

# Modelado de “buenas prácticas” docentes en experiencias e-learning

José V. Ballester <sup>1</sup>, J. Ignacio Panach <sup>1</sup>, Félix Buendía <sup>2</sup>, Óscar Pastor <sup>1</sup>

<sup>1</sup> Dpto. de Sistemas Informáticos y Computación, Universitat Politècnica de València,  
Camino Vera s/n 46002 Valencia – SPAIN

{jvballester, jpanach, opastor}@dsic.upv.es

<sup>2</sup> Dpto. de Informática de Sistemas de Computadores, Universitat Politècnica de València,  
Camino Vera s/n 46022 Valencia – SPAIN

fbuendia@disca.upv.es

**Abstract.** Los intentos de sistematizar los procesos formativos en cualquier ámbito educativo se han convertido en una especie de “santo grial”. Para representar las “buenas prácticas” educativas se ha utilizado el concepto de patrón de diseño que ha tenido una amplia difusión en diversos ámbitos como elemento de representación de soluciones a problemas recurrentes. En nuestro caso, se trata de modelar los patrones e-learning con el fin de extraer una información que pueda ser útil tanto para los docentes que puedan aplicar dichos patrones como para los desarrolladores de aplicaciones basadas en este tipo de tecnologías. En este trabajo se propone modelar los patrones e-learning con una adaptación de un método MDA (Model Driven Architecture) denominado OO-Method. OO-Method es un método orientado a objetos que facilita la generación de un sistema de información a partir de modelos conceptuales.

Keywords: e-learning, MDA, OO-Method, patrones

## 1 Introducción

La mayor parte de los profesores universitarios son expertos en los conocimientos técnicos y las aplicaciones prácticas en su ámbito de enseñanza. Así los profesores de Programación en la carrera de Ingeniería Informática son expertos en los lenguajes de programación de ordenadores y contenidos técnicos de la materia que imparten, pero suelen carecer de conocimientos respecto a las metodologías pedagógicas y práctica docente.

Muchos docentes intentan mejorar sus competencias “imitando” a aquellos profesores de los que más han aprendido durante su etapa de estudiantes y aplicando lo que consideran que son las “mejores prácticas” en la docencia. Pero en la mayoría de los casos esta hipótesis es errónea, porque lo que les ha servido a ellos no es generalizable.

La forma de facilitar el aprendizaje de las cualidades y competencias que describen a un buen profesor es sistematizar este tipo de experiencias por parte de los expertos en la materia. El problema se ha visto agravado con la aparición de las tecnologías e-learning, dado que se ha incrementado notablemente el número de aplicaciones educativas y ha dificultado aún más la necesaria sistematización.

Hace más de una década que muchos investigadores han centrado sus esfuerzos en el campo de investigación del “e-learning”, y en particular desde iniciativas a nivel europeo [7] que han hecho hincapié en su importancia estratégica en diversos contextos formativos. Los autores no terminan de establecer una única definición para el término e-learning, la mayor parte hacen referencia a cuestiones específicas como la enseñanza “online”, a distancia a través de Internet o mediante el uso de plataformas o campus virtuales. Sin embargo, se ha extendido una concepción más amplia del e-learning que cubre todos aquellos procesos o actividades de aprendizaje apoyadas en el uso de las denominadas TIC (Tecnologías de la Información y la Comunicación) tal como indican Horton [9] o el área de formación e-learning del gobierno del Reino Unido [5].

Como ya se ha indicado, una de las técnicas más intuitiva y rápida para mejorar nuestra experiencia y conocimientos es copiar las “mejores prácticas”. Adaptar las soluciones utilizadas por los expertos en la resolución de problemas y aplicar su conocimiento a las nuevas situaciones que se nos presenten. Este es el concepto que emana del término *patrón*, por tanto, la técnica de establecer patrones surge de la necesidad de capturar y poder compartir el conocimiento.

Christopher Alexander [1] fue el primero en formalizar el concepto de “*patrón de diseño*” en el campo de la arquitectura civil y estableció una manera de documentar la esencia de una solución a un problema recurrente en el diseño arquitectónico.

Según su definición, un patrón “*describe un problema que ocurre una y otra vez en nuestro entorno y, a continuación, se detalla el núcleo de la solución a ese problema, de tal forma que se puede utilizar esta solución más de un millón de veces, sin tener que hacerlo del mismo modo dos veces*” [1]. Por lo general, los patrones no se utilizan de forma aislada sino en combinación con otros patrones, con el fin de resolver problemas complejos partiendo de la solución de problemas más simples. Actualmente la técnica de patrones se emplea en muchas áreas de conocimiento, y tanto los de ámbitos pedagógicos [2], [17], [21] como en entornos e-learning [19], [20] son un buen ejemplo.

Con la llegada de las nuevas tecnologías e-learning, el cambio de escenario hace que sea obligatoria una adaptación de los modelos formativos. El enfoque habitual de los docentes se encuentra centrado en la enseñanza presencial y orientan todas sus habilidades docentes en el entorno de un aula, mientras que en e-learning desaparece el concepto físico de aula.

