Actas de las XXII Jenui. Almería, 6-8 de julio 2016
ISBN: 978-84-16642-30-4
Páginas: 319-326

Una herramienta visual para el diseño y la coordinación de asignaturas en una titulación

Jesús Martínez, Alejandro Fernández, Beatriz Velarde
Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga
29071 Málaga, España
jmcruz@lcc.uma.es

Resumen

En este artículo se presenta una herramienta para el diseño de asignaturas que permite la creación visual de conceptos a impartir, la relación entre ellos y las actividades propuestas para evaluar su comprensión. Cada asignatura puede ser compartida, de forma que aparecerá disponible para que otras asignaturas puedan relacionar sus propios contenidos con los ya existentes en ella. Al utilizar un desarrollo dirigido por modelos en la aplicación, la forma en la que se representa cada asignatura permite la creación rápida de nuevas herramientas: desde las que facilitan la presentación de la información en formatos compatibles para campus virtuales, SCORM, HTML, etc., como las orientadas a la gestión y coordinación de contenidos y evaluaciones en un mismo curso y titulación.

Abstract

This paper introduces a novel tool for the design and coordination of courses and curricula visually. The tool follows a model-driven development approach, where contents and activities are managed in a way that can be converted automatically into formats available in popular e-learning tools, such as Moodle (with SCORM) or HTML-based ones. It is worth noting that the tool is able to share course contents in a way that other courses can benefit from existing concepts and activities, which improves the overall coordination between them.

Palabras clave

Diseño de cursos, herramienta visual, coordinación, mapa conceptual

1. Introducción

El diseño de cualquier asignatura es un proceso vivo que debe tener en cuenta tanto los requisitos más clásicos (la titulación y el plan de estudios, el curso en el que se imparte, su carácter troncal u optativo, el número de créditos asignados, las competencias que se deben cubrir...), como también otros factores más dinámicos (el grupo clase, los medios materiales disponibles, espacios, horarios...). Con todo esto, se diseñan y planifican una serie de contenidos a impartir y unas actividades dentro de un itinerario formativo que evalúen la correcta adquisición de las competencias. El resultado de este proceso es parte de la *programación docente* de cada titulación.

Actualmente, cada universidad ofrece soluciones particularizadas para la confección y gestión anual de programaciones docentes. El procedimiento y los mecanismos empleados obviamente difieren, pero se comparte una tendencia común al uso de herramientas informatizadas donde se gestionan los elementos que caracterizan a cada asignatura para un curso académico. Independientemente de quiénes sean los encargados de la introducción o actualización de todos esos datos, se requiere de su revisión y aprobación última por parte de ciertas entidades de control, como las Comisiones de Ordenación Académica, Juntas de Centro, Vicerrectorado/s... Esta revisión suele ser menos automática de lo deseable, y el proceso completo consume muchas horas de todas las personas implicadas.

No se ha mencionado hasta ahora la imprescindible labor de coordinación entre asignaturas, una tarea continuada que debe contar con una serie de procedimientos de actuación que tiendan a ser lo más formales posibles. Dicha labor recae sobre el Coordinador de Grado/Máster, que ha de velar por el correcto desarrollo de un plan de estudios, en dos frentes. En la denominada coordinación vertical, se persigue que una asignatura tenga en cuenta el diseño de aquellas otras que se impartieron con anterioridad y sobre las que se basan sus contenidos. En la coordinación horizontal, se trabaja

para que las asignaturas de un mismo curso o cuatrimestre estén adecuadamente sincronizadas, en caso de que haya dependencia de contenidos, y las cargas de trabajo semanales sean adecuadas. Aún más crítico es el caso en el que varias asignaturas colaboren para la realización de un proyecto común.

Con todo lo expuesto, queda de manifiesto que la existencia de soportes formales (y automatizados), repercutirán en una labor menos costosa de diseño, mantenimiento, validación y coordinación de asignaturas. En este artículo se presenta una herramienta para dotar de este soporte formal a los procesos de diseño y coordinación: el *Diseñador de Cursos*. Esta propuesta ofrece un espacio de trabajo basado en el popular entorno Eclipse[1], que permite la confección de asignaturas mediante la creación visual de bloques temáticos y conceptos, las relaciones entre ellos y la secuenciación de actividades formativas. Para ello, se ofrecen dos formatos visuales a modo de organización: un Diagrama de Conceptos y otro de Actividades. Además, se puede dotar de contenido a cada tipo de actividad (adjuntando archivos con guiones de prácticas y esqueletos), describir su duración y/o decidir el porcentaje de calificación global.

