BasicExpression>
```



```

        <ProcComponent-Ref idRef='PC-2' />
    </BasicExpression>
</Factor>
</Term>
</Math-Expression>
</Expression>
<Activation-Condition>
    <On-End />
</Activation-Condition>
</Evaluation>

```

A.2.1.5.2. <Relations>

Descripción: Especifica una lista de identificadores de componentes de proceso que pueden influir en la satisfacción del objetivo de aprendizaje.

Multiplicidad: El elemento <Relations> es opcional y puede aparecer una única vez.

Atributos: Ninguno

Elementos:

- ProcComponent-Ref

Ejemplo:

```

<Relations>
    <ProcComponent-Ref idRef='PC-1' />
    <ProcComponent-Ref idRef='PC-2' />
</Relations>

```

A.2.1.6. Elemento <Evaluation>

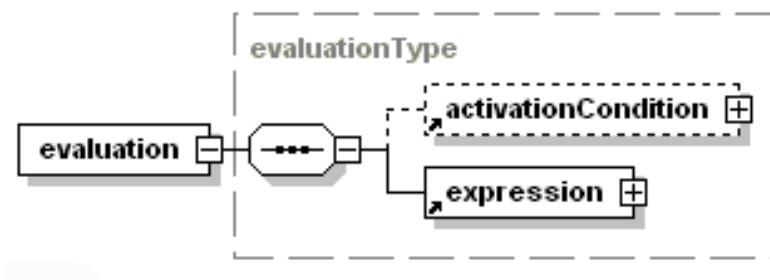


Figura A.6: Elemento Evaluation

A.2.1.6.1. <Activation-Condition>

Descripción: Condición que al ser evaluada a verdadera activará una evaluación, adaptación u observación. La condición puede definirse en base a una fecha o mediante una expresión en la que son combinados referencias a valores de componentes u objetivos de aprendizaje con operadores aritméticos, relacionales o lógicos para producir un valor.

Multiplicidad: Este elemento es opcional. En caso de no especificarse la evaluación se llevará a cabo una vez el proceso llegue a su fin.

Atributos: Ninguno

Elementos:

- DateExpression
- Expression
- On-Start
- On-End

A.2.1.6.2. <Expression>

Descripción: Combinación de referencias a valores de componentes u objetivos de aprendizaje con operadores aritméticos, relacionales o lógicos para producir un valor.

Multiplicidad: Este elemento es obligatorio. Por cada elemento <Evaluation> debe proveerse un único elemento <Expression>.

Atributos: Ninguno

Elementos:

- condition
- mathExpression
- relExpression
- logicExpression
- basicExpression

Ejemplo:

```

<Expression>
  <mathExpression>
    <term>
      <factor>
        <BasicExpression>
          <peek-ref idRef='Peek-Act1Result'>
            <UoL-ref ref='userT' />
          </peek>
        </BasicExpression>
      </factor>
    </term>
  </mathExpression>
</Expression>

```

A.2.1.7. Elemento <Activation-Condition>

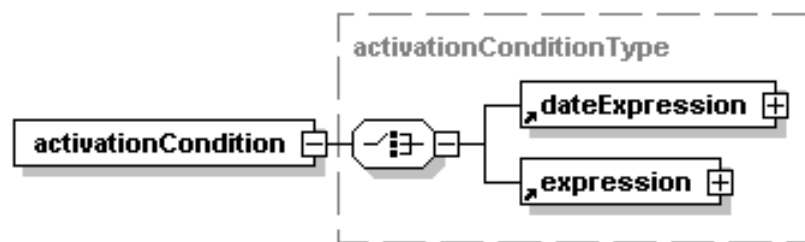


Figura A.7: Elemento Activation-Condition

A.2.1.7.1. <DateExpression>

Descripción: Determina un momento en el tiempo. Este momento puede ser una fecha exacta o bien referirse al comienzo del proceso o al momento de creación del elemento.

Multiplicidad: Este elemento es opcional. Por cada elemento <activation-condition> debe proveerse o bien un elemento <DateExpression> o bien un elemento <Expression>.

Atributos: Ninguno

Elementos:

- Time
- ReferredFrom

Ejemplo:

```
<DateExpression>  
  <Time>10/12/2006 10:00:00</Time>  
</DateExpression>
```

A.2.1.7.2. <Expression>

Ver definición elemento <Expression> anterior.

A.2.1.7.3. <On-Start>

Descripción: La activación se llevará a cabo al comienzo del proceso.

Multiplicidad: El elemento <On-Start> es opcional y puede aparecer una única vez.

Ejemplo:

```
<On-Start/>
```

A.2.1.7.4. <On-End>

Descripción: La activación se llevará al concluir el proceso.

Multiplicidad: El elemento <On-End> es opcional y puede aparecer una única vez.

Ejemplo:

```
<On-End/>
```

A.2.2. Elementos relacionados con los procesos de adaptación

En esta sección se describe la representación en XML del elemento *Poke* y sus componentes.

A.2.2.1. Elemento <Poke>

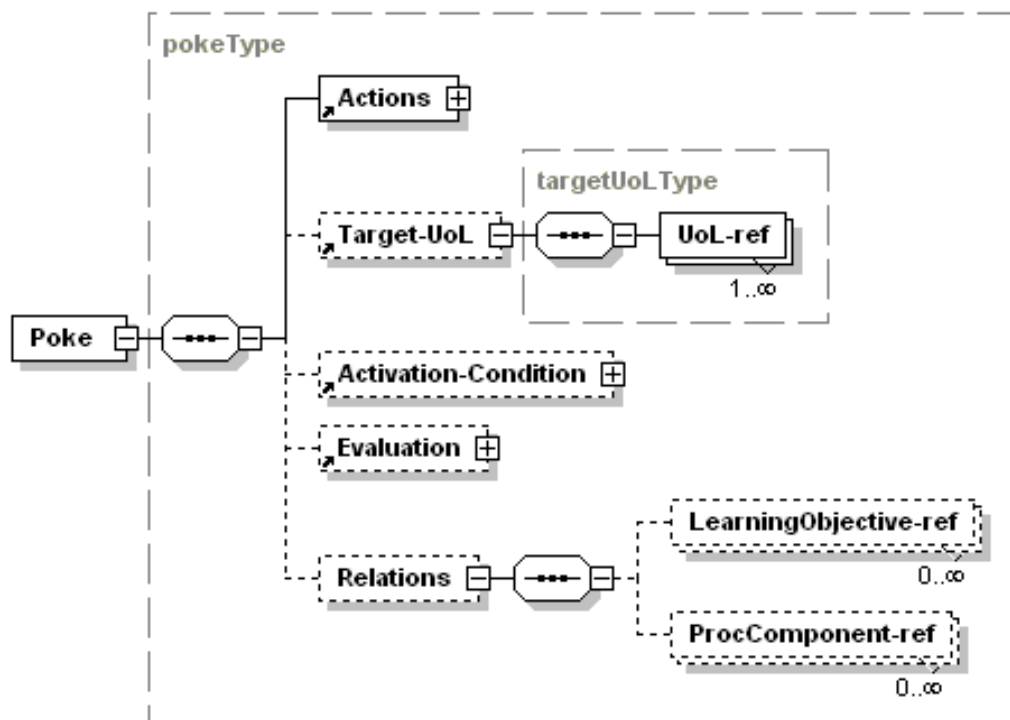


Figura A.8: Elemento Poke

Descripción: Describe una modificación sobre el diseño del proceso. Puede ser especificado tanto de manera previa a la ejecución como durante el desarrollo de la misma, indicando en último caso aquellas instancias de la UoL sobre la que debe aplicarse.

Multiplicidad: Indeterminada a este nivel.

Atributos:

- **Identificador (obligatorio):** Identificador único del elemento. Tipo de datos = ID.
- **Descripción (opcional):** Descripción en lenguaje natural del objetivo y naturaleza de la adaptación. Tipo de datos = String.

Elementos:

- Actions
- Target-UoL

- Activation-Condition
- Evaluation
- Relations

A.2.2.1.1. <Actions>

Descripción: Describe la serie de modificaciones que van a ser llevadas a cabo sobre el diseño del proceso de aprendizaje. Las acciones pueden consistir en la modificación de una característica de algún elemento del diseño, la inserción de uno nuevo o la eliminación de uno ya existente.

Multiplicidad: Por cada elemento <Poke> debe encontrarse definido un elemento <Actions>.

Atributos: Ninguno

Elementos:

- Insertion
- Deletion
- Modification

Ejemplo:

```
<actions>
  <deletion>
    <idElement-Ref ref='Actividad1-3' />
  </deletion>
  <insertion>
    <idElement-Ref ref='Nvo-Actividad' />
    <ParentElement-Ref ref='Actividad1' pos='3' />
  </insertion>
</actions>
```

A.2.2.1.2. <Target-UoL>

Descripción: Contiene la lista de referencias a las instancias de la UoL sobre las cuales debe aplicarse la modificación.

Multiplicidad: El elemento es opcional y en caso de incluirse debe aparecer una única vez. Si no aparece se supone la modificación debe ser aplicada sobre todas las instancias en ejecución de la UoL

Atributos: Ninguno

Elementos:

- UoL-ref

Ejemplo:

```
<target-UoL>
  <UoL-ref ref='User1' />
  <UoL-ref ref='User2' />
</target-UoL>
```

A.2.2.1.2.1. <UoL-Ref>

Descripción: Referencia a una instancia de la UoL.

Multiplicidad: El elemento <UoL-Ref> es obligatorio y puede aparecer repetidas veces.

Atributos:

- **ref (obligatorio):** Identificador de la instancia. Tipo de datos = IDRef-EML.

Ejemplo:

```
<UoL-ref ref='User1' />
```

A.2.2.1.3. <Activation-Condition>

Ver definición elemento <Activation-Condition> anterior.

A.2.2.1.4. <Evaluation>

Ver definición elemento <Evaluation> anterior.

A.2.2.1.5. <Relations>

Descripción: Contiene la lista de componentes de proceso y objetivos de aprendizaje que pueden verse afectados por la acción adaptativa.

Multiplicidad: El elemento es opcional y en caso de incluirse debe aparecer una única vez.

Atributos: Ninguno

Elementos:

- Proc-Component-Ref
- Learning-Objective-Ref

Ejemplo:

```
<relations>
  <ProcComponent-Ref idRef='PC-1' />
  <ProcComponent-Ref idRef='PC-2' />
  <LearningObjective-Ref idRef='LO-1' />
</relations>
```

A.2.2.2. Elemento <Actions>

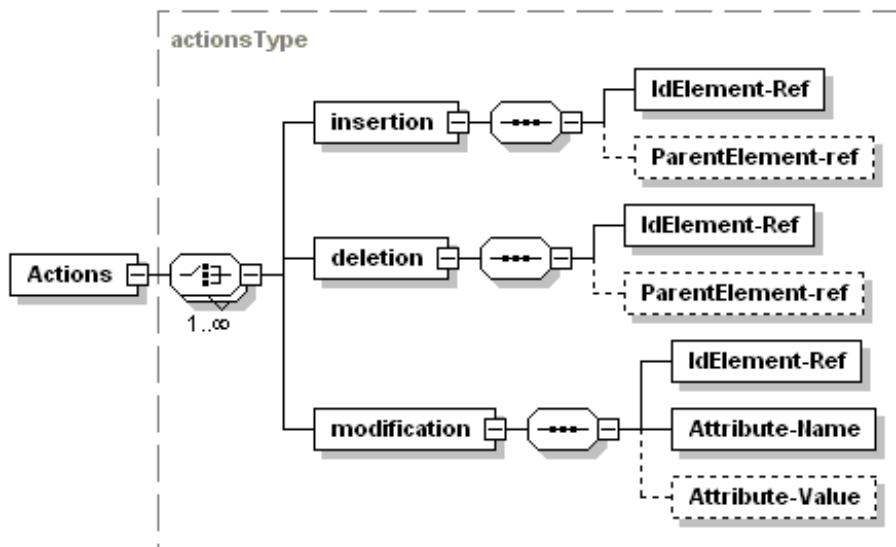


Figura A.9: Elemento Actions

A.2.2.2.1. <Insertion>

Descripción: Describe la inserción de un elemento nuevo en el diseño de aprendizaje.

Multiplicidad: Es opcional y puede aparecer repetidas veces por cada elemento <Actions>.

Atributos: Ninguno

Elementos:

- IdElement-Ref

- ParentElement-Ref

Ejemplo:

```
<insertion>
  <idElement-Ref ref='Nvo-Actividad' />
  <ParentElement-Ref ref='Actividad1' pos='3' />
</insertion>
```

A.2.2.2.1.1. <IdElement-Ref>

Descripción: Referencia al elemento de diseño que va a ser añadido.

Multiplicidad: El elemento <IdElement-Ref> es obligatorio y debe aparecer una única vez.

Atributos:

- **ref (obligatorio):** Identificador del elemento del diseño. Tipo de datos = IdRef-EML.

A.2.2.2.1.2. <ParentElement-Ref>

Descripción: Referencia a un elemento del diseño del proceso sobre el cual se va a llevar a cabo la inserción del nuevo elemento

Multiplicidad: El elemento <UoL-Ref> es opcional y puede aparecer una única vez.

Atributos:

- **posición (opcional):** Número de hijo que el nuevo elemento pasará a ocupar. Tipo de datos = Number.
- **ref (obligatorio):** Identificador del elemento del diseño. Tipo de datos = IdRef-EML.

A.2.2.2.2. <Deletion>

Descripción: Describe la supresión de un elemento del diseño de aprendizaje. Se puede eliminar únicamente una referencia concreta del elemento dentro del diseño, o por el contrario su definición junto con todas las referencias. En el primero de los casos se deberá indicar el elemento padre del que se borrará la referencia.

Multiplicidad: Es opcional y puede aparecer repetidas veces por cada elemento <Actions>.

Atributos: Ninguno

Elementos:

- IdElement-Ref
- ParentElement-Ref

Ejemplo:

```
<deletion>
  <IdElement-Ref ref='Actividad1-3' />
  <ParentElement-Ref ref='Actividad1' />
</deletion>
```

A.2.2.2.2.1. <IdElement-Ref>

Ver definición elemento <IdElement-Ref> anterior.

A.2.2.2.2.2. <ParentElement-Ref>

Ver definición elemento <ParentElement-Ref> anterior.

A.2.2.2.3. <Modification>

Descripción: Describe la modificación de una características de un elemento del diseño de aprendizaje.

Multiplicidad: Es opcional y puede aparecer repetidas veces por cada elemento <Actions>.

Atributos: Ninguno

Elementos:

- IdElement-Ref
- Attribute-Name
- Attribute-Value

Ejemplo:

```
<modification>
  <idElement-Ref ref='Actividad1' />
  <Attribute-Name>number-to-select</Attribute-Name>
  <Attribute-Value>2</Attribute-Value>
</modification>
```

A.2.2.2.3.1. <IdElement-Ref>

Ver definición elemento <IdElement-Ref> anterior.

A.2.2.2.3.2. <Attribute-Name>

Descripción: Nombre de la propiedad o característica del elemento que va a ser modificada.

Multiplicidad: El elemento <Attribute-Name> es obligatorio y debe aparecer una única vez por cada elemento <Modification>.

A.2.2.2.3.3. <Attribute-Value>

Descripción: Valor que va a ser asignado a la propiedad o característica del elemento.

Multiplicidad: El elemento <Attribute-Value> es opcional y puede aparecer una única vez por cada elemento <Modification>.

A.2.3. Elementos relacionados con los procesos de observación

En esta sección se describe la representación en XML del elemento *Peek* y sus componentes.

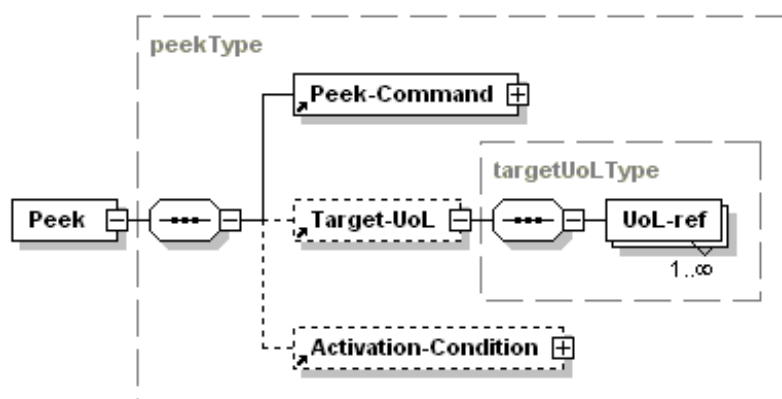
A.2.3.1. Elemento <Peek>

Figura A.10: Elemento Peek

Descripción: Define una observación a llevar a cabo sobre un elemento del diseño del proceso. Las observaciones pueden pre-definirse en tiempo de diseño o ser introducidas por el instructor durante la propia ejecución.

Multiplicidad: No determinada a este nivel.

Atributos:

- **Identificador (obligatorio):** Identificador único del elemento. Tipo de datos = ID.
- **Descripción (opcional):** Descripción en lenguaje natural del objetivo y naturaleza de la observación. Tipo de datos = String.
- **Tipo de datos (obligatorio):** Tipo de datos del valor recuperado por la observación. Tipo de datos = Token.
- **Valor inicial (opcional):** Valor inicial de la observación. Tipo de datos = String.
- **Tipo de peek (obligatorio):** Tipo de observación. Puede ser *instantánea* o *continua*.

Elementos:

- Peek-Command
- Target-UoL
- Activation-Condition

A.2.3.1.1. <peekCommand>

Descripción: Define de la observación a llevar a cabo.

Multiplicidad: Es obligatorio. Debe definirse un <peekCommand> por cada elemento <peek>.

Atributos: Ninguno

Elementos:

- IdElement-Ref
- Attribute-Name
- Characteristic-Name

Ejemplo:

```
<Peek-Command>
  <idElement-Ref ref='Actividad1' />
  <Characteristic-Name>completion-status</Characteristic-Name>
</Peek-Command>
```

A.2.3.1.2. <Target-UoL>

Descripción: Contiene la lista de referencias a las instancias de la UoL sobre las cuales debe aplicarse la observación. Su definición es análoga a la del elemento <Target-UoL> empleado en el elemento <poke>.

Multiplicidad: El elemento es opcional y en caso de incluirse debe aparecer una única vez. Si no aparece se supone la observación debe llevarse a cabo sobre todas las instancias en ejecución de la UoL.

Atributos: Ninguno

Elementos:

- UoL-ref

A.2.3.1.2.1. <UoL-Ref>

Ver definición elemento <UoL-Ref> anterior.

A.2.3.1.3. <Activation-Condition>

Ver definición elemento <Activation-Condition> anterior.

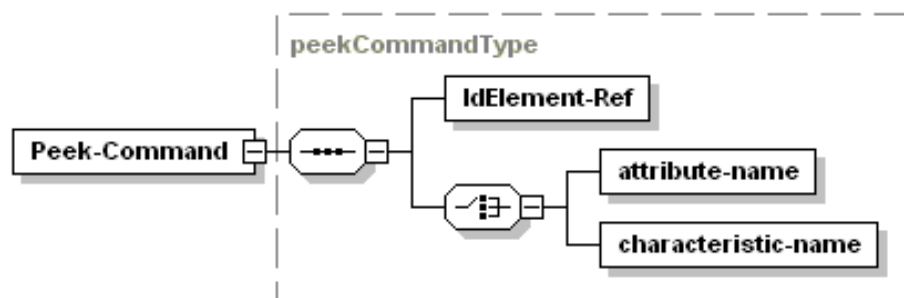
A.2.3.2. Elemento <Peek-Command>

Figura A.11: Elemento Peek-Command

A.2.3.2.1. <IdElement-Ref>

Descripción: Referencia al elemento de diseño que va a ser observado.

Multiplicidad: El elemento <IdElement-Ref> es obligatorio y debe aparecer una única vez por cada elemento <Peek-Command>.

Atributos:

- **ref (obligatorio):** Identificador del elemento del diseño. Tipo de datos = IdRef-EML.

Ejemplo:

```
<idElement-Ref ref='Test1' />
```

A.2.3.2.2. <Attribute-Name>

Descripción: Nombre del atributo del elemento cuyo valor va a ser recuperado.

Multiplicidad: El elemento es opcional.

Ejemplo:

```
<Attribute-Name>number-to-select</Attribute-Name>
```

A.2.3.2.3. <Characteristic-Name>

Descripción: Nombre de la característica del elemento cuyo valor va a ser recuperado.

Multiplicidad: El elemento es opcional.

Ejemplo:

```
<Characteristic-Name>  
    number-of-times-visited  
</Characteristic-Name>
```

A.2.4. Elementos comunes a todos los procesos

En esta sección se describe la representación en XML de distintos elementos cuyo empleo es común tanto en los procesos de evaluación, como en los de adaptación y observación.

A.2.4.1. Elemento <Expression>

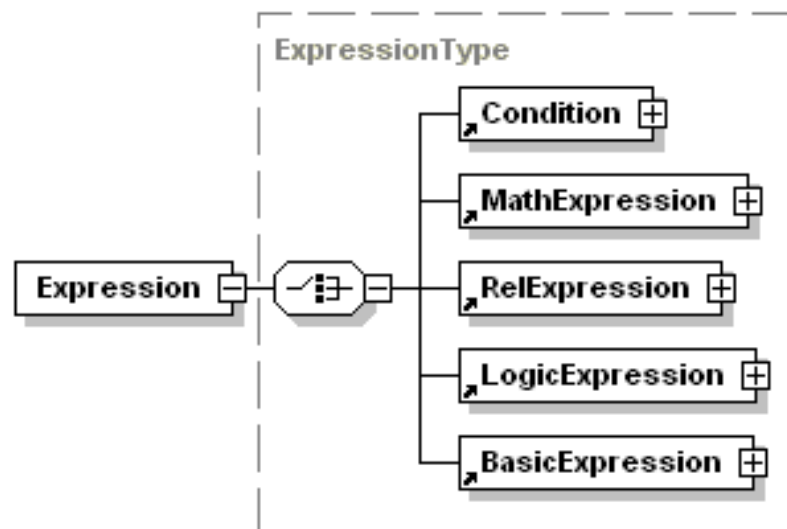


Figura A.12: Elemento Expression

A.2.4.1.1. <Condition>

Descripción: Especifica una condición que viene definida por una expresión lógica o relacional. Según el resultado de la evaluación de la condición sea verdadero o falso se pasará a procesar la segunda expresión o la tercera respectivamente.

Multiplicidad: Este elemento es opcional y puede aparecer una única vez a este nivel por cada elemento <Expression>.

Atributos: Ninguno

Elementos:

- RelExpression
- LogicExpression
- Expression
- Expression

Ejemplo:

```
<Condition>
  <RelExpression>
    <RelOperator operator='equalTo' />
```

```

    <Expression>
      <PeekRef idRef="P-CompletionAct1">
        <UoL-Ref ref="User1" />
      </PeekRef>
    </Expression>
  <Expression>
    <BasicExpression>
      <LangString>>true</LangString>
    </BasicExpression>
  </Expression>
</RelExpression>
<Expression>
  <BasicExpression>
    <PeekRef idRef="P-ResultAct1">
      <UoL-Ref ref="User1" />
    </PeekRef>
  </BasicExpression>
</Expression>
<Expression>
  <BasicExpression>
    <LangString>0</LangString>
  </BasicExpression>
</Expression>
</Condition>

```

A.2.4.1.2. <MathExpression>

Descripción: Describe una expresión matemática.

Multiplicidad: Este elemento es opcional y puede aparecer una única vez a este nivel por cada elemento <expression>.

Atributos: Ninguno

Elementos:

- Term
- Sum

- Subtract

Ejemplo:

```

<MathExpression>
  <Term>
    <Factor>
      <BasicExpression>
        <LangString>2<LangString/>
      </BasicExpression>
    </Factor>
  </Term>
  <Sum>
    <Term>
      <Factor>
        <BasicExpression>
          <LangString>4<LangString/>
        </BasicExpression>
      </Factor>
    </Term>
  </Sum>
</MathExpression>

```

A.2.4.1.3. <RelExpression>

Descripción: Describe una expresión relacional.

Multiplicidad: Este elemento es opcional y puede aparecer una única vez a este nivel por cada elemento <expression>.

Atributos: Ninguno

Elementos:

- RelOperator
- MathExpression
- MathExpression

Ejemplo:

```

<RelExpression>
  <RelOperator operator='equalTo' />
  <Expression>
    <PeekRef idRef='P-CompletionAct1'>
      <UoL-Ref ref='User1' />
    </PeekRef>
  </Expression>
  <Expression>
    <BasicExpression>
      <LangString>true</LangString>
    </BasicExpression>
  </Expression>
</RelExpression>

```

A.2.4.1.4. <LogicExpression>

Descripción: Describe una expresión lógica.

Multiplicidad: Este elemento es opcional y puede aparecer una única vez a este nivel por cada elemento <expression>.

Atributos: Ninguno

Elementos:

- Not
- And
- Or

Ejemplo:

```

<LogicExpression>
  <Not>
    <Expression>
      <PeekRef idRef='P-CompletionAct1'>
        <UoL-Ref ref='User1' />
      </PeekRef>
    </Expression>
  </Not>

```

</LogicExpression>

A.2.4.1.5. <BasicExpression>

Descripción: Define una expresión básica formada por un único valor o referencia.

Multiplicidad: Este elemento es opcional y puede aparecer una única vez a este nivel por cada elemento <expression>.

Atributos: Ninguno

Elementos:

- LangString
- Peek-Ref
- PlainComponent-ref
- ProcComponent-ref
- LearningObjective-ref
- Time

A.2.4.2. Elemento <Condition>

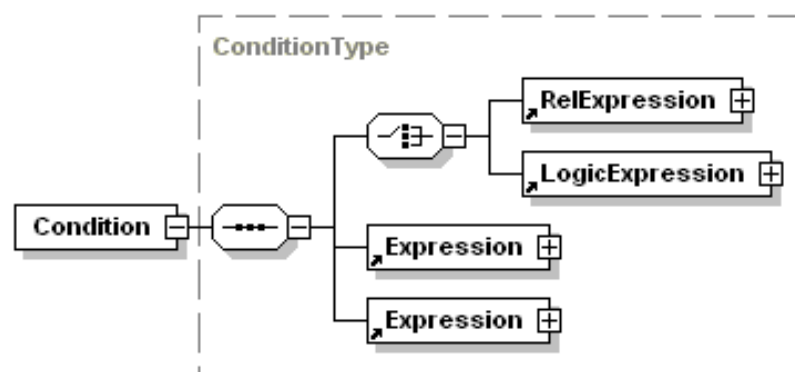


Figura A.13: Elemento Condition

A.2.4.2.1. <RelExpression>

Ver definición elemento <RelExpression> anterior.

A.2.4.2.2. <LogicExpression>

Ver definición elemento <LogicExpression> anterior.

A.2.4.2.3. <Expression>

Ver definición elemento <Expression> anterior.

A.2.4.2.4. <Expression>

Ver definición elemento <Expression> anterior.

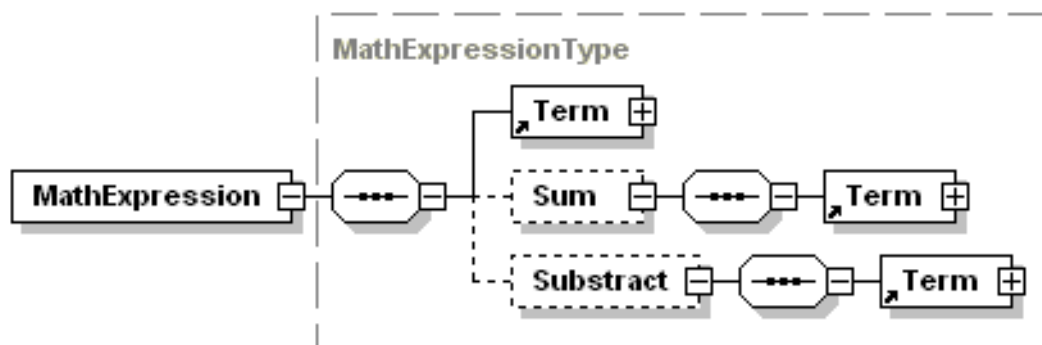
A.2.4.3. Elemento <MathExpression>

Figura A.14: Elemento MathExpression

A.2.4.3.1. <Term>

Descripción: Describe un termino dentro de una expresión matemática.

Multiplicidad: Este elemento es obligatorio y aparecerá una única vez a este nivel por cada elemento <mathExpression>.

Atributos: Ninguno

Elementos:

- Factor
- Mult
- Div

Ejemplo:

<Term>

<Factor>

```

<BasicExpression>
  <LangString>4</LangString>
</BasicExpression>
<Mult>
  <Factor>
    <MathExpression>
      <Term>
        <Factor>
          <BasicExpression>
            <LangString>4</LangString>
          </BasicExpression>
        </Factor>
      </Term>
    <Sum>
      <Term>
        <Factor>
          <BasicExpression>
            <LangString>7</LangString>
          </BasicExpression>
        </Factor>
      </Term>
    </Sum>
  </MathExpression>
</Factor>
</Mult>
</Factor>
</Term>

```

A.2.4.3.2. <Sum>

Descripción: Operador suma. Contiene un termino que se sumará al primer termino la expresión matemática.

Multiplicidad: Este elemento es opcional y podrá aparecer una única vez a este nivel por cada elemento <mathExpression>.

Atributos: Ninguno

Elementos:

- term

Ejemplo:

```

<Sum>
  <Term>
    <Factor>
      <BasicExpression>
        <LangString>4</LangString>
      </BasicExpression>
    </Factor>
  </Term>
</Sum>

```

A.2.4.3.3. <Subtract>

Descripción: Operador resta. Contiene un termino que se restará al primer termino la expresión matemática.

Multiplicidad: Este elemento es opcional y podrá aparecer una única vez a este nivel por cada elemento <mathExpression>.

Atributos: Ninguno

Elementos:

- term

Ejemplo:

```

<Subtract>
  <Term>
    <Factor>
      <BasicExpression>
        <LangString>4</LangString>
      </BasicExpression>
    </Factor>
  </Term>
</Subtract>

```

A.2.4.4. Elemento <Term>

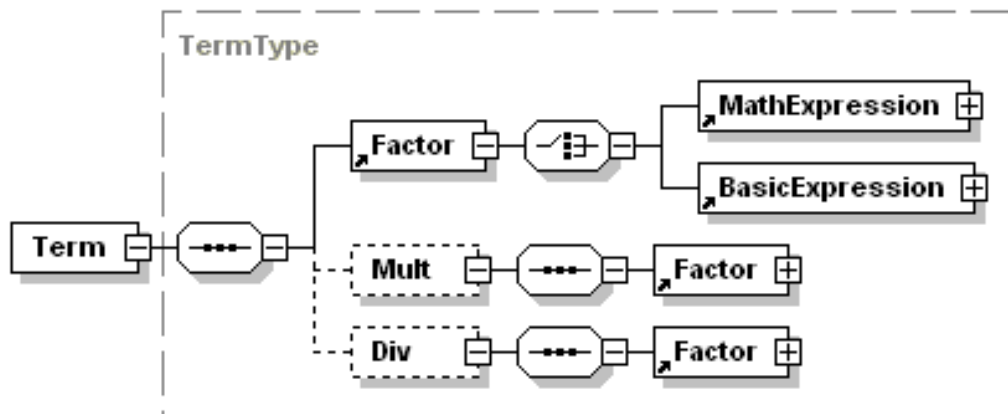


Figura A.15: Elemento TermExpression

A.2.4.4.1. <Factor>

Descripción: Describe un factor dentro de un termino de una expresión matemática.

Multiplicidad: Este elemento es obligatorio y aparecerá una única vez a este nivel por cada elemento <Term>.

Atributos: Ninguno

Elementos:

- MathExpression
- BasicExpression

Ejemplo:

```
<Factor>
  <BasicExpression>
    <LangString>4</LangString>
  </BasicExpression>
</Factor>
```

A.2.4.4.2. <Mult>

Descripción: Operador multiplicación. Contiene un factor que será multiplicado al primer factor del termino.

Multiplicidad: Este elemento es opcional y podrá aparecer una única vez a este nivel por cada elemento <Term>.

Atributos: Ninguno

Elementos:

- Factor

Ejemplo:

```
<Mult>
  <Factor>
    <BasicExpression>
      <LangString>4</LangString>
    </BasicExpression>
  </Factor>
</Mult>
```

A.2.4.4.3. <Div>

Descripción: Operador división. Contiene un factor que dividirá al primer factor del termino.

Multiplicidad: Este elemento es opcional y podrá aparecer una única vez a este nivel por cada elemento <Term>.

Atributos: Ninguno

Elementos:

- Factor

Ejemplo:

```
<Div>
  <Factor>
    <BasicExpression>
      <LangString>4</LangString>
    </BasicExpression>
  </Factor>
</Div>
```


A.2.4.5. Elemento <RelExpression>

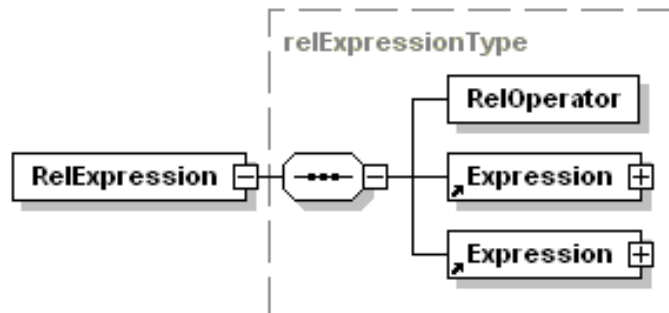


Figura A.16: Elemento RelExpression

A.2.4.5.1. <RelOperator>

Descripción: Operador de la expresión relacional.

Multiplicidad: Este elemento es obligatorio y aparecerá una única vez a este nivel por cada elemento <RelExpression>.

Atributos: Ninguno

- **operator (obligatorio):** Operador de la expresión relacional. Tipo de datos = Token.

Elementos: Ninguno

Ejemplo:

```
<RelOperator operator="notEqualTo" />
```

A.2.4.5.2. <Expression>

Ver definición elemento <Expression> anterior.

A.2.4.5.3. <Expression>

Ver definición elemento <Expression> anterior.

A.2.4.6. Elemento <LogicExpression>

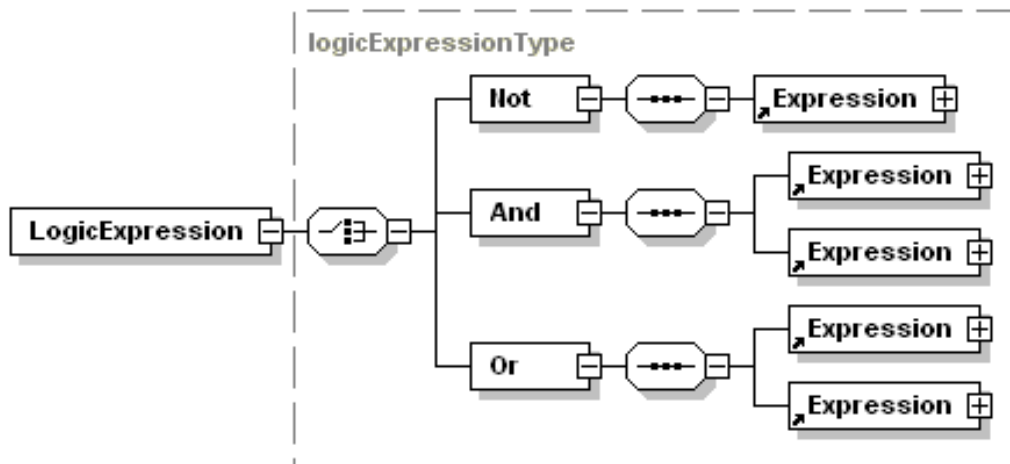


Figura A.17: Elemento LogicExpression

A.2.4.6.1. <Not>

Descripción: Operador negación.

Multiplicidad: Este elemento es opcional y puede aparecer una única vez a este nivel por cada elemento <Logic-Expression>.

Atributos: Ninguno

Elementos:

- Expression

Ejemplo:

```
<Not>
  <Expression>
    <PeekRef idRef='P-CompletionAct1'>
      <UoL-Ref ref='User1' />
    </PeekRef>
  </Expression>
</Not>
```

A.2.4.6.2. <And>

Descripción: Operador *and*.

Multiplicidad: Este elemento es opcional y puede aparecer una única vez a este nivel por cada elemento <Logic-Expression>.

Atributos: Ninguno

Elementos:

- Expression
- Expression

Ejemplo:

```
<And>
  <Expression>
    <PeekRef idRef='P-CompletionAct1'>
      <UoL-Ref ref='User1' />
    </PeekRef>
  </Expression>
  <Expression>
    <PeekRef idRef='P-CompletionAct1'>
      <UoL-Ref ref='User2' />
    </PeekRef>
  </Expression>
</And>
```

A.2.4.6.3. <Or>

Descripción: Operador *or*.

Multiplicidad: Este elemento es opcional y puede aparecer una única vez a este nivel por cada elemento <Logic-Expression>.

Atributos: Ninguno

Elementos:

- Expression
- Expression

Ejemplo:

```

<Or>
  <Expression>
    <PeekRef idRef='P-CompletionAct1'>
      <UoL-Ref ref='User1' />
    </PeekRef>
  </Expression>
  <Expression>
    <PeekRef idRef='P-CompletionAct1'>
      <UoL-Ref ref='User2' />
    </PeekRef>
  </Expression>
</Or>

```

A.2.4.7. Elemento <BasicExpression>

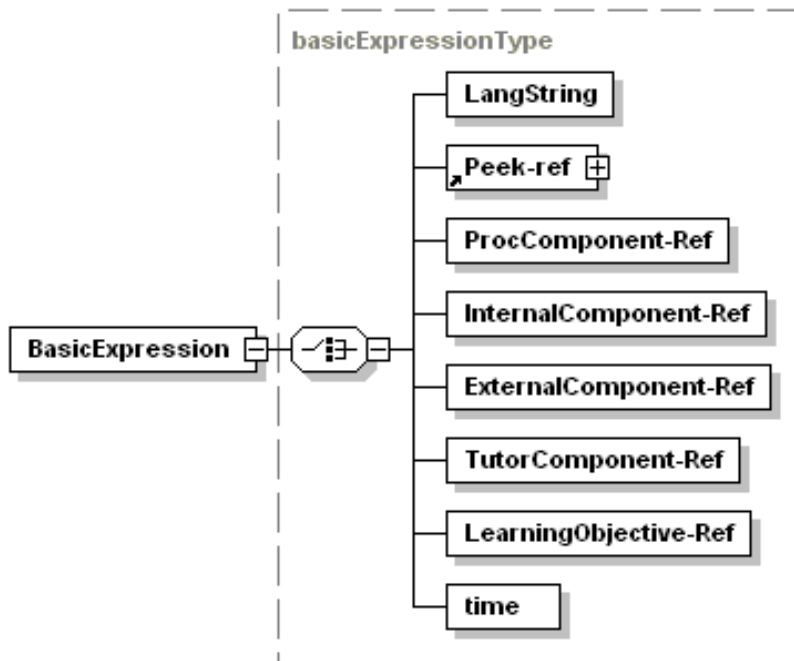


Figura A.18: Elemento BasicExpression

A.2.4.7.1. <LangString>

Descripción: Especifica un valor introducido como un String.

Multiplicidad: Este elemento es opcional y puede aparecer una única vez a este nivel por cada elemento `<basicExpression>`.

Atributos: Ninguno

Ejemplo:

```
<basicExpression>
  <langString>8</langString>
</basicExpression>
```

A.2.4.7.2. `<Peek-Ref>`

Descripción: Referencia a uno o varios valores recuperados por una observación concreta. Así, se puede referir al valor recuperado por la observación sobre la propia instancia (*self*) o sobre instancias de otros participantes del proceso. En este último caso se deberán incluir los identificadores de dichas instancias. instancia.

Multiplicidad: El elemento `<Peek-Ref>` es opcional y sólo puede aparecer una única vez.

Atributos:

- **idRef (obligatorio):** Identificador de la observación. Tipo de datos = ID.

Elementos:

- self
- UoL-Ref

Ejemplo:

```
<Peek-Ref id-ref='Peek1'>
  <UoL-ref ref='User1' />
  <UoL-ref ref='User2' />
  <UoL-ref ref='User3' />
</Peek-Ref>
```

A.2.4.7.3. <InternalComponent-Ref>

Descripción: Referencia a un componente interno definido en el perfil de evaluaciones.

Multiplicidad: El elemento <InternalComponent-Ref> es opcional y sólo puede aparecer una única vez.

Atributos:

- **idref (obligatorio):** Identificador del componente interno. Tipo de datos = IDRef.

Ejemplo:

```
<InternalComponent-Ref idRef='I-testResult' />
```

A.2.4.7.4. <ExternalComponent-Ref>

Descripción: Referencia a un componente externo definido en el perfil de evaluaciones.

Multiplicidad: El elemento <ExternalComponent-Ref> es opcional y sólo puede aparecer una única vez.

Atributos:

- **idref (obligatorio):** Identificador del componente externo. Tipo de datos = IDRef.

Ejemplo:

```
<ExternalComponent-Ref idRef='E-userdata' />
```

A.2.4.7.5. <TutorComponent-Ref>

Descripción: Referencia a un componente interno definido en el perfil de evaluaciones.

Multiplicidad: El elemento <TutorComponent-Ref> es opcional y sólo puede aparecer una única vez.

Atributos:

- **idref (obligatorio):** Identificador del componente de tutor. Tipo de datos = IDRef.

Ejemplo:

```
<TutorComponent-Ref idRef='T-tutoropinion' />
```

A.2.4.7.5.1. <ProcComponent-Ref>

Descripción: Referencia a un componente de proceso definido en el perfil de evaluaciones.

Multiplicidad: El elemento <Proc-component-Ref> es opcional y sólo puede aparecer una única vez.

Atributos:

- **idref (obligatorio):** Identificador del componente. Tipo de datos = IDRef.

Ejemplo:

```
<ProcComponent-Ref idRef='PC-theory' />
```

A.2.4.7.5.2. <LearningObjective-Ref>

Descripción: Referencia a un objetivo de aprendizaje definido en el perfil de evaluaciones.

Multiplicidad: El elemento <LearningObjective-Ref> es opcional y sólo puede aparecer una única vez.

Atributos:

- **idref (obligatorio):** Identificador del learning objective. Tipo de datos = IDRef.

Ejemplo:

```
<LearningObjective-Ref idRef='L01' />
```

A.2.4.7.6. <Time>

Descripción: Representa un momento en el tiempo. El formato empleado es dd/mm/yyyy hh24:mi:ss.

Multiplicidad: Este elemento es obligatorio. Por cada elemento <DateExpression> debe proveerse un elemento <Time>.

Elementos:

- Time
- ReferredFrom

Ejemplo: `<Time>10/12/2006 10:00:00</Time>`

A.2.4.8. Elemento `<Peek-Ref>`

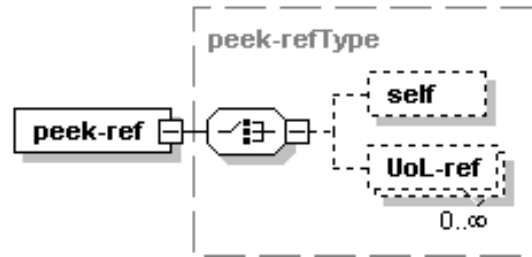


Figura A.19: Elemento Peek-Ref

A.2.4.8.1. `<Self>`

Descripción: Indica que la observación se refiere a la misma instancia propia del usuario.

Multiplicidad: El elemento `<self>` es opcional y puede aparecer una única vez.

Ejemplo:

`<self/>`

A.2.4.8.2. `<UoL-Ref>`

Ver definición elemento `<UoL-Ref>` anterior.

A.2.4.9. Elemento `<DateExpression>`

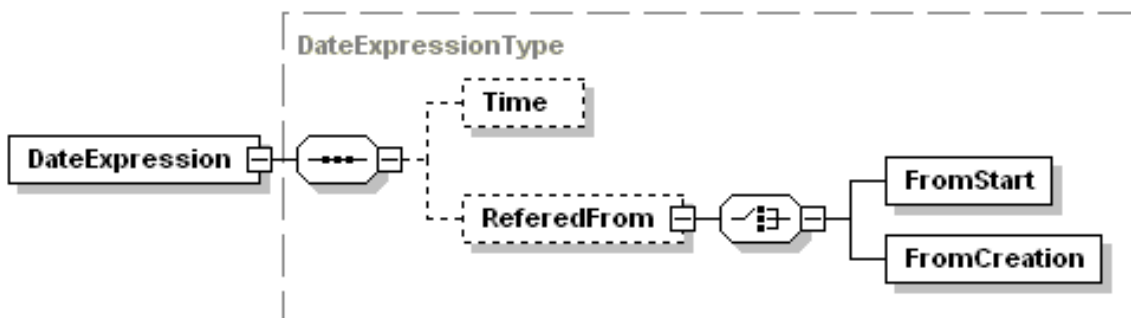


Figura A.20: Elemento DateExpression

A.2.4.9.1. <Time>

Ver definición anterior elemento <Time>.

A.2.4.9.2. <ReferredFrom>

Descripción: Especifica si la expresión temporal debe de tomarse referida al momento de comienzo o al momento de creación del elemento. En caso de no especificarse la expresión temporal debe tratarse como una fecha del calendario.

Multiplicidad: Este elemento es opcional.

Elementos:

- Start
- Creation

Ejemplo:

```
<ReferredFrom>  
  <Creation/>  
</ReferredFrom>
```

A.2.4.9.2.1. <Creation>

Descripción: Especifica que la expresión temporal debe de tomarse referida al momento de creación del elemento.

Multiplicidad: Este elemento es obligatorio.

A.2.4.9.2.2. <Start>

Descripción: Especifica que la expresión temporal debe de tomarse referida al momento de inicio del proceso.

Multiplicidad: Este elemento es obligatorio.

Apéndice B

Notación de tablas y diagramas

En este apéndice se recoge el formato empleado en la descripción de los diagramas y tablas del modelo de información.

B.1. Formato de los diagramas

- Únicamente se muestran elementos (no atributos)
- Los diagramas son estructuras en árbol que son leídas de izquierda a derecha. Un elemento situado a la izquierda contiene los elementos a su derecha. El elemento situado más a la izquierda es la raíz del árbol.
- Únicamente se representan tres árboles en todo el documento: un árbol con la raíz en el elemento *EvaluationsProfile*, otro con raíz en el elemento *Poke* y otro con raíz en el elemento *Peek*. Cada árbol es descompuesto sucesivamente con objeto de facilitar la presentación de sus elementos. Esta descomposición se lleva a cabo de izquierda a derecha y de la parte superior a la parte inferior, salvo el elemento común a los tres árboles, *Expression*, que por motivos de claridad es descompuesto en último lugar. Los elementos que son completamente descompuestos en sus elementales no requieren de nuevas explicaciones en subsecuentes diagramas.
- Una relación OR se representa en el diagrama mediante <
- Una relación AND se representa en el diagrama mediante [
- El símbolo * denota que el elemento aparece cero o más veces en su contenedor
- El símbolo + denota que el elemento aparece una o más veces en su contenedor
- El símbolo ? denota que el elemento es opcional

- Cuando ninguno de los símbolos (*, +, ?) aparece delante del nombre del elemento significa que el elemento aparece una única vez

B.2. Formato de las tablas

- No.: Número del elemento en la jerarquía.
- Nombre: Nombre del elemento o atributo. Los atributos son especificados en cursiva.
- Descripción: Texto descriptivo de la función del elemento.
- Obl.: Indica si el elemento es obligatorio (S) o opcional (N).
- M.: Indica la multiplicidad del elemento. Puede adoptar los siguientes valores:
 - 1: el elemento aparece una sola vez.
 - 0..1: el elemento aparece una o ninguna vez.
 - 0..n: el elemento puede aparecer de cero a n veces.
 - 1..n: el elemento puede aparecer de 1 a n veces.
 - -: La multiplicidad no está determinada. Se empleará en la definición de los elementos de nivel más alto dentro de la tabla.
- Tipo: Indica el tipo del elemento o atributo. Puede adoptar los siguientes valores:
 - Cont. o Contenedor: El elemento se emplea para englobar uno o más elementos del mismo tipo.
 - Elec. o Elección: El elemento se descompone en una secuencia de elementos escogidos entre distintos posibles.
 - Sec. o Secuencia: El elemento se descompone en una secuencia ordenada de elementos.
 - String: El elemento es una cadena de caracteres.
 - Vacío: Elemento que no contiene más caracteres de datos.

- **Token**: el elemento únicamente puede adoptar un determinado valor dentro de un vocabulario específico.
- **ID**: el elemento es un identificador.
- **IDRef**: el elemento es una referencia a un componente de la extensión.
- **IDRef-EML**: el elemento es una referencia a un componente del diseño del proceso de aprendizaje o a una instancia del mismo.

B.3. Tipos de datos

- **Boolean**: Representa un valor binario. Valores posibles: *true* y *false*.
- **Integer**: Representación del concepto matemático de número entero. El rango de valores posibles va desde -2^{63} a 2^{63} .
- **Real**: Representación del concepto matemático de número real.
- **String**: Representa una cadena de caracteres. El máximo número de caracteres es 2000.
- **File**: Representa un fichero binario.
- **DateTime**: Representa una fecha siguiendo el formato definido por el ISO8601 [45].
- **Duration**: Representa una determinada cantidad de tiempo. El formato seguido es el mismo que el empleado en el ISO8601 [45]:
PnYnMnDTnHnMnS, donde nY representa el número de años, nM el número de meses, nD el número de días, 'T' es el carácter separador de fecha/tiempo, nH el número de horas, nM el número de minutos y nS el números de segundos.
- **Text**: Representa una cadena de caracteres larga. El máximo número de caracteres es 64000.

Todos los tipos de datos pueden tener asignado el valor *<no-value>*.

2. El modelo propuesto para la descripción de adaptaciones le parece:

	Nada	Poco	Indiferente	Bastante	Mucho
Claro					
Expresivo					
Consistente					
Util					
Reutilizable					
Original					
Comentarios					

3. En cuanto a la introducción de adaptaciones en tiempo de ejecución considera que resulta:

	Nada	Poco	Indiferente	Bastante	Mucho
Util					
Necesaria					
Original					
Sencilla					
Realizable					
Eficiente					
Comentarios					

4. ¿Cómo de útil considera poder especificar las siguientes características de las adaptaciones?

	Nada	Poco	Indiferente	Bastante	Mucho
Condiciones de activación					
Activación basada en eventos de otros usuarios					
Temporización de la adaptación					
Introducción en tiempo de ejecución					
Introducción en tiempo de publicación					
Frecuencia de la aplicación					
Fórmula para evaluar éxito					
Comentarios					

5. Considera que el modelo propuesto para la implementación de monitorizaciones y observaciones sobre el desarrollo del proceso es:

	Nada	Poco	Indiferente	Bastante	Mucho
Util					
Necesario					
Original					
Realizable					
Eficiente					
Comentarios					

6. El modelo propuesto para la descripción de las evaluaciones de las adaptaciones le parece:

	Nada	Poco	Indiferente	Bastante	Mucho
Util					
Necesario					
Expresivo					
Original					
Realizable					
Eficiente					
Comentarios					

7. El modelo propuesto para la descripción de la evaluación de del éxito del proceso le parece:

	Nada	Poco	Indiferente	Bastante	Mucho
Util					
Necesario					
Expresivo					
Original					
Realizable					
Eficiente					
Comentarios					

8. El método propuesto para la automatización de la evaluación de las adaptaciones le parece:

	Nada	Poco	Indiferente	Bastante	Mucho
Util					
Necesario					
Expresivo					
Original					
Realizable					
Eficiente					
Comentarios					

9. El método propuesto para la automatización de la evaluación del éxito del proceso le parece:

	Nada	Poco	Indiferente	Bastante	Mucho
Util					
Necesario					
Expresivo					
Original					
Realizable					
Eficiente					
Comentarios					

10. El método de diseño iterativo de procesos educativos considera que es:

	Nada	Poco	Indiferente	Bastante	Mucho
Util					
Necesario					
Práctico					
Realizable					
Eficiente					
Comentarios					

11. La arquitectura presentada para la extensión de motores de ejecución de procesos de aprendizaje especificados mediante EMLs considera que es:

	Nada	Poco	Indiferente	Bastante	Mucho
Flexible					
Extensible					
Independiente del EML					
Intrusiva					
Eficiente					
Comentarios					

12. El mecanismo para la validación de diseños de procesos educativos adaptados y especificados mediante EML considera que es:

	Nada	Poco	Indiferente	Bastante	Mucho
Util					
Necesario					
Flexible					
Realizable					
Eficiente					
Comentarios					

13. Señale las principales deficiencias del modelo, en caso de que haya encontrado alguna.

Principales deficiencias

14. Señale las principales ventajas que la utilización del modelo supondría en su/s área/s de interés.

Principales ventajas

¿Estaría dispuesta/o a utilizar un sistema que implementase este modelo?

Sí No

Gracias por su colaboración.

Apéndice D

Resultados de la evaluación

D.1. Resultados de la evaluación

Tabla D.1: Intereses de los evaluadores

Id	Educación	Entornos	Ingeniería	Sistemas	Otros
1	S	S	-	S	
2	S	S	-	-	
3	-	S	-	-	
4	S	-	-	S	
5	S	S	S	S	
6	S	S	-	-	
7	-	S	S	S	
8	S	S	S	-	
9	-	-	-	-	Human-Computer Interaction
10	-	-	-	-	
11	S	S	-	-	
12	-	-	S	S	Accesibilidad y Diseño Universal
13	-	S	S	-	
14	-	S	S	-	
15	S	S	S	S	

Tabla D.2: Categorías de los evaluadores

Id	Experto	Diseñador	Instructor	Estudiante
1	S	-	-	-
2	S	S	-	-
3	S	-	-	-
4	S	S	S	-
5	S	-	-	-
6	S	S	S	-
7	S	-	-	-
8	S	-	-	-
9	S	-	S	-
10	-	-	S	-
11	-	-	-	S
12	S	S	S	-
13	-	S	S	-
14	-	-	-	S
15	S	-	-	-

Tabla D.3: Respuestas a la pregunta 1 del cuestionario

Id	Comprensible	Apropiada	Comentarios
1	5	4	
2	4	4	
3	4	4	<p>Simplymente dos comentarios:</p> <ul style="list-style-type: none"> - El tema de si la notación es adecuada en realidad depende de la audiencia. Sólo para los usuarios familiarizados con notaciones 'técnicas' como XML consideraron la notación comprensible y adecuada - Idem para los diagramas (XML) de los distintos elementos, en ocasiones me han aconsejado a mí misma que utilice una notación más genérica para facilitar la comprensión de los mismos (para el público no familiarizado con ellos)
4	3	3	
5	5	4	
6	5	5	
7	5	4	Although the notation used in general for describing the proposed solution is appropriate and according to the literature, I have the feeling that when we are referring to "runtime adaptation" we mean adaptation of the learning process that is triggered and executed at runtime and not detection of possible modifications of the learning process to be executed in a later run.
8	4	4	Los conceptos están claros, al menos para un informático, que se aplican en la metodología. Quizás sería interesante evaluarlo por parte de diseñadores/docentes no informáticos.
9	5	4	
10	5	5	
11	4	4	En general, es posible que se eche en falta una sección de definiciones, ya que hay términos, conceptos y acrónimos que pueden resultar desconocidos al lector. Digo esto desde mi profundo desconocimiento de la estructura típica o estándar de este tipo de documentos.
12	4	4	He echado en falta una leyenda para interpretar los diagramas