Los elementos a tener en consideración en e-learning [4] son:

- Retos pedagógicos: seguir modelos y patrones pedagógicos definidos y que no sea un mero contenedor de información
- Interfaz adecuada y menús intuitivos
- No se “enseña” igual que en la docencia presencial

En este contexto se han realizado múltiples investigaciones para identificar y determinar los patrones e-learning. Este nuevo conjunto de patrones e-learning no es un subconjunto de los patrones formativos. Aparecen nuevos problemas que no encuentran correspondencia con los problemas formativos recogidos en los patrones, otros no tienen aplicación en un entorno virtual y hay algunos que requieren una nueva solución al problema identificado. Por tanto, en este caso es necesario redefinir el patrón formativo para crear uno nuevo.

En la literatura existe gran cantidad de trabajos como los de [6], [10], [19] y [20] que proponen los patrones e-learning como un método adecuado de transmitir las buenas prácticas educativas a los docentes.

Gran parte de los proyectos de investigación orientados a modelar los patrones e-learning, proyecto como E-LEN [19], E-DILEMA [18] y PCeL (Person-Centered e-Learning) [20]), han formalizado los patrones definiendo un conjunto de campos o datos referentes al patrón, y el contenido de dichos campos lo han expresado de forma textual.

Es sabido que una descripción textual presenta muchos inconvenientes en el momento de ser interpretada por un docente para ser aplicada, puede resultar ambigua, genérica, plantea problemas idiomáticos, imprecisa, etc.

Por otra parte, con la publicación del libro *Design Patterns: Elements of Reusable Object-Oriented Software* [10], la metodología de patrones se implantó con mucho éxito en la ingeniería del software, permitiendo convertir los patrones de diseño en el pilar central para la reutilización de software. Este concepto es clave en los métodos MDA (Model Driven Architecture).

Para modelar dichos patrones software se han definido herramientas y métodos específicos. Estos modelos aportan la concreción y exactitud necesaria para que el software creado resuelva perfectamente las necesidades planteadas. Se han definido incluso modelos de transformación que permiten generar de forma automática el código del programa a partir de los modelos conceptuales [16]. Y, al mismo tiempo, estos patrones software permiten una abstracción conceptual que los hacen fácilmente comprensibles para las personas no expertas en ingeniería software ya que utilizan sencillos diagramas gráficos para representar el modelo conceptual.

Por tanto, los patrones utilizados en la ingeniería software mantienen las ventajas de facilidad de comprensión de la descripción textual, pero eliminan las ambigüedades e inconvenientes del texto, porque requieren un mayor nivel de detalle y concreción.

La contribución de este artículo es utilizar los métodos de MDA para modelar los patrones instructivos, en concreto los patrones de e-learning. Con ello se pretende formalizar la descripción de los patrones, que muchas veces es ambigua y por tanto, difícilmente aplicable a entornos e-learning específicos. Además, incluyendo los patrones en un entorno MDA se podrían generar las aplicaciones e-learning mediante transformaciones de modelos a código, tal como el paradigma MDA propone. Para lograr introducir los patrones en un entorno MDA será necesario un primer análisis de los patrones e-learning para poder conocer el suficiente detalle para trasladarlos al modelo software. Este proceso permitirá identificar más concretamente las

posibilidades de uso de cada patrón, lo que redundará en beneficio de los docentes que quieran utilizarlos, ya que establecerá un vocabulario único,

En resumen, la modelización de patrones permitirá sistematizar el proceso formativo en un contexto e-learning con el fin de extraer una información que pueda ser útil tanto para los docentes que puedan aplicar dichos patrones como para los desarrolladores de aplicaciones basadas en este tipo de tecnologías.

En el resto de apartados se revisa la propuesta de MDA aplicada a patrones e-learning y se analiza su utilización en ciertos casos. En el apartado 2 se realiza una revisión del estado de la investigación alrededor de los patrones instructivos y de e-learning, evidenciando las carencias de estos métodos. En el apartado 3 se propone la aplicación de un método MDA al modelado de los patrones e-learning como solución al problema, en concreto, se utilizará el OO-Method. La propuesta se aplica a un caso práctico en el apartado 4, al patrón de DEBATE de E-LEN. El último apartado lo dedicaremos a las conclusiones.

## **2 Estado del arte**

El modelado de procesos formativos ha recibido una amplia atención en los últimos tiempos y se han propuesto múltiples opciones para su representación, desde las más sencillas basadas en mapas conceptuales [12], pasando por sistemas hipermedia [3] hasta los llamados Lenguajes de Modelado Educativo [11]. La mayoría de estas propuestas plantean la especificación de las características de un proceso formativo, pero sin entrar a valorar su idoneidad. En nuestro caso se ha optado por el uso de patrones de diseño como elemento para modelar aquellas prácticas docentes que han tenido una amplia aceptación, y en particular, aquellos patrones relacionados con la representación de experiencias e-learning.