Una importante novedad de esta herramienta consiste en el uso de un modelo de representación de asignaturas que sigue la metodología del desarrollo de software dirigido por modelos (DSDM) [2], lo que permite la creación de un ecosistema de herramientas de soporte y ampliación muy interesante, como la generación automática de contenido para Moodle o la Web, de informes y análisis para coordinación, etc. Tal y como se describe en el artículo, el uso de la metodología DSDM ha permitido incluir una característica fundamental: las asignaturas se almacenan en un repositorio de control de versiones, que permite la evolución de la misma y su compartición en red. De esta forma, varios docentes pueden trabajar en paralelo diseñando un curso e, incluso, pueden importar conceptos y bloques temáticos de asignaturas existentes que sirvan como base a nuevos conceptos a impartir en otras asignaturas.

El artículo está organizado de la siguiente forma. La sección 2 ofrece un respaldo teórico al uso de diagramas visuales para la representación de conceptos y actividades en una asignatura. En la sección 3 se describen las capacidades de la herramienta Diseñador de Cursos y en la sección 4 se dan detalles de su uso en coordinación. La sección 5 introduce las capacidades de adaptación y evolución de la herramienta para distintos contextos educativos y, por último, la sección 6 resume las conclusiones más importantes y destaca líneas de trabajo futuro.

2. Conceptos y diagramas: un poco de teoría

Las ventajas de la presentación de conceptos y sus relaciones a través de diagramas visuales son bien conocidas entre la comunidad educativa. En muchos casos, los docentes los usan para presentar los contenidos de un tema, pero también es habitual utilizarlos como ejercicios de síntesis para los estudiantes, o como técnica de estudio.

Uno de los diagramas más empleados es el mapa conceptual. Según la teoría de grafos, corresponde a un grafo dirigido y etiquetado, en el que cada nodo representa un cierto concepto, y las aristas que los unen etiquetan verbos que los relacionan. En el mercado existen herramientas con buenas características para crear y visualizar mapas conceptuales, como FreeMind[3] y CMaps Tools[4], ambas de libre disposición.

En un mapa conceptual, hay pocos límites para dotar de significado a las aristas del grafo. Pero lo que es bueno para poder expresar cualquier tipo de relación imaginable entre conceptos, no lo es tanto como para poder ser utilizados en la elaboración de asignaturas, donde el foco reside en ser capaces de extraer información precisa que ayude en la elaboración, más o menos automática, de programaciones docentes e informes de coordinación. Afortunadamente, el problema de cómo acotar el amplio margen de libertad expresiva en un mapa conceptual se ha tratado con éxito en dominios conocidos por la comunidad informática. Por ejemplo, el lenguaje visual UML define la relación entre entidades a través de un conjunto limitado de opciones[1]. En efecto, en los diagramas de clases que sirven representan un sistema software orientado a objetos, los nodos son clases, que se relacionan básicamente por herencia, realización, agregación y composición. De esta forma, a partir de un diagrama de clases bien construido se puede generar código fuente en algún lenguaje de programación orientado a objetos de forma automática y sin ambigüedad.

Por otra parte, la secuenciación de actividades en un curso se representa mejor con diagramas de flujo, también habituales en contextos informáticos. De nuevo, hace falta algún tipo de limitación en la capacidad expresiva de dichos diagramas y, así, UML propone los denominados diagramas de actividad.

Directamente inspirado en los diagramas de clase y de actividad UML se presentó una metodología para el diseño de asignaturas en [5]. La fig.1 plantea, en un diagrama de clases UML, un posible escenario donde se han diseñado clases que están enfocadas a definir, con fines meramente ilustrativos, algunos conceptos de un bloque temático enfocado en la enseñanza de

[1] Si bien se abre la posibilidad a definir nuevos tipos de relaciones a través de los denominados “estereotipos”

