

Aprendizaje práctico de patrones de diseño en asignaturas de programación de nivel III

Raúl Marticorena Sánchez, Carlos López Nozal, César Ignacio García Osorio, Carlos Pardo Aguilar

Escuela Politécnica Superior, Departamento de Ingeniería Civil,
Área de Lenguajes y Sistemas Informáticos, Universidad de Burgos,
e-mail: {rmartico, clopezno, cgosorio, cpardo}@ubu.es

Resumen

Uno de los múltiples objetivos de las asignaturas de programación de nivel III¹ es la presentación del concepto *patrón de diseño* y la aplicación por parte de los alumnos, de un conjunto limitado de patrones de diseño. Debido a la amplia cantidad de contenidos que se puede añadir a este tipo de asignaturas, se debe establecer un plan estratégico para poder presentar adecuadamente los patrones y la aplicación práctica de los mismos, siempre bajo un intento de motivar al alumno y con un pequeño coste de aprendizaje. El presente trabajo establece una propuesta de definición de un plan estratégico, una aplicación de ese plan en una asignatura de programación de nivel III, y una evaluación por parte del alumnado de la presentación práctica de la asignatura relacionada con patrones de diseño.

1. Introducción

La última actualización del *Computing Curricula 2001* [1], incorpora en muchas de sus diferentes propuestas de cursos de programación la unidad *SE1.Software Design*², dentro del núcleo de dichas asignaturas. Esta unidad incorpora el concepto de patrones de diseño para elevar el nivel de abstracción en el planteamiento de soluciones. Siguiendo esta recomendación se pueden asignar

¹ Se entiende por programación de nivel III la última asignatura de programación impartida en la titulación técnica.

² SE significa el área de Software Engineering y 1 el número de unidad dentro del área.

responsabilidades concretas a las asignaturas implicadas en la enseñanza de patrones con el fin de obtener el objetivo de aprendizaje de “*selección y aplicación de los patrones de diseño adecuado en la construcción de una aplicación software*” [1].

El siguiente paso es como transmitir este nuevo concepto a los alumnos y en qué asignaturas del plan docente. El presente artículo justifica en una primera sección la necesidad de añadir los patrones en asignaturas de programación un nivel III. En su sección posterior presenta un modelo *de aprendizaje por experiencia* [5] y un caso práctico de aplicación de este modelo. Un ciclo de aprendizaje por experiencia comienza con una experiencia que es guiada sistemáticamente para garantizar que se realizan los enlaces entre lo observado y la teoría o la práctica del concepto sobre el que se esta experimentando.

El artículo termina mostrando una evaluación del modelo de aprendizaje por parte de los alumnos junto con unas conclusiones.

2. Patrones de diseño en asignaturas de programación de nivel III

Aunque el tema de patrones de diseño puede ser introducido en apartados dedicados a diseño en las asignaturas de ingeniería del software, la experiencia nos ha demostrado, que en estas asignaturas, existe una tendencia general en el alumnado de ver este nuevo concepto de patrón como algo teórico difícil de aplicar. Este hecho esta originado por la falta de la maduración necesaria para comprender todos los elementos necesarios en la descripción de un patrón, en concreto todo lo relacionado con un mayor

conocimiento de un lenguaje de programación orientado a objetos. Esto dificulta la introducción completa de un patrón en asignaturas de ingeniería del software.

En la Tabla 1 se muestran los diferentes elementos necesarios para la descripción de un patrón según los autores [3] y [4]. En la tabla se han marcado en gris los elementos de las plantillas de descripción de patrones relacionados con la implementación. Para poder explicar estos elementos es necesario la utilización de algún lenguaje de programación orientado a objetos. Se justifica de esta forma la descripción detallada de estos elementos en las asignaturas de programación de nivel III.