13	5	5	
14	4	4	
15	4	4	

Tabla D.4: Respuestas a la pregunta 2 del cuestionario

Id	Claro	Expresivo	Consistente	Útil	Reutilizable	Original	Comentarios
1	4	5	4	5	4	5	
2	5	4	3	5	4	5	
3	4	4	4	4	5	5	Sobre todo las cuestiones del modelo plasmadas en los ejemplos. Haber visto más ejemplos del uso de pokes con activationCondition, evaluation (hay uno en la adaptación 4, pero sin explicación), relations, etc e incluso haber probado yo misma a especificar una adaptación de que me haya surgido en mi propia práctica me ayudaría en la valoración del modelo.
							Con targetUOL os referís a la instancia de la UOL (i.e. el run en IMSLID), por tanto no crea que sea el mejor término. Si no aparece, la adaptación se aplica sobre todas las instancias en ejecución, pero ¿de todas las UOL? Imagino que no, puesto que los elementos referenciados en "actions" serán los considerados en una UOL concreta.
							El uso de property-name en los cambios de tipo modificación en un primer momento me llevó a la idea de que el modelo sólo permitiría la modificación de properties (de IMSLID). ¿Por qué no utilizáis attribute-name?
4	4	3	4	3	4	2	
5	5	4	5	4	3	4	
6	4	4	5	4	4	5	
7	4	3	3	4	4	5	My impression by analysing the proposed model is that it can be reused in many different cases (learning flows). However, it seems that it fits more to the IMS Learning Design ENML language than other EMILs (e.g. PALO)
8	3	4	3	4	3	4	Los conceptos están claros, el problema es a la hora de aplicarlos a una UOL, con un mayor número de ejemplos puedes aprender de dichos diseños.
9	5	5	4	5	4	4	
10	5	5	5	5	5	4	As a researcher from outside the e-learning world, it seems to me that the provided evidence for the utility and generality of the model is reasonable.
11	4	3	3	5	4	3	
12	5	4	4	4	4	4	
13	5	5	5	5	5	4	Dado que no soy experto en el área y no conozco la documentación completa, no puedo determinar con veracidad el nivel de originalidad, sin embargo me parece novedoso en función de lo explicado.
14	4	4	3	5	5	4	
15	4	3	4	4	4	3	

Tabla D.5: Respuestas a la pregunta 3 del cuestionario

Id	Util	Necesaria	Original	Realizable	Sencilla	Eficiente	Comentarios
1	5	4	5	5	5	4	
2	5	5	4	5	4	5	
3	5	5	4	4	4	4	Lo de sencillo no lo tengo tan claro porque no lo he visto funcionar pero a priori parece que sí, otra cuestión a discutir es si sería más sencillo realizar las adaptaciones mediante el uso de herramienta de autoría (las mismas que se utilizan en la creación inicial de la UoL y con las que por tanto ya está familiarizado el usuario) integradas con los motores de EMLs. Aunque esta aproximación tiene desventajas comparado con vuestra propuesta: en muchas ocasiones no interesa modificar de verdad una UoL como tal. Las necesidades pueden ser tan distintas en cada situación que no siempre interesa que las adaptaciones permanezcan en la UoL. Es verdad que algunas mejoras pueden ser genéricas, pero otras serán tan dependientes de la situación (ej. tiempo disponible, número de alumnos participando, diferencias en cuanto a la preparación previa de los alumnos, etc.) que interesa "almacenar por separado" las adaptaciones (como decís, convenientemente etiquetadas ;) pero no integradas en la UoL
4	5	5	3	4	2	3	
5	5	5	4	3	3	4	
6	5	5	4	4	4	4	
7	4	5	3	3	3	3	The proposed solution seems feasible, however it remains to be seen how many of the adaptation parameters can be fully detected without the intervention of a human Learning Designer.
8	4	4	3	2	2	3	
9	5	4	5	5	4	3	
10	5	5	4	5	5	5	I am not completely aware of the literature and thus I cannot deeply judge the originality of this work.
11	5	4	3	4	2	3	
12	5	5	4	5	4	3	
13	5	5	5	5	5	5	
14	5	4	5	3	2	3	
15	5	4	4	4	5	5	

Tabla D.6: Respuestas a la pregunta 4 del cuestionario

Id	Condiciones	Usuarios	Temporización	Ejecución	Publicación	Frecuencia	Fórmula	Comentarios
1	5	5	5	5	5	4	5	
2	4	5	4	5	5	4	4	
3	4	4	4	5	4	3	4	¿Es posible especificar la frecuencia de la adaptación? Ahora mismo no se me ocurre un escenario donde me pueda parecer útil. Las condiciones de la adaptación en general me parecen muy útiles.
4	4	5	5	5	3	3	4	
5	3	4	4	3	4	4	5	
6	4	5	3	5	3	4	4	
7	3	5	3	4	4	5	3	
8	4	5	3	3	3	3	4	
9	5	5	4	4	4	4	5	
10	5	5	5	5	4	5	4	
11	4	3	3	5	1	1	4	Realmente, la introducción en tiempo de publicación no lo veo demasiado útil. La introducción en tiempo de ejecución es, a mi juicio, el punto más destacado.
12	4	3	5	3	4	3	4	
13	5	5	5	5	5	5	5	
14	5	5	4	5	5	3	3	
15	5	4	4	4	5	4	4	No estoy segura de a qué se refiere "frecuencia de la aplicación". La he contestado pensando en si será frecuente tener que aplicar adaptaciones.

Tabla D.7: Respuestas a la pregunta 5 del cuestionario

Id	Util	Original	Realizable	Eficiente	Necesario	Comentarios
1	5	5	4	4	4	
2	5	5	4	5	4	
3	4	4	4	4	5	Es crítico...
4	4	3	4	3	4	
5	5	5	4	4	4	
6	4	4	4	4	5	
7	4	5	3	3	5	
8	4	3	2	3	5	
9	5	4	5	3	5	
10	5	4	5	5	5	see comment to question 3
11	4	3	4	3	4	Desconozco la originalidad y eficiencia del modelo; la eficiencia pienso que dependerá mucho de la implementación o implantación del mismo
12	5	4	5	3	5	
13	5	5	5	5	4	
14	5	4	2	3	4	
15	5	3	4	3	5	

Tabla D.8: Respuestas a la pregunta 6 del cuestionario

Id	Útil	Necesario	Original	Realizable	Eficiente	Expresivo	Comentarios
1	5	5	5	5	4	4	
2	4	5	4	5	4	5	
3	4	4	4	4	4	3	Me falta información para valorar bien este modelo
4	4	4	4	4	3	4	
5	5	5	4	4	4	3	
6	4	5	5	4	4	4	
7	4	5	5	3	3	3	
8	4	4	3	3	3	4	
9	4	5	4	5	3	5	
10	5	5	4	5	5	5	see comment to question 3
11	3	2	3	4	3	3	
12	2	4	3	4	3	2	
13	5	5	4	5	5	5	Misma justificación de la respuesta 2.
14	5	4	4	3	2	3	
15	4	4	3	4	4	3	

Tabla D.9: Respuestas a la pregunta 7 del cuestionario

Id	Util	Necesario	Original	Realizable	Eficiente	Expresivo	Comentarios
1	5	5	4	5	4	4	
2	4	5	4	5	4	4	
3	4	4	4	4	4	3	No estoy muy segura en mis respuestas a este punto. ¿Se refiere al perfil de evaluaciones? O "simplemente" a la "expression" considerada en "learning-objective"? Necesitaría saber si qué expressions son posibles para valorar bien este punto...
4	3	3	2	4	3	4	
5	5	5	4	4	4	3	
6	4	5	4	4	4	4	
7	3	4	5	3	3	3	
8	3	3	3	3	3	3	
9	5	5	4	5	3	4	
10	5	5	4	5	5	5	see comment to question 3
11	4	3	3	4	3	4	
12	4	4	4	5	3	4	
13	5	5	5	5	5	5	
14	5	4	4	3	2	3	
15	5	5	3	5	4	3	

Tabla D.10: Respuestas a la pregunta 8 del cuestionario

Id	Útil	Necesario	Original	Realizable	Eficiente	Expresivo	Comentarios
1	5	3	5	4	3	4	
2	5	5	5	4	4	4	
3	4	4	4	4	4	4	Es muy útil como primer indicador de las tendencias generales sobre el efecto de la adaptación...
4	4	4	4	3	2	4	
5	4	5	2	3	2	3	
6	5	5	5	5	4	5	
7	3	3	5	3	3	4	
8	3	3	3	3	3	3	
9	5	5	4	5	3	4	
10	5	5	4	5	5	5	see comment to question 3
11	2	2	3	4	3	3	
12	4	5	4	5	3	4	
13	5	4	5	5	5	5	
14	5	4	4	2	2	3	
15	5	4	3	4	4	4	

Tabla D.11: Respuestas a la pregunta 9 del cuestionario

Id	Util	Necesario	Original	Realizable	Eficiente	Expresivo	Comentarios
1	5	4	5	4	4	4	
2	4	5	4	5	4	4	
3	4	4	4	4	4	4	Mismo comentario pero insisto en que es muy importante tener en cuenta las características de las situaciones concretas en el proceso de evaluación. Cada situación de aprendizaje es única y muchas de las adaptaciones responderán a necesidades particulares, por lo que no procederá su integración en la UoL
4	4	4	3	4	3	4	
5	4	5	4	3	3	4	
6	5	5	5	5	4	5	
7	4	3	5	3	3	3	
8	3	3	3	3	3	3	
9	5	5	4	5	3	4	
10	5	5	4	5	5	5	see comment to question 3
11	4	3	3	4	3	4	
12	4	5	4	5	3	4	
13	5	4	5	5	5	5	
14	5	4	5	3	2	3	
15	5	4	3	4	4	4	

Tabla D.12: Respuestas a la pregunta 10 del cuestionario

Id	Útil	Necesario	Práctico	Realizable	Eficiente	Comentarios
1	5	4	4	4	3	
2	5	4	5	4	4	
3	4	4	5	4	4	Ver comentario a punto anterior
4	4	4	4	4	3	
5	4	5	4	4	4	
6	5	5	5	5	5	
7	5	5	4	5	4	
8	3	3	5	3	3	
9	4	5	4	5	3	
10	5	5	5	5	5	
11	5	4	3	4	3	
12	5	5	5	5	3	
13	5	5	5	5	5	
14	5	5	5	4	4	
15	5	4	5	4	4	

Tabla D.13: Respuestas a la pregunta 11 del cuestionario

Id	Flexible	Extensible	Independiente	Intrusiva	Eficiente	Comentarios
1	4	4	4	2	4	
2	4	5	4	2	4	
3	4	5	4	2	4	Me gustaría ver ejemplos con otros EMLs...
4	4	4	2	4	3	
5	3	3	2	3	3	
6	4	5	5	3	4	
7	3	5	2	3	3	It seems that the proposed extensions fit more to the IMS Learning Design EML language than other EMLs
8	2	2	5	4	3	
9	5	5	4	3	4	
10	5	5	4	4	5	The architectural approach sounds interesting
11	3	3	3	3	3	No tengo cómo valorar la arquitectura
12	5	5	4	2	3	
13	5	5	5	1	5	
14	5	5	4	3	2	
15	5	5	4	2	3	

Tabla D.14: Respuestas a la pregunta 12 del cuestionario

Id	Útil	Necesario	Flexible	Realizable	Eficiente	Comentarios
1	5	4	5	5	4	
2	4	5	5	4	4	
3	4	3	4	4	4	
4	4	4	3	4	3	
5	5	5	4	4	4	
6	4	4	4	4	4	
7	4	4	3	3	3	
8	4	5	4	4	3	
9	4	5	4	5	4	
10	5	5	5	5	5	The idea of using an ontology to test the validation phase seems appropriate
11	4	3	2	4	3	
12	4	4	4	5	3	
13	5	5	5	5	5	
14	5	5	4	4	2	
15	5	5	4	4	2	

Tabla D.15: Respuestas a la pregunta 13 del cuestionario

Id	Deficiencias
1	ninguna
2	No he detectado deficiencias dignas de reseñar
3	Vé comentarios anteriores
4	
5	
6	
7	The provided solution seem dependent from the IMS Learning Design language, I would suggest that future research activities would focus on testing the proposed model on more event-driven EML languages than activity oriented EMLs, so as to validate the reusability and the generality of the model.
8	<p>1. Aunque las experiencias de Ingeniería del Software parece que pueden ser transportadas al campo de la educación, serán necesarias realizar más de una experiencia y particularmente útil si estas fueran realizadas por diseñadores de UOL no involucrados en el desarrollo de la metodología.</p> <p>2. Sería necesario aplicarlo a más de un ejemplo.</p> <p>3. La integración en un motor de EML existente es bastante invasiva y no está muy detallada. Aun cuando, este proceso es dependiente de la plataforma, sería interesante plantear algún tipo de arquitectura orientada a servicios donde algunos de los servicios pudieran haber sido implementado por los autores de la propuesta, dejando el resto a los integradores del motor existente con esta aproximación. Además existen propuestas como CCSI que permiten integrar motores de LD con otros servicios de aprendizaje.</p>