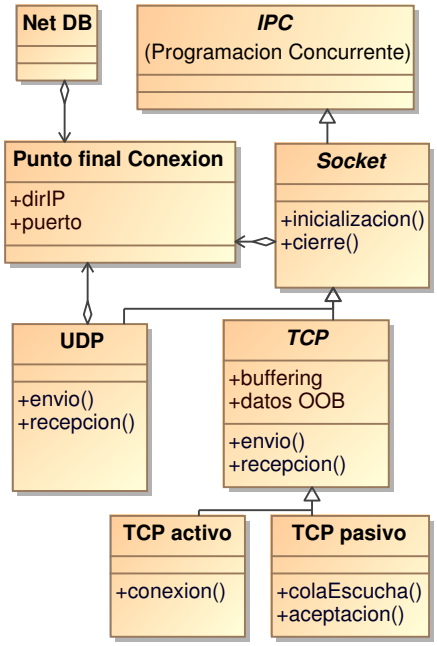

Figura 1: Relación entre conceptos (visión UML)

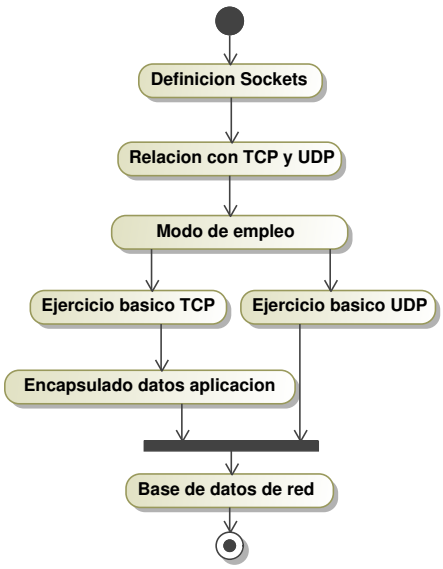

Figura 2: Secuenciación de actividades (visión UML)

software de comunicaciones para *TCP/IP*. En la figura se aprecia el concepto de *Socket*, un derivado del concepto de *IPC* (comunicación entre procesos), conocido desde un bloque temático anterior *Programación Concurrente*, y que a su vez se especializa en los sockets *UDP* y *TCP*. En este diagrama se ha decidido incluir en la parte de atributos alguna de las propiedades ligadas a cada concepto. Por ejemplo, en la definición de punto final de conexión se requiere una dirección IP y de un puerto de transporte. En la parte de operaciones, se ha decidido resaltar los métodos más importantes que utilizará el estudiante en los *Sockets* y que, en este caso, también requieren de explicación por parte del docente. El diagrama incorpora algo más de notación UML. Por ejemplo, las clases cuyo nombre aparece en cursiva se denominan abstractas, e indican que el concepto requiere ser ampliado por otras clases. En efecto, en la fig. 1 el concepto *IPC* se concreta en un *Socket* y, a su vez, el socket de tipo *TCP* será concretamente un socket TCP *activo* o *pasivo*.

Una vez que se tienen los diagramas con los bloques temáticos y los temas o conceptos a tratar, el siguiente paso en la metodología de diseño presentada en [5] consistía en organizar el flujo de aquellas actividades conducentes a su aprendizaje. Para esta planificación se recurre a un diagrama de actividad UML. La fig. 2 es un ejemplo de diagrama preparado para la enseñanza de los conceptos correspondientes al bloque temático de TCP/IP. El flujo de control permite observar cómo se avanza en el temario de forma intuitiva, a través de acciones de docencia. En el caso de la acción *Modo de empleo*, el control se bifurca para indicar que se pueden realizar dos acciones independientes entre sí, en paralelo, con sendos ejercicios para practicar sockets TCP y UDP, respectivamente.

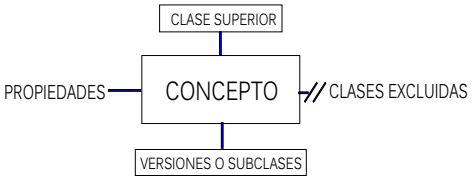

Figura 3: Esquema de un mentefacto

La herramienta que se presenta en este artículo, el Diseñador de Cursos, se inspira en [5], pero también toma prestadas características de otro tipo de diagrama visual para la representación de conceptos: el mentefacto (ver figura 3). Atribuido a Aristóteles, es un formato gráfico aún más esquematizado para representar la estructura interna de los conceptos de manera formal. Nótese cómo, en la figura 3, un concepto se relaciona con otros mediante generalización (o exclusión), e incorpora propiedades únicas y particulares que lo diferencian del resto.