Por un lado, los patrones pedagógicos describen problemas que se presentan con frecuencia en el proceso de enseñanza-aprendizaje. Se tratan los problemas asociados a la motivación de los alumnos, selección de contenidos, selección de materiales, secuencia de contenidos, selección de actividades, evaluación, procedimientos de evaluación, criterios de calificación, etc. [17]

Entre los catálogos de patrones pedagógicos más referenciados se encuentran [22]:

- Educación en las aulas
- Grupos de estudio
- Seminarios
- Tiempos de un curso
- Pautas pedagógicas

En este artículo nos vamos a centrar en los patrones relacionados con la enseñanza e-learning, que incluyen patrones pedagógicos y los patrones asociados al modo en que se imparte la docencia y al entorno virtual donde se realiza.

Los Entornos Virtuales de Aprendizaje (EVA) utilizados en e-learning presentan cuestiones pedagógicas específicas y, a veces, las mismas pero con distinta solución a las planteadas en el entorno de enseñanza presencial. Por tanto, requerirán de la definición de nuevos patrones.

Existen numerosas iniciativas de aplicación de los catálogos de patrones para la mejora de los procesos de implementación de alternativas de aprendizaje virtual: en las áreas de producción de objetos de aprendizaje (GLO Maker, WebQuest), para mejorar los procesos organizativos en instituciones educativas (estudio en la Central Queensland University, Australia), catalogación de patrones en e-learning (proyecto E-LEN, E-DILEMA y proyecto PCeL), etc.

Los proyectos mencionados en el área de e-learning tienen como objetivo crear un catálogo de buenas prácticas docentes en este entorno, pero la diferencia estriba en el foco del problema que hace cada uno de ellos:

- E-LEN: tiene como objetivo determinar la infraestructura y organización necesarias para la “*creación de centros de e-learning*”, para su posterior difusión.
- E-DILEMA: El proyecto tenía por objeto proporcionar soporte a profesores y estudiantes del ámbito universitario en el desarrollo de sus *actividades académicas* y se encontraba enfocado principalmente a la modalidad de aprendizaje mixto o blended-learning (b-learning), en el que las actividades relacionadas con el desarrollo de cursos on-line son complementadas con clases presenciales. Las actividades del proyecto se orientaban a asegurar el éxito de las experiencias de aprendizaje a distancia, por lo que buena parte de ellas se centraron en la reunión y posterior difusión de buenas prácticas relacionadas con el uso, la planificación y la implementación de sistemas de soporte a dicha modalidad de aprendizaje.
- PCeL: Básicamente es un enfoque al b-learning desde los principios de la *educación humanista*. Se ha creado un repositorio de patrones PCeL que describe las prácticas comunes de una manera uniforme, estructurada, y, en parte, en formato visual.

Estas propuestas basan su solución de modelar los patrones utilizando una descripción textual de los diferentes datos que componen la estructura del patrón. Como son, por ejemplo, el código de identificación, nombre corto del patrón, categoría, descripción del problema que resuelve, entorno(s) donde se aplica, ejemplos de utilización, requisitos, recomendaciones, patrones relacionados, etc.

Alexande	Hillside	Bergin	4 Groupware (GoF)	ELEN
Imagen mostrando un ejemplo	Nombre	Nombre	Nombre	Nombre
Contexto	Problema	Imagen	Autores	Categoría
Resumen del Problema	Contexto	Contexto	Iteración N°	Abstract
Problema	Fuerzas	Fuerzas	Familia	Problema
Solución	(componentes del problema)	Solución	Problema	Análisis
Diagrama de la solución	Solución	Discusión/ consecuencias/ Implementación	Escenario	Soluciones conocidas
Patrones relacionados	Contexto resultante	Recursos especiales	Contexto	Preguntas de investigación
	Racionalización (reflexión del patrón)	Patrones relacionados	Indicaciones	Contexto
		Ejemplos	Soluciones	Condiciones
		Contraindicaciones	Participantes	Discusión/consecuencias
		Referencias	Racionalización	Referencias
			Reglas de seguridad	Patrones relacionados
			Usos conocidos	Autores
			Patrones relacionados	Fecha
			Referencias	Créditos
			Citas	

Tabla 1. Estructuras de descripción de patrones

Todas ellas pretenden crear patrones comunes de ámbito universal, pero existe un problema importante que impide su universalización, no se ponen de acuerdo respecto a una estructura común de cómo expresar cada uno de los patrones identificados y los inconvenientes para tratar la información textual descriptiva, que es desestructurada.

En la Tabla 1 se muestra un resumen de los diferentes datos que incorporan los patrones de cinco de los más conocidos catálogos [4].

En la Figura 1 se muestra un ejemplo concreto de uno de estos catálogos, el de E-LEN [19], donde se indica el contenido textual de los diferentes campos que componen la estructura definida para cada patrón. Este patrón es el que utilizaremos en el apartado 4 para aplicar nuestra propuesta.