9	
10	I do not clearly see an evidence of lacks in this system but I would suggest to consider the “trigger” paradigm and comparing it with the “watcher” approach chosen in this work. It seems to me that there is a lot of event-driven communication going-on in this work and thus the watcher sounds but I would have argued also onto trigger mechanisms.
11	
12	
13	
14	
15	

Tabla D.16: Respuestas a la pregunta 14 del cuestionario

Id	Ventajas
1	todas
2	Es original y aporta una flexibilidad que las actuales herramientas y especificaciones no permiten.
3	Mi área de interés es el “aprendizaje colaborativo apoyado por ordenador” donde los requisitos de flexibilidad son claves. Creo que el modelo soporta algunos de estos requisitos. Habría que ver cómo se podría utilizar (o ampliar) para cuestiones relacionadas con la colaboración. ¿?Tenemos que hablar! :)
4	Cualquier extensión de IMS LD sobre temas específicos que rompa la rigidez actual, sobre todo en fases de diseño y ejecución, es recomendable
5	
6	La prueba de validez desarrollada en paralelo sobre Moodle y en forma de UoL muestra la fácil adaptación del modelo a todos los escenarios actuales basados en la plataforma Moodle, que es la utilizada en los entornos en los que trabajo
7	The continuous update of the learning process based on data from its real use is a process that reflects how human designers work. As a consequence, such a process is very useful and important, especially in the case of complex Subject Domains where designing from scratch a learning flow has many parameters that cannot be foreseen during the design phase. This means that the investigated issues are of great importance and that the provided solutions (although not fully tested) are useful.
8	1. Permite realizar un desarrollo iterativo sobre las UoL. 2. La idea de transportar los cambios dinámicos a la definición estética (en tiempo de diseño) de la UoL es extremadamente útil.
9	
10	The visitor design pattern, the adaptation poke and the validation based on an ontology seems to me very appropriate to manage such systems. Moreover the approach technically sounds proving competence both at architectural design and modeling level.
11	Es una gran ventaja poder hacer modificaciones en tiempo de ejecución del proceso de aprendizaje. En mi experiencia en el campo, que se reduce a la asignatura de EAO, aprendí que mejor preparar bien las UoL antes de publicarlas, puesto que la gestión de cambios era más bien penosa. Este modelo aporta solución
12	Actualmente utilizo Moodle, por lo que creo que el modelo y su implementación sobre Moodle podría ayudarme a mejorar la evaluación por un lado, y a mejorar la calidad de los contenidos.
13	Este es un avance bastante bueno para en una segunda fase contemplar la adaptación tanto de las actividades como de los contenidos y materiales empleados.
14	Desde el punto de vista de las áreas de aprendizaje, creo que el modelo es muy útil para flexibilizar los procesos de enseñanza, ya que permite realizar modificaciones del esquema educativo del curso en tiempo de ejecución y publicación.
15	Me ha gustado especialmente el mecanismo para extender un motor de ejecución ya existente mediante el patrón visitor. La posibilidad de introducir adaptaciones en tiempo de ejecución me parece interesante y necesaria. También me ha gustado el uso de una ontología de IMS LD para la validación de las UoL adaptadas.

Bibliografía

- [1] ADL, “Advanced Distributed Learning”, <http://www.adlnet.org/>.
- [2] ADL (2003), “Sharable Courseware Object Reference Model (SCORM) Version 1.3, Application profile working draft 1.0”, *Informe técnico*, Advanced Distributed Learning (ADL).
- [3] AICC, “Aviation Industry CBT Comittee”, <http://www.aicc.org/>.
- [4] Amorim, R.; Lama, M.; Sánchez, E.; Riera, A. y Vila, X. (2005), “An ontology to describe semantically the IMS Learning Design specification”, *Journal of Educational Technology and Society*, vol. 9(1).
- [5] Amorin, R.; Sánchez, E. y Lama, M. (2006), “The problem of LD execution and the need for an IMS LD level B ontology”, en W. Nejdl y K. Tochtermann (eds.), “EC-TEL 2006”, vol. 4227 de *Lecture Notes in Computer Science*, págs. 537–542, Springer.
- [6] ARIADNE, “Alliance of Remote Instructional Authoring and Distribution Networks for Europe”, <http://ariadne.unil.ch/>.
- [7] Beck, K.; Beedle, M.; van Bennekum, A.; Cockburn, A.; Cunningham, W.; Fowler, M.; Grenning, J.; Highsmith, J.; Hunt, A.; Hunt; Jeffries, R.; Kern, J.; Marick, B.; Martin, R.; Mellor, S.; Schwaber, K.; Sutherland, J. y Thomas, D. (2002), “Manifesto for agile software development”, <http://AgileManifesto.org>.
- [8] Berge, Z. L. y Collins, M. (1995), “Computer-mediate scholarly discussion groups”, *Computers & Education*, vol. 24(3), págs. 183–189.
- [9] Berggren, A.; Burgos, D.; Fontana, J. M.; Hinkelman, D.; Hung, V.; Anthony, H. y Tielemans, G. (2005), “Practical and pedagogical issues for teacher adoption of IMS Learning Design Standards in Moodle LMS”, *Journal of Interactive Media in Education*, vol. 02.
- [10] Biglan, A. (1995), *Changing culture practices: A contextualistic framework for intervention research*, Reno, NV: Context Press.
- [11] Brusilovsky, P. (1996), “Methods and techniques of adaptive hypermedia”, *User Modeling and User-Adapted Interaction*, vol. 6(2-3), págs. 87–129.

- [12] Brusilovsky, P. (1999), "Adaptive and intelligent technologies for web-based education", *Künstliche Intelligenz, Special issue on Intelligent Systems and Teleteaching*, vol. 4, págs. 19–25.
- [13] Brusilovsky, P. (2004), "Knowledgetree: A distributed architecture for adaptive e-learning", en "Proc. of the Thirteenth International World Wide Web Conference", New York, USA.
- [14] Caeiro-Rodriguez, A.-R. L., M. y Llamas-Nistal, M. (2005), "A perspective and pattern-based evaluation framework of emls' expressiveness for collaborative learning: application to ims ld", en "Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies, ICALT 2005", Taiwan.
- [15] Can Studios Ltd., "Canvas learning", <http://www.canvaslearning.com>.
- [16] Carr, B. y Goldstein, I. (1977), "Overlays, a theory of modelling for computer aided instruction", *Informe técnico*, MIT AI Laboratory, AI Memo 406, MIT, Cambridge, MA.
- [17] Carro, R.; Pulido, E. y Rodríguez, P. (2000), "An object-oriented approach to task tree management in the tangow system", *The Electronic Journal of the Argentine Society for Informatics and Operations Research*, vol. 3(1).
- [18] Cockburn, A. (2002), *Agile Software Development*, Boston: Addison-Websley.
- [19] COGIGRAPH Technologies - Télé-université's LICEF research center, "MOT™ / MOT Plus™ - Knowledge Modeling Editor", <http://www.licef.teluq.quebec.ca/gp/eng/productions/mot.htm>.
- [20] Collins, M. y Berge, Z. (1997), "Moderating online electronic discussion groups", American Educational Research Association. Chicago, IL.
- [21] Conlan, O. y Wade, V. (2004), "Evaluation of APeLS - an adaptive elearning service based on the multi-model, metadata-driven approach", en "Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2004) Proceedings", Eindhoven, The Netherlands.
- [22] D. Hernández-Leo et al. (2006), "Creating by reusing learning design solutions", en "Proceedings of 4º Simposio Internacional de Informática Educativa (SIEE'06)", Cáceres (España).
- [23] Dewey, J. (1953), *Essays in experimental logic*, New York: Dover.
- [24] Dey, A. K. (2001), "Understanding and using context", en "CHI 2000 Workshop on the What, Who, Where, When and How of Context-Awareness", vol. 5(1).
- [25] Dodero, J. M.; Zarraonandia, T.; Fernández, C. y Díez, D. (2007), "Generative adaptation and reuse of competence development programmes", *IMS Learning Design and Adaptation (Special Issue Editorial). Journal of Interactive Media in Education*, [To be published].

- [26] Dougiamas, M. (1998), "A journey into constructivism", <http://dougiamas.com/writing/constructivism.html>.
- [27] Duggleby, J. (2000), *How to be an Online tutor*, Morgan Kaufman Ed.
- [28] F. Buendía-García y P. Díaz Pérez (2003), "A framework for the management of digital educational contents conjugating instructional and technical issues", *Educational Technology and Society*, vol. 6(4), págs. 48–59.
- [29] Fox, E. (2006), "Constructing a pragmatic science of learning and instruction with functional contextualism", *Educational Technology, Research and Development*, vol. 54(1), págs. 5–36.
- [30] Gamma, E.; Helm, R.; Johnson, R. y Vlissides, J. (1995), *Design Patterns*, Addison Wesley, Reading, MA.
- [31] Gilbert, J. E. y Han, C. Y. (1999), "Arthur: Adapting instruction to accommodate learning style", en P. D. Bra y J. Leggett (eds.), "Proceedings of WebNet'99, World Conference of the WWW and Internet", págs. 433–438, Honolulu, HI, AACE.
- [32] Hayes, S. C. (1993), *Varieties of scientific contextualism*, cap. Analytic goal and the varieties of scientific contextualism, págs. 11–27, Reno: Context Press, S.C. Hayes, L. J. Hayes, M. Sato and K. Ono ed.
- [33] Hernández-Leo, D.; Asensio-Pérez, J. I. y Dimitriadis, Y. (2005), "Computational representation of collaborative learning flow patterns using IMS Learning Design", *Educational Technology and Society*, vol. 8(4), págs. 75–89.
- [34] Hillman, D. C.; Willis, D. J. y Gunawardena, C. N. (1994), "Learner-interface interaction in distance education: An extension of contemporary models and strategies for practitioners", *The American Journal of Distance Education*, vol. 8(2), págs. 30–42.
- [35] Holmberg, B. (1989), *Theory and practice of distance education*, London/New York: Routledge.
- [36] IEEE (2000), "Public and private information, draft 6", http://www.edutool.com/papi/drafts/06/papi_learner_06.doc.
- [37] IEEE Learning Technologies Standards Committee (2002), "Learning Objects Meta-Data Specification. Version 6.3", disponible en: <http://ltsc.ieee.org/wg12/files/>.