3. La herramienta Diseñador de Cursos

El Diseñador de Cursos permite, como se ha comentado antes, crear una asignatura desde cero, de forma

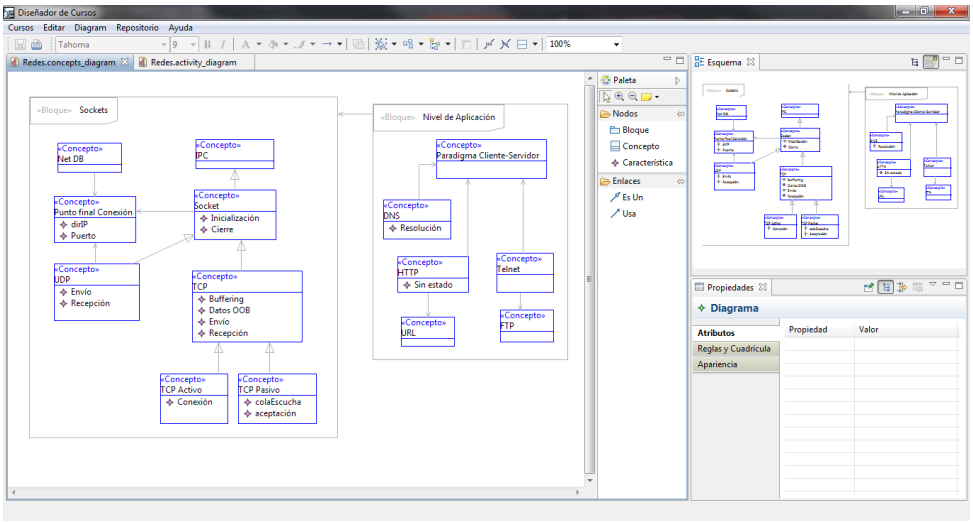

Figura 4: Ejemplo de diagrama de conceptos

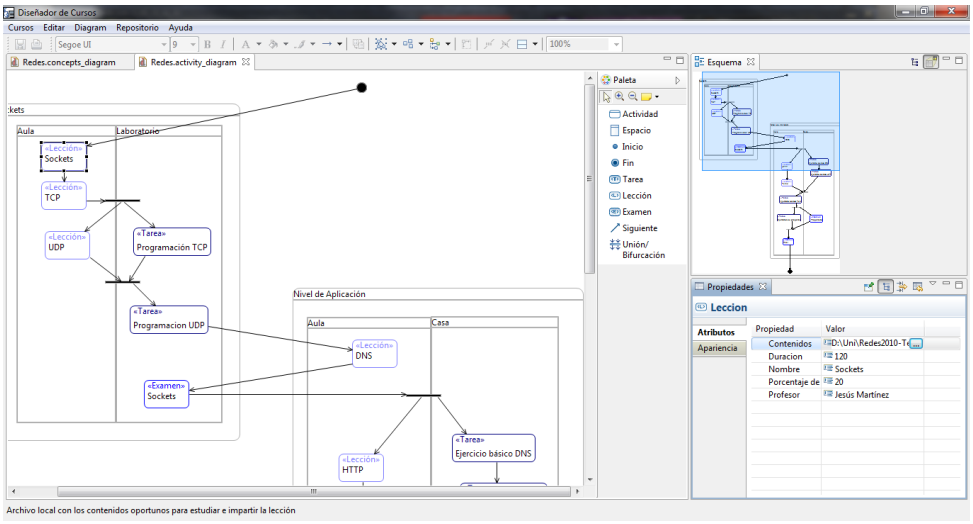

Figura 5: Ejemplo de diagrama de actividades

visual y agrupando conceptos en bloques temáticos. Las relaciones entre ellos siguen la nomenclatura del mentefacto (generalización y agregación en UML). La figura 4 muestra una captura de pantalla de un ejemplo de diagrama de conceptos, donde se puede apreciar el entorno de trabajo clásico de Eclipse, pero particularizado para nuestros propósitos. Los bloques y conceptos se vinculan a actividades, que se diseñan en su propio diagrama (ver figura 5). Cabe destacar que aquí la herramienta guía al usuario para que no existan errores en el diagrama (que el flujo empiece y termine adecuadamente, que se eviten auto-enlaces, etc.)

La herramienta se distribuye en un archivo comprimido que no necesita de ninguna instalación previa. Para arrancar la herramienta basta con ejecutar el fichero con extensión .exe presente en la carpeta en la que se realice la descompresión. Actualmente, se han generado versiones para Windows en 32 y 64 bits, aunque es posible obtener también versiones para otros sistemas operativos populares (como Linux o Mac OS X). Para su correcto funcionamiento, el requisito adicional es tener instalada la máquina virtual de Java (a partir del JRE/JDK 1.6).