Name: DEBATE

The exploration of contradictory views can promote a deeper understanding of a subject. It can stimulate each participant to develop their own opinions and explore their reasons for them. However, students are sometimes reluctant to challenge each others' views.<sup>17</sup>

When a group of students engages in a discussion it is common for individuals to hold contradictory views. A constructive exploration of these differences can assist learning by causing the students to examine their beliefs and opinions and assess them in the light of others' views. In some cases, students come to a consensus. In others, students continue to hold different beliefs but have a greater understanding of both their own, and opposing points of view.

Sometimes students do not spontaneously generate any differences of opinion about the subject matter, but different opinions exist in the field under discussion. In this case it can be useful for the teacher to cause the students to explore the contradictory views by engaging in a debate.

Providing a structure for debate can enable students to explore different opinions in a fair and equal manner that is not confrontational or challenging to individual students' sense of self-esteem.

Therefore

Identify two or more students to 'speak' to each side of a motion. Give them equal time/space to make a contribution setting out their views. Once they have made their contributions, open the discussion to everyone in the discussion group. At the end of the period defined as the lifetime for the debate, invite the proposing and opposing speakers to sum up. End the debate by taking a vote on the motion.

**Related patterns:**  
DISCUSSION ROLES, VOTES...

**Figura 1.** Patrón DEBATE definido por E-LEN

A la vista de este ejemplo, es evidente que la descripción *textual* con la que se modela el patrón presenta muchos inconvenientes en el momento de ser interpretada por un docente para su aplicación práctica, ya que resulta ambigua, genérica e imprecisa. Como posible solución, proponemos en este artículo sistematizar los patrones instructivos en un contexto e-learning utilizando MDA.

### 3 Propuesta

Con el objetivo de modelar un patrón instructivo en experiencias e-learning que sea intuitivo y fácilmente comprensible para los docentes vamos a proponer un método MDA (Model Driven Architecture). Es decir, vamos a modelar patrones e-learning con el método usado para los patrones software.

MDA es un estándar propuesto por la Object Management Group (OMG) [13]. En este estándar se definen una serie de modelos que representan el sistema software a distintos niveles de abstracción, y que van desde lo más abstracto posible (la representación del sistema sin tener en cuenta su computación) hasta el código que implementa el sistema.

MDA confiere a los modelos la mayor importancia en el proceso de desarrollo de aplicaciones, en detrimento del código. El modelo se convierte en el elemento más valioso (hasta el punto de afirmar que, en MDA, todo es un modelo) puesto que a partir de él, mediante una serie de transformaciones se puede obtener el código de la aplicación.

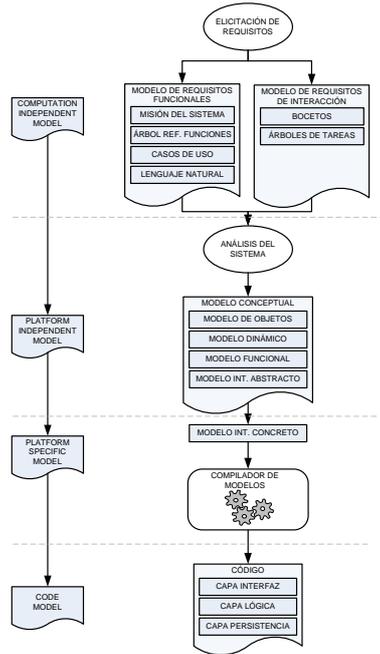
Por tanto, la principal ventaja de MDA es que el analista es más eficiente en el desarrollo de sistemas, ya que sólo debe modelar el sistema y es el proceso de transformación el encargado de su implementación. El grado de automatización del proceso de transformación de modelos va desde el cien por cien de automatización, hasta que el analista tenga que aplicar las transformaciones de manera manual. Cuanto más se automatice el proceso, menos trabajo debe realizar el analista. Las *Tecnologías de Transformación de Modelos (TTM)* surgen dentro del ámbito MDA proponiendo el desarrollo de sistemas completamente funcionales a partir de transformaciones automáticas entre modelos de distintos niveles de abstracción. Los métodos MDA abogan por construir sistemas software a partir de modelos conceptuales mediante transformaciones, pero no especifican cómo llevar a cabo dichas transformaciones.