Plantillas de GoF	Plantilla de Grand
Nombre y clasificación	Nombre
Propósito	Sinopsis
También conocido	
Motivación	Contexto
Aplicabilidad	Fuerzas
Estructura	Solución
Participantes	
Colaboraciones	
Consecuencias	Consecuencias
Implementación	Implementación
Código de ejemplo	
Usos Conocidos	Java API
Patrones Relacionados	Patrones Relacionados

Tabla 1 Plantillas de descripción de patrones de diseño

3. Propuesta del plan de aprendizaje de patrones en asignaturas de Programación Avanzada

3.1. Contexto específico de la UBU³

Antes de matizar algunos aspectos concretos del plan de estudios de Ingeniería Técnica de Informática de Gestión en la Universidad de Burgos, comentar que desde el prisma puramente técnico únicamente se pretende presentar los patrones como herramienta aplicable en el desarrollo de un sistema informático. En la titulación de Ingeniería Técnica no se pretende hacer un recorrido completo de un catálogo de

³ UBU Universidad de Burgos

patrones. Se delega la profundización en estos aspectos a la titulación de Ingeniería.

Asignatura	Créditos	Cuatrimestre
Análisis e Ingeniería del Software	9 + 3	3º y 4º
Metodología de la Programación	3 + 3	4º
Programación Avanzada	3 + 3	5º

Tabla 2 Asignaturas relacionadas con la docencia de patrones en el plan de estudios de la UBU.

Con el objetivo de poder establecer unos prerrequisitos en la materialización de dicho modelo de aprendizaje, se va comentar brevemente las dependencias con asignaturas relacionadas en el contexto particular del plan de estudios de Ingeniería Técnica en Informática de Gestión en la Universidad de Burgos (ver Tabla 2).

Las asignaturas de Ingeniería del Software y Metodología de Programación, ambas troncales, introducen paralelamente en el cuarto cuatrimestre los conceptos asociados a la orientación a objetos. Análisis e Ingeniería del Software desde la parcela del análisis y diseño, para ello utiliza el lenguaje de modelado UML. La asignatura de Metodología de Programación se centra más en un enfoque de implementación. Ambas hacen referencias casi al final del cuatrimestre a los patrones sin que los alumnos tengan tiempo para aplicar prácticamente los patrones de diseño.

La asignatura de Programación Avanzada o programación de nivel III de carácter optativo, se encuadra en una posición temporal perfecta (5º cuatrimestre) para poder madurar y aplicar los patrones de diseño pero también debe satisfacer otros objetivos:

- Mapeo de modelos de objetos. Java versus UML.
- Introducción a la programación orientada a eventos.
- Persistencia en Orientación a Objetos.
- Introducción a la programación concurrente.

Debido a los múltiples objetivos de la asignatura se necesita establecer un plan previo para que el alumno pueda aplicar patrones desde

un prisma de motivación y con el menor esfuerzo posible.

Es necesario establecer una coordinación entre las asignaturas de ingeniería del software y las asignaturas de programación para introducir completamente los conceptos de patrones de diseño en Ingeniería Técnica Informática. Esta coordinación se basa fundamentalmente en la elección de un mismo conjunto de patrones para poder hacer referencia a ellos en asignaturas que se impartan posteriormente en el tiempo.

A partir de este contexto específico se extraen las precondiciones necesarias que deben cumplir los alumnos para poder aplicar el plan:

- Tener claros los conceptos de orientación a objetos, y haberlos utilizado tanto a nivel de análisis y diseño como desde un punto de vista de programación con algún lenguaje concreto.
- Tener una idea intuitiva de patrón de diseño.

3.2. Propuesta del plan de aprendizaje de patrones de diseño

Típicamente la responsabilidad de la programación orientada a eventos propuesta por *Computing Curricula 2001* [1] suelen recaer en asignaturas de programación de nivel III. Generalmente se utilizan interfaces gráficas para introducir la programación orientada a eventos. Huelga decir que el alumno muestra un alto grado de satisfacción y motivación en este tipo de prácticas a pesar de la complejidad de las mismas. El plan se basa en poder utilizar esta motivación para preparar inconscientemente al alumno en la aplicación práctica del patrón y ver las mejoras que este aporta.

A continuación se enuncian las actividades y los artefactos⁴ conseguidos en cada actividad para establecer el plan.

1. Elección del patrón o patrones que se quiere que el alumno aplique. Se obtiene como resultado las plantillas de los patrones a aplicar.