Entre las acciones existentes como menús y botones en una barra de herramientas, se encuentran:

- Curso. Incorpora las opciones habituales de creación de una nueva asignatura, abrir una existente o cerrar una previamente abierta. También se permite refinar las características de una asigna-

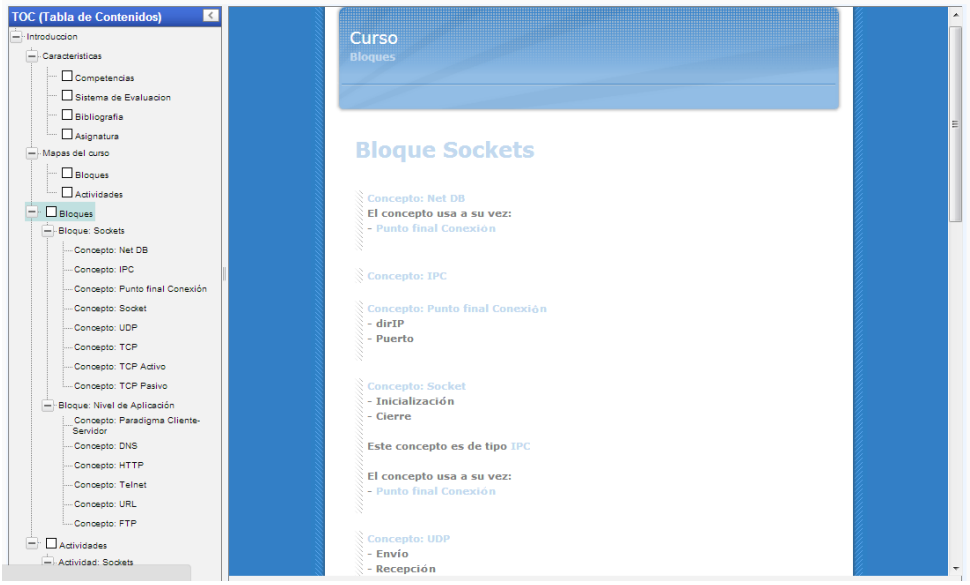

Figura 6: Aspecto del curso en formato SCORM

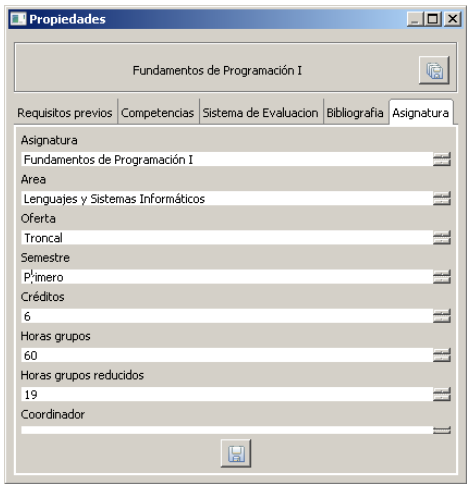

Figura 7: Características de una asignatura

tura (ver figura 7) y exportar a formato HTML y SCORM (figura 6). Por último, se dan opciones para la impresión de la asignatura actualmente abierta en el editor.

- Validar. Permite realizar un diagnóstico sobre la construcción de la asignatura. Se genera un informe en texto y se identifican aquellos elementos visuales ausentes que podrían impedir la correcta exportación automática a otros formatos.
- Diagrama. Contiene opciones relacionadas con la visualización y presentación de los elementos gráficos que forman cada diagrama.
- Almacén (EMFStore). Permite compartir la asignatura en un repositorio al que se esté conectado (tanto local como remoto). La herramienta incluye en su distribución un repositorio local para el almacenamiento y control de versiones de cada curso.
- Ayuda. Incorpora una breve guía de arranque y los parámetros de conexión con el repositorio local.