Como ejemplo de TTM, este artículo propone OO-Method[16], es un método de desarrollo software que soporta el estándar MDA, ya que diferencia los distintos artefactos software del sistema en los diferentes niveles de abstracción definidos en la propuesta MDA. Por un lado representa el espacio del problema en base a modelos conceptuales, y por otro representa el espacio de la solución en base al código correspondiente en un lenguaje de programación específico. La elección de OO-Method con respecto al resto de TTMs ha estado motivada por sus principales ventajas:

1. Genera aplicaciones totalmente funcionales a partir de modelos conceptuales sin que el analista tenga que escribir una sola línea de código.
2. Sus modelos conceptuales son lo suficientemente abstractos como para representar todos los aspectos de un sistema.
3. OO-Method tiene una implementación industrial en la herramienta OlivaNOVA [14], por lo que los patrones modelados podrán ser fácilmente implementados con esta herramienta, que es capaz de representar de forma gráfica los modelos conceptuales de OO-Method y generar el código que implementa lo expresado en

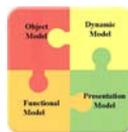
dichos modelos. Este proceso de transformación se lleva a cabo mediante su compilador de modelos.

La Figura 2 [15] hace una comparativa entre los niveles de abstracción de MDA y los modelos de OO-Method. Esta comparativa muestra dónde va ubicado cada modelo de OO-Method dentro del proceso MDA.



**Figura 2.** OO-Method

En la Figura 3 vemos con mayor detalle los elementos que conforman el modelo conceptual:



**Figura 3.** Modelo conceptual

Por tanto, para describir el modelo conceptual usamos cuatro submodelos en el que cada uno de ellos se centra en un aspecto o visión del problema:

- Modelo de Objetos: Este modelo permite especificar la estructura de clases identificadas en el dominio del problema, así como las relaciones estructurales y las relaciones de agentes sobre los servicios de las clases. El Modelo de Objetos se representa mediante un *Diagrama de Clases* (DC) en el que se indican los atributos y los servicios para cada una de las clases.

- Modelo Dinámico: En este modelo se representan los aspectos del sistema referentes al control, a las posibles secuencias de eventos que pueden ocurrir en la vida de los objetos, y a la interacción entre éstos. Para describir la secuencia posible de eventos se utilizan *Diagramas de Transición de Estados* (DTE) para cada clase, y para describir la comunicación/interacción entre objetos se utiliza un *Diagrama de Interacción*.
- Modelo Funcional: Este modelo se utiliza para especificar el efecto que tienen los eventos sobre el estado de los objetos. La funcionalidad del sistema se representa dotando a este modelo de un esquema declarativo para recoger toda la información que describe los efectos de los eventos sobre los objetos del sistema.
- Modelo de Presentación: En este modelo se especifica la interfaz de usuario y está formado por dos vistas: el *Modelo de Interacción Abstracto* que representa qué elementos se visualizarán en la interfaz, y el *Modelo de Interacción Concreto*, que representa cómo se visualizarán estos elementos en la interfaz. El primer modelo a explicar es el Modelo de Interacción Abstracto. Este modelo se construye mediante la especificación de una serie de patrones de interfaz divididos en tres niveles [16].

Como trabajo preliminar de nuestra propuesta proponemos utilizar el Modelo de Objetos (representaremos el DC) y el Modelo Dinámico (únicamente el DTE) para modelar los patrones e-learning. En trabajos futuros se ampliará la definición al resto de modelos.

Para evidenciar las características de estos modelos y ver la viabilidad de nuestra propuesta, en el siguiente apartado se muestra la aplicación de estos modelos de OO-METHOD a un caso de estudio, el patrón *debate* propuesto por la iniciativa E-LEN que hay en la Figura 3.

#### 4 Caso de Estudio

A la vista del contenido del DEBATE de E-Len mostrado en la Figura 1, es evidente que la descripción *textual* que modela el patrón presenta muchos inconvenientes en el momento de ser interpretada por un docente para su aplicación práctica, porque resulta ambigua, genérica e imprecisa.

En este apartado vamos a construir el Modelo de Objetos y Modelo Dinámico del patrón DEBATE de E-LEN. Los DTE se especifican a nivel de cada clase identificada en el DC.

Ya hemos indicado que los patrones utilizados en la ingeniería software mantienen las ventajas de facilidad de comprensión de la descripción textual, pero eliminan las ambigüedades e inconvenientes del texto, porque requieren un mayor nivel de detalle y concreción. Por este motivo es necesario refinar el modelo DEBATE de la Figura 1, hay que establecer las distintas posibilidades y alternativas de solución que están respaldadas por el patrón.

Entorno del problema:
<i>“The exploration of contradictory views can promote a deeper understanding of a subject. It can stimulate each participant to develop their own opinions and explore their reasons for them. However, students are sometimes reluctant to challenge each others’ views”</i>
Entorno de la solución:
<i>“Identify two or more students to ‘speak’ to each side of a motion. Give them equal time/space to make a contribution setting out their views. Once they have made their contributions, open the discussion to everyone in the discussion group. At the end of the period defined as the lifetime for the debate, invite the proposing and opposing speakers to sum up. End the debate by taking a vote on the motion”</i>
Patrones relacionados:
<i>DISCUSSION ROLES, VOTES...</i>

**Tabla 2.** Elementos fundamentales del DEBATE definido por E-LEN

Analizando la solución propuesta por E-LEN deducimos las siguientes características de la funcionalidad de la solución:

El *Debate* está compuesto por *Usuarios* que proponen *Temas* y los contrastan con sus opiniones en el entorno de un *Foro*. Hay 2 perfiles de *Usuarios* que realizan distintas funciones dentro del Debate, los *Profesores* y los *Estudiantes*. En los *Usuarios* incluiremos la información personal común. En los *Temas* será necesario incluir información respecto al Objetivo docente que se persigue (Pensamiento o Trabajo en Equipo), el tipo de Participación (Individual o Grupal). Si el trabajo se realiza en grupos, detallaremos los integrantes en *Participantes*. El *Foro* será moderado por los *Profesores*, que lo “Crearán” y establecerán sus características: título, visibilidad (es un foro público o privado); fijaran el período de actividad del foro (fecha vigencia) y el período de inscripción para poder participar; el formato de las intervenciones (orden predefinido de las intervenciones, libre, control de tiempos de respuesta, etc.); el tipo de Evaluación: que se realizará (Votación, Evaluación Profesor, No evaluable, etc.). Los *Participantes* se inscribirán en el *Foro* para poder participar. Al final del período de inscripción, el *Profesor* “Abrirá” el *Foro*. A partir de este momento, los alumnos podrán plantear sus *Opiniones* sobre el *Tema* según el formato de intervención establecido en el *Foro*. Transcurrido el plazo previsto, el profesor procederá a “Iniciar las evaluaciones”, por tanto, ya no se podrán emitir nuevas *Opiniones*.

En caso de que se establezca una evaluación por Votación, los *Participantes* podrán incluir sus *Evaluaciones* de cada una de *Opiniones* emitidas. Pero si se había establecido que era una evaluación por parte del *Profesor*, será él el que emita las *Evaluaciones*. En cualquiera de estos casos, el profesor terminará la evaluación pasando a posibilitar la “Inclusión de Conclusiones”, y también se realizará esta acción en caso de que no sea evaluable. A partir de este momento los *Participantes* aportan sus *Conclusiones* finales a la vista del resto de informaciones. Superado el período de vigencia, el *Foro* se dará por “Cerrado” y no podrá ser modificado, únicamente consultado. Se pueden asociar *Recursos* al *Tema*, a cada *Opinión* aportada, a las *Evaluaciones* realizadas o a las *Conclusiones* obtenidas. Estos

*Recursos* tendrán un objetivo (Documentación adicional, Presentación de datos, etc.); serán de un tipo (Docente o Investigación); copyright (tipo de derechos de uso); visibilidad (público o restringido a participantes); vigencia (para el caso de tareas programadas). Cada uno de los *Recursos* podrá ser de un tipo diferente, *Video*, *Documento*, *URL* o *Audio*. Si a lo indicado añadimos las restricciones de cardinalidad, es decir, cada elemento con cuantos (0,1 o muchos) se puede relacionar tendremos una descripción más exacta de la solución planteada por el patrón.

A continuación vamos a realizar una breve descripción de cómo modelar todas las características descritas mediante OO-Method:

- Los términos marcados en *cursiva* durante la descripción de la funcionalidad del sistema anterior constituyen las *Clases* del sistema (cajas).
- Si tenemos diferentes Tipos de una Clase (tipos de usuario) se indica con la flecha (↑) que representa la Herencia (“es un”).
- Para reflejar que un objeto esta formado por varias partes utilizamos la Agregación (“parte de”), representado por un rombo negro (◆).
- y unimos las Clases para reflejar las Relaciones que están implícitas en el contexto descrito (Líneas).
- Para terminar se etiqueta la Cardinalidad (*número máximo y mínimo de elementos de la clase que pueden relacionarse con la otra*) en ambos sentidos de la relación (*min..max*).
- A cada *Clase* le asociamos la información que tiene (Atributos) y las operaciones o acciones que cambian su estado (Eventos), que en el caso de que se establezca una relación entre elementos se denomina Shared Events.

Al aplicar todas estas reglas obtendremos el diagrama de la Figura 4 que constituye el DC.

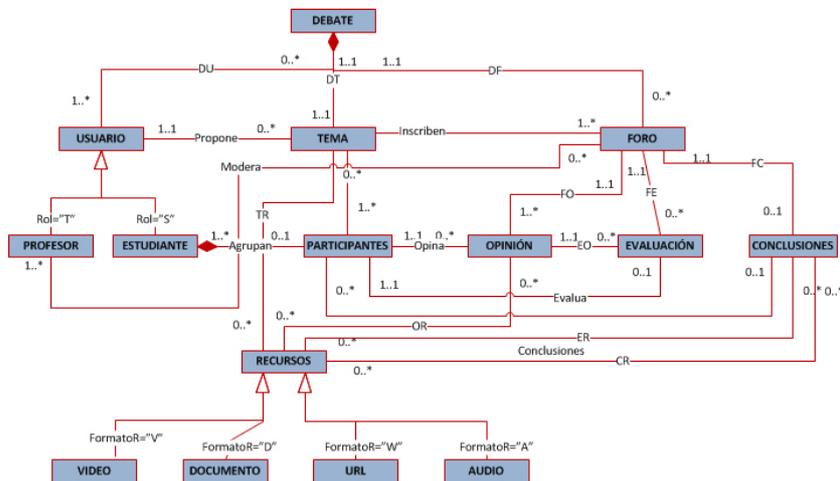
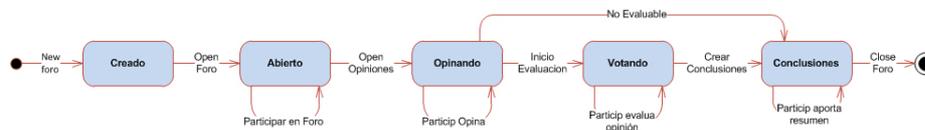