- [38] IMS Global Learning Consortium, "IMS Global Cartridge.", <http://www.imsglobal.org/commoncartridge.html>.
- [39] IMS Global Learning Consortium, "Instructional Management Systems", <http://imsproject.org/>.

- [40] IMS Global Learning Consortium (2001), “IMS Learner Information Package Information Model. Version 1.0 final specification”, <http://www.imsglobal.org/profiles/index.html>.
- [41] IMS Global Learning Consortium (2003), “IMS Learning Design Information Model, version 1.0 - final specification”, <http://www.imsglobal.org/learningdesign/ldv1p0/imsldinfov1p0.html>.
- [42] IMS Global Learning Consortium (2003), “IMS Simple Sequencing Information and Behavior Model. Version 1.0 Public Draft Specification”, http://www.imsglobal.org/simplesequencing/ssv1p0/imsss_infov1p0.html.
- [43] IMS Global Learning Consortium (2004), “IMS Content Packaging Information Model, Version 1.1.4 Final Specification”, http://www.imsglobal.org/content/packaging/cpv1p1p4/ims_cp_infov1p1p4.html.
- [44] IMS Global Learning Consortium (2006), “IMS Question & Test Interoperability Overview Version 2.1 Public Draft (revision 2) Specification”, .
- [45] ISO (2004), “ISO 8601: Data elements and interchange formats – Information interchange – Representation of dates and times”, <http://www.iso.org/iso/en/prods-services/popstds/datesandtime.html>.
- [46] Jacobson, I.; Griss, M. y Johnson, P. (1997), *Software Reuse. Architecture, Process and Organization for Business Success*, AddisonWesley.
- [47] Jeffery, A. y Currier, S. (2003), “What is... IMS Learning Design?”, http://www.cetis.ac.uk/lib/media/WhatIsLD_web.pdf.
- [48] Karampiperis, P. . S. D. (2005), “Towards next generation activity-based web-based educational systems”, en “Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies (ICALT 05)”, págs. 868–872.
- [49] Kiczales, G.; Lamping, J.; Menhdhekar, A.; Maeda, C.; Lopes, C.; Loingtier, J. y Irwin, J. (1997), “Aspect-Oriented Programming”, en M. Akşit y S. Matsuoka (eds.), “Proceedings European Conference on Object-Oriented Programming”, vol. 1241, págs. 220–242, Berlin, Heidelberg, and New York: Springer-Verlag.
- [50] Kiefer, M.; Lausen, G. y Wu., J. (1995), “Logical foundations of object-oriented and frame-based languages”, *Journal of ACM*, vol. 42, págs. 741–843.
- [51] Kingsland, A. (1992), “Cal need not be resource extravagant”, en “Proceedings from ITTE 1992”, Brisbane.
- [52] Koper, E. J. R. y Manderveld, J. M. (2004), “Educational modelling language: modelling reusable, interoperable, rich and personalised units of learning”, *British Journal of Educational Technology*, vol. 35(5), págs. 537–552.
- [53] Koper, R. (2001), “Modeling units of study from a pedagogical perspective. The pedagogical meta-model behind EML”, *Informe técnico*, Educational Technology Expertise Centre, Open University of the Netherlands.

- [54] Koper, R. y Olivier, B. (2004), “Representing the learning design of Units of Learning”, *Educational Technology and Society*, vol. 7(3), págs. 97–111.
- [55] Lama, M.; Sánchez, E.; Amorim, R. y Vila, X. (2005), “Semantic description of the IMS Learning Design specification”, en “AIED-Workshop on Semantic Web technologies for E-Learning (SW-EL 05)”, Amsterdam.
- [56] LAMS International, “Learning Activity Management System”, <http://www.lamsinternational.com/>.
- [57] LMML, “Learning Material Markup Language Framework”, <http://www.lmml.de>.
- [58] LTSC, “Learning Technology Standards Committee”, <http://ltsc.ieee.org/>.
- [59] M. Svahnberg et al. (2002), “A taxonomy of variability realization techniques”, *Technical paper*, Blekinge Institute of Technology. Sweden.
- [60] Macromedia, “Macromedia-authorware”, <http://www.adobe.com/products/authorware/>.
- [61] Mager, R. (1975), *Preparing Instructional Objectives*, Belmont, Lake Publishing Co., 2nd edition ed.
- [62] Martel, C.; Vignollet, L.; Ferraris, C.; David, J. y Lejeune, A. (2006), “LDL: An alternative EML”, en “Sixth IEEE International Conference on Advanced Learning Technologies (ICALT’06)”, págs. 1107–1108.
- [63] Martel, C.; Vignollet, L.; Ferraris, C.; David, J. y Lejeune, A. (2006), “Modeling collaborative learning activities on e-learning platforms”, en “Sixth IEEE International Conference on Advanced Learning Technologies (ICALT’06)”, págs. 707–709.
- [64] Martens, H.; Vogten, H.; Rosmalen, P. v. y Koper, E. J. R. (2004), “Coppercore”, from SourceForge: <http://coppercore.org>.
- [65] McAndrew, P. y Weller, M. (2005), *Learning Design*, cap. Chapter 17. Applying Learning Design to Supported Open Learning, págs. 281–290, Berlin: Springer-Verlag.
- [66] Microsoft, “Learning Resource iNterchange”, <http://http://www.microsoft.com/learning/elearning.mspx>.
- [67] Miller, G. G. (2001), “The characteristics of agile software processes”, en “The 39th International Conference of Object-Oriented Languages and Systems (TOOLS 39)”, Santa Barbara, CA.
- [68] Mizoguchi, R. y Bourdeau, J. (2000), “Using ontological engineering to overcome ai-ed problems”, *International Journal of Artificial Intelligence in Education*, vol. 11(2), págs. 107–121.

- [69] MOODLE Org, “MOODLE”, <http://moodle.org/>.
- [70] Moore, M. (1989), “Three types of interaction”, *American Journal of Distance Education*, vol. 3(2), págs. 1–6.
- [71] Oreizy, P.; Medvidovic, N. y Taylor, R. (1998), “Architecture-based runtime software evolution”, en “Proceedings of the 20th International Conference on Software Engineering”, págs. 177–186, Kyoto, Japan.
- [72] Papanikolaou, K. A.; Grigoriadou, M.; Kornilakis, H. y Magoulas, G. D. (2003), “Personalising the interaction in a web-based educational hypermedia system: the case of INSPIRE”, *User-Modeling and User-Adapted Interaction*, vol. 13(3), págs. 213–267.
- [73] Paquette, G. (2004), *Instructional Engineering in Networked Environments*, Pfeiffer-Wiley.
- [74] Rawlings, A.; Van Rosmalen, P.; Koper, R.; Rodriguez-Artacho, M. y Lefreire, P. (2002), “CEN/ISSS WS/LT Learning Technologies Workshop - Survey of Educational Modelling Languages (EMLs)”, <http://eml.ou.nl/forum/docs/EMLSurveyversion1.pdf>.
- [75] Reiser, R. (2002), *Trends and issues in instructional design and tecnology*, cap. A history of instructional design, págs. 26–53, Merrill Prentice Hall, R.A. Reiser & J.V. Dempsey ed.
- [76] RELOAD Project, “Reusable eLearning Object Authoring & Deliberly”, <http://www.reload.ac.uk/>.
- [77] Roberts, T.; Jones, D. y Romm, C. T., “Four models of online education”, en “Proceedings of TEND 2000”, Abu Dhabi, UAE.
- [78] Rodríguez, M. (2000), *Una arquitectura cognitiva para el diseño de entornos telemáticos de enseñanza y aprendizaje*, Tesis doctoral, Universidad Nacional de Educación a Distancia.
- [79] Sierra, R. (1986), *Tesis doctorales y trabajos de investigación científica*, Thomson editores Spain Paraninfo, S.A.
- [80] Specht, M. y Oppermann, R. (1998), “ACE - Adaptive Courseware Environment”, *The New Review of Hypermedia and Multimedia*, vol. 4(1), pág. 141–161.
- [81] Swanson, E. B. (1976), “The dimensions of maintenance”, en “Proceedings of the 2nd International Conference on Software Engineering (ICSE)”, pág. 492–497.
- [82] Towle, B. y Halm, M. (2005), “Designing adaptive learning environments with learning design”, en C. Tattersall y R. Koper (eds.), “Learning Design: A Handbook on Modelling and Delivering Networked Education and Training”, cap. 12, págs. 215–226, Springer-Verlag.

- [83] Van Rosmalen, P. y Boticario, J. (2005), “Using learning design to support design and runtime adaptation”, en C. Tattersall y R. Koper (eds.), “Learning Design: A Handbook on Modelling and Delivering Networked Education and Training”, cap. 12, págs. 291–301, Springer-Verlag.
- [84] Vassileva, J. (1997), “Dynamic course generation”, *Communication and Information Technologies*, vol. 5(2), págs. 87–102.
- [85] Vygostky, L. (1979), *El desarrollo de los procesos psicológicos superiores*, Barcelona: Crítica.
- [86] Weller, M., “The SLeD project: Investigating learning design and services, JISC e-learning focus”, <http://www.elearning.ac.uk/features/sledproject>.
- [87] Wenger, E. (1987), *Artificial Intelligence and Tutoring Systems: Computational and Cognitive approaches to the communication of knowledge*, Morgan Kaufman Ed.
- [88] World Wide Web Consortium (W3C) (2007), “XML Path Language (XPath) 2.0”, *Informe técnico*, World Wide Web Consortium (W3C).
- [89] Yang, G.; Kiefer, M.; Zhao, C. y Chowdhary, V., “Flora-2: Users’ manual (version 0.94)”, <http://flora.sourceforge.net/docs/floraManual.pdf>.
- [90] Yaverbaum, G. y Reisman, S. (1995), “Multimedia paradox: Where do we go from here?”, en “Proceeding from edMEDIA’ 95”, Graz, Austria.
- [91] Zarraonandia, T.; Doderó, J. M. y Fernández, C. (2005), “Adaptaciones al contexto en tiempo de ejecución de unidades de aprendizaje”, en “Simposio Nacional de Tecnologías de la Información y Comunicaciones de la Educación (SINTICE-05)”, Granada, Spain.
- [92] Zarraonandia, T.; Doderó, J. M. y Fernández, C. (2005), “Crosscutting runtime adaptations of LD execution”, en “UNFOLD-PROLEARN Joint Workshop”, Valkenburg, The Netherlands.
- [93] Zarraonandia, T.; Doderó, J. M. y Fernández, C. (2006), “Crosscutting runtime adaptations of LD execution”, *Journal of Educational Technology and Society*, vol. 9.
- [94] Zarraonandia, T.; Doderó, J. M.; Fernández, C.; Aedo, I. y Díaz, P. (2007), *Computers and Education: E-learning - from theory to practice*, cap. 15. Iterative Design of Learning Processes, Springer.
- [95] Zarraonandia, T.; Fernández, C. y Doderó, J. M. (2006), “A late modelling approach for the definition of computer-supported learning process”, en “ADALE Workshop on Adaptive Learning and Learning Design”, Dublin, Ireland.
- [96] Zeiliger, R. (1993), “Adaptive testing: contribution of the SHIVA model”, en D. Leclercq y J. Bruno (eds.), “Item banking: Interactive testing and self-assessment”, vol. 112 de *F*, págs. 54–65, Springer-Verlag, Berlin.