Por otra parte, el Diseñador de Cursos ofrece diferentes vistas al usuario. Por un lado, la vista del editor es el área de la aplicación donde se editan los diagramas. Está compuesta de dos pestañas, una en la que se edita el diagrama de conceptos del proyecto abierto, y otra en la que se edita el diagrama de actividades. En esta vista está incluida, además, la paleta, en la que se encuentran los distintos elementos que se pueden insertar en un diagrama (ver figura 4). Los elementos de la paleta que tienen tamaño variable (Tema, Concepto, Actividad, Espacio, Inicio, Fin, Tarea, Lección, Cuestionario y Unión/Bifurcación) se pueden insertar en un diagrama de dos formas:

- Pulsando una vez sobre el elemento en la paleta y pulsando a continuación en el diagrama. De esta manera, se añade el elemento al diagrama con el tamaño por defecto (que varía según el tipo de elemento).
- Pulsando una vez sobre el elemento en la paleta y pulsando a continuación en el diagrama, pero arrastrando el cursor mientras se mantiene pulsado el botón izquierdo del ratón. De esta forma se puede establecer el tamaño deseado para el elemento (aunque se podrá modificar en cualquier momento).

Los enlaces (*Es Un*, *Usa* y *Siguiente*) se insertan pulsando primero sobre el elemento en la paleta y después en el elemento origen en el diagrama, arrastrando el ratón hasta el elemento destino.

Otro tipo de vista es la vista de esquema, que presenta el diagrama completo en miniatura (figuras 4 y 5, arriba a la derecha). Allí se muestra sombreada la parte del diagrama visible en ese momento en la vista de diagrama. Por último, la vista de propiedades muestra características del elemento seleccionado en la vista de diagrama (figuras 4 y 5, abajo a la derecha).

3.1. El diagrama de conceptos

Cuando esta pestaña está activa, se permite utilizar estos elementos de paleta:

- *Tema*. Puede añadir un bloque temático sobre el fondo del diagrama y asociarse con otros bloques a través de la relación Usa.
- *Concepto*. Se puede añadir en el compartimento de un tema y asociarse con otros mediante las relaciones Es Un y Usa.
- *Característica*. Se puede añadir en el compartimento de un Concepto.
- *Es Un*. Representa un enlace que relaciona conceptos.
- *Usa*. Representa un enlace para relacionar temas y conceptos entre sí.

3.2. El diagrama de actividades

Cuando esta pestaña está activa, se permite utilizar los siguientes elementos de paleta:

- *Bloque de actividad*. Se añade sobre el fondo del diagrama y está asociado (de forma opcional) con un bloque temático en el diagrama de conceptos.
- *Espacio*. Representa un espacio de trabajo dentro de un bloque de actividad, con un nombre que tenga significado para el usuario del curso.
- *Inicio*. Es el nodo inicial del flujo en un bloque de actividad. Su carácter es obligatorio y único en un bloque.
- *Fin.* Es el nodo final del flujo en un bloque de actividad. Al igual que el elemento Inicio, es obligatorio y único en un bloque.
- *Diversas actividades (tarea, lección, etc.)*. Se enlazan entre sí creando la secuenciación en el tiempo de actividades. En la vista propiedades, se puede especificar la duración, porcentajes de calificación, adjuntar material asociado (que estará disponible como enlace en la exportación a HTML y SCORM), etc.
- *Unión/Bifurcación*. Es un elemento especial en el que se realiza la unión de distintos flujos y la bifurcación en nuevos flujos. Representa la simultaneidad en la realización de diversas actividades.
- *Siguiente*. Es el enlace que une un elemento con el siguiente en el flujo, incluyendo aquellos que pertenezcan a distintos espacios, pero contenidos en el mismo bloque de actividad.

Como resumen, el cuadro 1 resume las principales características de la aplicación, contrastándolas con las disponibles en las herramientas FreeMind y CMap Tools. Además, existen dos video-tutoriales disponibles en [6, 7], en los que se detallan algunas características específicas del Diseñador de Cursos, como la exportación a paquetes de e-learning SCORM.

4. Tareas de coordinación a través de la herramienta

El Diseñador de Cursos se ha empezado a utilizar en una tarea de coordinación vertical en los cinco grados de Ingeniería de Telecomunicación de la ETSI Telecomunicación de la Universidad de Málaga, durante el curso 2015/16. Dicha tarea es una acción de mejora encaminada a fortalecer las competencias de los estudiantes en programación. Para ello, los coordinadores de asignaturas con fuerte carga de programación han realizado mapas conceptuales que están sirviendo para detectar:

- conceptos propios de cada asignatura
- conceptos requeridos, impartidos previamente en otra asignatura
- conceptos requeridos, impartidos previamente, y que se amplían en una nueva asignatura

La acción de mejora, aún en curso, ha involucrado a todas las asignaturas listadas en el cuadro 2. Los mapas conceptuales resultantes de cada asignatura están sirviendo no solo para reflexionar sobre los contenidos a un nivel de detalle mayor del que se puede encontrar en las guías docentes o en la memoria verificada de cada titulación, sino que también están permitiendo unificar nomenclaturas, conceptos y encontrar formas de reubicar contenidos de unas asignaturas en otras. Por ejemplo, se está observado qué contenidos de las asignaturas troncales de primer curso podrían moverse a cursos superiores, allí donde dichos conceptos ya se estaban repasando y especializando con una orientación más aplicada. De hecho, una de las partes más delicadas de la tarea ha sido la identificación de conceptos heredados de una asignatura a otra (o bien simplemente requeridos, o requeridos para ser ampliados).