Figura 4. Modelo de Objetos: Diagrama de Clases.

Lógicamente se han omitido algunos detalles más complejos con el objetivo que el ejemplo resulte más sencillo de comprender y poder así ver la transformación. También se han ocultado los atributos y servicios de cada clase para facilitar su comprensión. El DC completo se incluye en el Apéndice 1.

Es evidente, que este modelo conceptual (DC) es mucho más preciso y aplicable a entornos de desarrollo que el planteado por E-LEN (textual).

El modelo Dinámico, representado por el DTE, en este caso en la Figura 5 se muestra el correspondiente a la clase *Foro*. Cada *Caja* representa los diferentes estados por los que pasa la Clase al ocurrir los Eventos asociados, que se representan por *Flechas*. En nuestro ejemplo, los eventos los hemos “entrecomillado” en la descripción anterior (Crearán, Abrirá, etc).



**Figura 5.** Modelo Dinámico: Diagrama de Transición de Estados de FORO

Al igual que el DC, el DTE también es un modelo muy intuitivo y sencillo, pero que cuenta con un gran poder de expresividad. Por lo visto en este caso de estudio, pensamos que el método OO-Method aplicado al modelado de los patrones e-learning puede ser de gran ayuda a la hora de desarrollar aplicaciones de e-learning. Si el DC es creado y completado por los expertos en la materia pedagógica pueden determinar todas las posibilidades a tener en consideración.

## 5 Conclusiones

En esta propuesta se ha presentado la aplicación de un método MDA, OO-Method, para modelar “buenas prácticas” docentes en entornos e-learning. Se ha presentado el método propuesto y su aplicación a un caso de estudio, el patrón DEBATE de E-LEN.

Hemos planteado un mismo caso de estudio desde dos perspectivas de modelado, el actual modelado mediante patrones textuales, hemos identificado sus debilidades y hemos realizado una propuesta para subsanarlas utilizando el modelo conceptual de OO-Method.

Los resultados de este trabajo son preliminares en base a la aplicación teórica a un reducido número de patrones e-learning, pero ya aportan una idea aproximada a la hora de sistematizar los procesos formativos mediante el uso de patrones e-learning modelados mediante un método MDA, en concreto OO-Method. No obstante, serán necesarios nuevos trabajos y experiencias para validar la aplicación presentada en este trabajo. Por último, también se plantea la posibilidad de crear Guías de aplicación de los patrones que ayuden de modo asistido a los docentes que quieran aplicarlos. Estas guías se podrían presentar en modo Wizard, que al finalizar aportase al docente las técnicas y metodologías docentes que debe aplicar

## Agradecimientos

Este trabajo ha sido realizado con el soporte del Ministerio de Ciencia e Innovación (MICINN) en el marco del proyecto PROS-Req (Referencia TIN2010-19130-C02-02).