El Diseñador de Cursos ha permitido obtener los lienzos de cada mapa conceptual en forma de imágenes que, una vez difundidas, han constituido la base sobre la que establecer las discusiones pertinentes en-

Características / Herramienta	FreeMind	CMap Tools	Diseñador de Cursos
Interfaz	escritorio	escritorio y web	escritorio (web en proceso)
Permite compartir los mapas	no	sí	sí
Control de versiones	no	no	sí
Permite plug-ins de terceros	sí	no	sí
Auntenticación de usuarios	no	sí	sí
Capacidades de exportación	xml, html	imagen, svg, xml, html	imagen, svg, scorm, html
Licencia	GPL V2+	propietaria, de uso libre	propietaria, de uso libre

Cuadro 1: Comparativa entre herramientas

Asignatura	Curso	Cuatrimestre	Grado/s
Programación I	1	1	común a todos
Programación II	1	2	común a todos
Fundamentos de Software de Comunicaciones	2	2	común a todos
Microcontroladores	2	2	común a todos
Programación Concurrente	3	1	telemática, sist. electrónicos
Aplicaciones y Servicios	3	1	telemática
Sistemas Operativos en Tiempo Real	3	2	telemática
Sistemas Empotrados	3	2	sist. electrónicos

Cuadro 2: Asignaturas involucradas en la tarea de coordinación

tre coordinadores. La concreción en la forma en que se relacionan conceptos visualmente ha sido bien valorada, así como las capacidades de la herramienta. Se espera que los mapas conceptuales de cada asignatura se refinen a partir de las reflexiones obtenidas, y que se utilicen plenamente las capacidades de control de versiones y compartición de cursos disponibles en la aplicación.

5. Capacidades de adaptación y ampliaciones

El Diseñador de Cursos está basado en la plataforma Eclipse, que permite un diseño modular y flexible al estar basado en una arquitectura orientada a componentes (siguiendo el estándar OSGi [8]). Así, es habitual encontrar diferentes versiones de este popular entorno de desarrollo, orientadas a diferentes lenguajes (como Java o C++), pero también se permite extenderlo a través de *plug-ins*.

Una de las características de Eclipse es que permite crear una versión empaquetada, con determinadas extensiones, y redistribuible como ejecutable para diferentes plataformas, con lo que se denomina Rich Client Platform, o RCP. Siguiendo esta filosofía, el Diseñador de Cursos ha particularizado el entorno de ventanas, añadiendo menús y opciones orientadas al diseño de asignaturas. Para gestionar la parte gráfica y los datos asociados a una asignatura, se han utilizado las herramientas de Eclipse orientadas al Desarrollo Dirigido por Modelos [9]: Eclipse Modeling Framework (EMF), Graphical Modeling Framework (GMF), Graphical Editing Framework (GEF), MOFScript y EMFStore. El primero de ellos permite la creación de metamodelos que describen la estructura y relación entre datos (de ahí salen los metamodelos para conceptos y actividades). Los framework gráficos permiten, por un lado, la creación de elementos gráficos que representen a los de un metamodelo (con GMF) y su representación en el entorno de Eclipse (GEF). Gracias a estos elementos, se permite la creación de diagramas visuales para conceptos y actividades (que son modelos conformes a sus correspondientes metamodelos). MOFScript es una herramienta de transformación de modelos a texto, y aquí se usa para generar la exportación a páginas HTML y a SCORM. Además, se ha utilizado el lenguaje Object Constraint Language (OCL) [10] para expresar restricciones que deben cumplir los diagramas para considerarse válidos y poder exportarse correctamente (elementos con nombre, flujos bien definidos con inicio y fin, evitar auto-enlaces, etc.). Por último, la tecnología EMFStore permite serializar y gestionar los datos de cualquier modelo EMF como si de un repositorio de información se tratase. Es interesante hacer notar que no se trata de una simple base de datos (alojada en el disco duro local o en red), sino de un completo control de versiones que permite la colaboración a través de la edición simultánea de un mismo modelo y la detección de conflictos.

Las capacidades de adaptación de la herramienta son muy amplias, a través de nuevos *plug-ins* que particularicen las necesidades concretas de una universidad, centro o titulación. Algunas posibilidades incluyen asistentes para la creación a través de plantillas de los aspectos básicos del curso, el añadido de nuevos diagramas (y metamodelos), la generación de nuevos formatos de exportación o la conexión a servicios web, entre otras muchas.

Por otra parte, uno de los mayores problemas para cualquier herramienta está en su mantenimiento y evolución futura. En el caso del Diseñador de Cursos, habrá que lidiar con nuevas versiones de Java y sistemas operativos. Afortunadamente, desde hace poco existe la posibilidad de cambiar un entorno basado en Eclipse RCP hacia la web, utilizando la tecnología Remote Application Platform (RAP). De esta forma, todo el entorno de trabajo aparecería en un navegador web y, a través de Javascript, se podrían realizar las mismas funciones que en su versión RCP. Desde el punto de vista de los autores, la dependencia de las herramientas en línea tiene ciertos inconvenientes lógicos (más lentitud, dependencia de la disponibilidad de la red), por lo que abogan por la creación de esta nueva versión web y por su coexistencia con la actual.

6. Conclusiones y Trabajo Futuro

El Diseñador de Cursos presentado en este artículo es una herramienta que sobrepasa las meras capacidades de creación y relación visual de conceptos. Su enfoque también está orientado a la coordinación, en diversas facetas. Por un lado, permite a los docentes de una titulación la mejora en coordinación intra-asignatura (construyendo un curso de forma conjunta) e inter-asignatura (construyendo una asignatura a partir de conceptos y actividades existentes previamente en otras). Otra de las principales ventajas de la herramienta es que está basada en modelos fácilmente exportables e importables según el paradigma del DSDM, lo que posibilita la creación de nuevas herramientas ajustables a la idiosincrasia de cada titulación, centro o universidad. Por ejemplo, sería posible la elaboración automática de informes para un coordinador de titulación en los que aparecieran, para un curso y cuatrimestre, las actividades por semana, su grado de dificultad, etc. De esta forma, se facilitaría enormemente la labor seguimiento de cargas de trabajo excesivas. También sería posible, de nuevo de forma automática, revisar una guía docente y generar avisos a los docentes ante elementos incompletos o que no se adaptasen a la normativa. La principal línea de trabajo futuro con la herramienta consistirá, pues, en la incorporación de estas nuevas características como extensiones.

Referencias

[1] The Eclipse Foundation, "Eclipse." `http://eclipse.org`, 2016.

[2] C. Pons, R. S. Giandini, and G. Pérez, *Desarrollo de software dirigido por modelos*. Editorial de la Universidad Nacional de La Plata (EDULP)/McGraw-Hill Educación, 2010.

[3] Muller, Jorg and Polansky, Daniel and Novak, Petr and Polivaev, Dimitri and Foltin, Christian, "FreeMind: free mind mapping software." `http://freemind.sourceforge.net`, 2016.

[4] Florida Institute for Human & Machine Cognition, "CMapTools." `http://cmap.ihmc.us`, 2016.

[5] Martínez Cruz, Jesús, "Nuevos aliados en el diseño de asignaturas: UML y MDA para profesores," in *Actas de las XIV Jornadas de Enseñanza universitaria de la Informática*, pp. 513 – 520, 2008.

[6] Velarde, Beatriz, "Diseñador de Cursos: características generales." `https://www.dropbox.com/s/ld76m84p08mwi18/Caracteristicas.mp4?dl=0`, 2016.

[7] Velarde, Beatriz, "Diseñador de Cursos: exportación a SCORM." `https://www.dropbox.com/s/frjd8m29hpcflua/scorm.mp4?dl=0`, 2016.

[8] The OSGi Alliance, "OSGi: The Dynamic Module System for Java." `http://www.osgi.org`, 2016.

[9] The Eclipse Foundation, "Eclipse Modeling Project." `http://eclipse.org/modeling/`, 2016.

[10] Object Management Group, "OMG Object Constraint Language (OCL), Version 2.3.1." `http://www.omg.org/spec/OCL/2.3.1/`, January 2012.