## Referencias

1. Alexander, C., Ishikawa, S., Silverstein, M. A pattern language: Towns, buildings, construction. Oxford: Oxford University Press, (1977)
2. Bergin, J.: Fourteen Pedagogical Patterns (2002). <http://www.csis.pace.edu/~bergin/PedPat1.3.html>. Consultado 17/05/2011
3. Buendía, F., & Díaz, P.: A Framework for the Management of Digital Educational Contents Conjugating Instructional and Technical Issues, Educational Technology & Society, 6 (4), 48-59. (2003)
4. Cristóbal, J., Navarro, A; Buendía, F.: Caracterización de Catálogos de Patrones de elearning en Instituciones Universitarias. ISELEAR (2010)
5. DFES Task Force Report. Get On with IT - The Post 16 E-Learning Strategy Task Force Report. (2002) <http://www.dfes.gov.uk/elearningstrategy/news.cfm?aID=91> Consultado 02/06/2011
6. Dodge, B.: WebQuest: A Technique for Internet-based Learning. 2, Distance educator, Vol. 1, págs. 10-13. (1995)
7. EC: Communication from the Commission: E-Learning – Designing "Tejas at Niit" tomorrow's education. Brussels: European Commission. (2000)
8. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. s.l. : Addison-Wesley, (1995)
9. Horton, W: E-learning, by Design, Ed. Pfeiffer. (2001)
10. Jones, D., Stewart, S.: The case for patterns in online learning. <http://www.eric.ed.gov/DFS/ED448724.pdf> (1999)
11. Martínez-Ortiz, I. et al.: A Conceptual Introduction and a High-Level Classification. Cap 3 Educational Modeling Languages. Centro de Estudios Superiores Felipe II. Madrid, Spain;
12. Novak, J: Learning, Creating, and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations. 1998 Updated Routledge, (2010)
13. Object Management Group (OMG) <http://www.omg.org/>
14. OlivaNOVA es una herramienta de Care Technologies: <http://www.care-t.com>
15. Panach, I: Incorporación de Mecanismos de Usabilidad en un Entorno de Producción de Software Dirigido por Modelos. DSIC. Universitat Politècnica de València (2010)
16. Pastor, O., Molina, J.C.: Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling. Springer, New York (2007)
17. Rodríguez-Jiménez, J. M.: Patrones pedagógicos en educación virtual. RED. Revista de Educación a Distancia, número monográfico X (2009). <http://www.um.es/ead/red/M10> Consultado 26/02/2011
18. The E-dilema Project. programa Sócrates de la Unión Europea (2003) <http://ares.cnice.mec.es/informes/21/contenidos/25.htm> Consultado el 12 /04/2011.
19. The E-LEN Project (2005). <http://www2.tisip.no/ELEN/> Consultado 20/02/2011
20. Universidad de Viena: The Person-Centered e-Learning Pattern Repository (2006). <http://elearn.pri.univie.ac.at/patterns/> Consultado 10/03/2011

21. Weisburg, M.: Documenting good education and training practices through design patterns (2004). [http://ifets.ieee.org/discussions/discuss\\_june2004.html](http://ifets.ieee.org/discussions/discuss_june2004.html) Consultado 1/06/2011

### Apéndice 1

El DC completo con la información de los atributos y servicios de cada clase sería el mostrado en la siguiente figura:

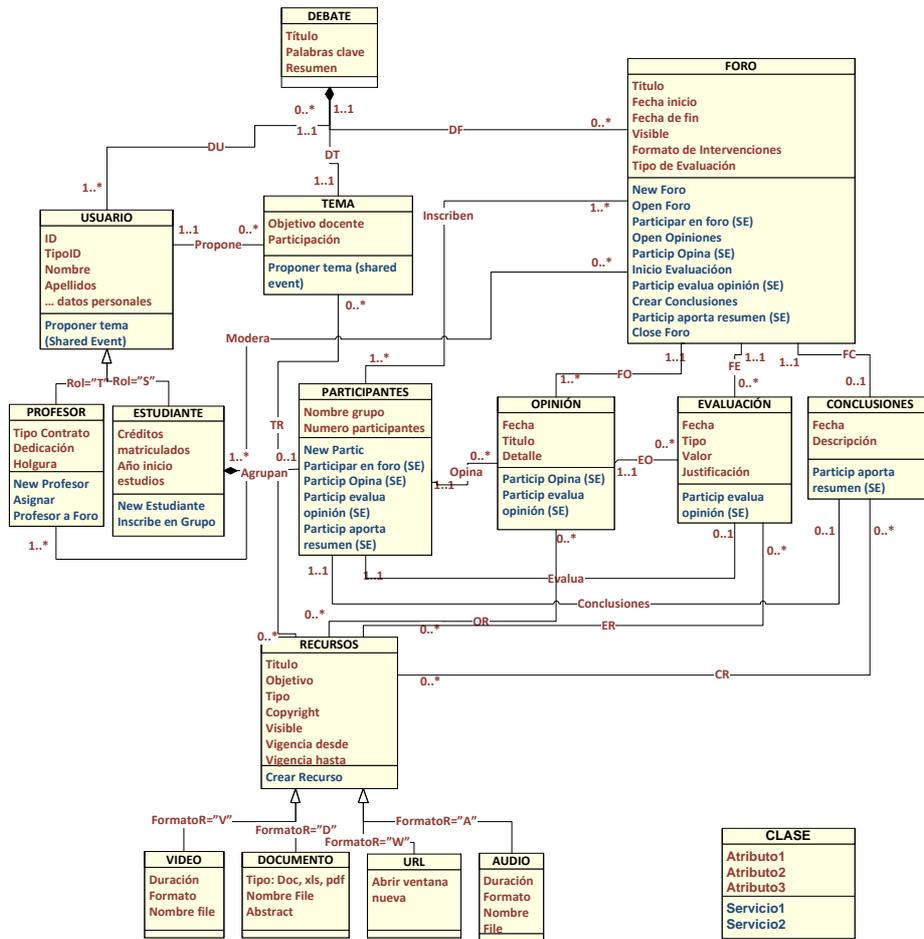


Figura 6. Modelo de Objetos: Diagrama de Clases - COMPLETO