2. Definición de un contexto sencillo donde poder aplicar el patrón. Se obtiene el enunciado preeliminar de un problema que pueda contener el patrón.
3. Proponer un enunciado de práctica envolviendo el contexto y añadiendo un interface gráfico al mismo para practicar la programación orientada a eventos. Se obtiene el enunciado de la práctica a realizar por los alumnos.
4. Realización de la práctica propuesta por parte de los alumnos, sin que tengan una explicación previa del patrón.
5. Después de la explicación teórica del patrón elegido por parte del profesor, refactorizar la práctica utilizando el patrón elegido analizando las mejoras introducidas por el diseño.

Las actividades 1, 2 y 3 requieren un esfuerzo coordinado del personal docente encargado en la planificación de la asignatura.

El objetivo primordial de la actividad 4 es proporcionar al alumno la posibilidad de proponer su solución, sin conocer la solución dictada por el patrón. De esta forma el alumno descubre la gran cantidad de tiempo empleado para solucionar el problema.

La actividad 5 se aprovecha del gran esfuerzo realizado inconscientemente por los alumnos en la actividad 4 para conseguir una aplicación del patrón elegido y que vean los beneficios que aporta la aplicación del patrón.

3.3. Precedentes

La elaboración del plan nace de la experiencia obtenida en el curso 2000-2001. En dicho curso, se planteó en primer lugar una práctica en modo texto y posteriormente llevarla a cabo dotándola de una interface gráfica, pero condicionados a la reutilización de las clases iniciales.

Como resultado de la misma, se observó que el alumno capta rápidamente la necesidad de diseñar módulos independientes reutilizables y a su vez comprueba la utilidad de que dichas clases se puedan incorporar rápidamente en la construcción de la segunda práctica, sin modificaciones, y acelerando de forma inmediata el proceso de desarrollo.

Pese a este éxito parcial, se observó que los diseños divergían en varias soluciones, que no

⁴ producto obtenido como consecuencia de una actividad

siendo del todo incorrectos, eran complejos e introducían problemas de implementación, no siempre fáciles de explicar por parte del docente. De hecho, la comunicación docente alumno se entorpecía debido al volumen de distintas soluciones, con la complejidad asociada a cada una y no siempre bien entendida por el propio alumno.

Ante esta situación en la que el alumno llega a la solución, pero no de la forma más adecuada, se plantea la necesidad de imponer algún tipo de estrategia que lleve a los alumnos hacia un diseño adecuado de la solución.

Dicho diseño no debe ser impuesto, sino razonado, a través de la idea de patrones, describiendo su necesidad y ventajas aportadas, a través del plan de aprendizaje.

3.4. Aplicación del plan

En este apartado se va a describir la aplicación del plan de aprendizaje propuesto en la asignatura de Programación Avanzada durante el primer cuatrimestre del curso 2001-2002.

Actividad 1: Elección del patrón. Patrón mediador.

Actividad 2: Definir un contexto sencillo donde poder aplicar el patrón. El patrón mediador puede aparecer en el diseño de sistemas (siguiendo la idea de construcción de sistemas en la orientación a objetos ya introducida en la asignatura de Metodología de la Programación [2]) cuando existen dependencias entre los diferentes módulos del sistema, necesitando tener referencias entrelazadas entre los objetos. Por tanto en la actividad 3 hay que establecer un diseño donde el acoplamiento y cohesión de los diferentes módulos sea el correcto, manteniendo todo el conjunto de propiedades inherentes a la programación orientada a objetos. En nuestra situación particular se buscará dependencias entre diferentes componentes gráficos.

Actividad 3: Proponer un enunciado de la práctica que contenga el patrón elegido. Para aumentar el grado de motivación de los alumnos se elige la realización de un juego donde esté contenido el patrón mediador en alguna de las pantallas de interacción con el usuario.

Enunciado propuesto: Caza del Barco

La práctica consiste en realizar un applet java que permita jugar a la caza del barco. El juego está pensado para que un jugador sobre un tablero intente hundir un barco que ocupa físicamente una casilla del mismo. La dificultad del juego estriba en que el jugador, no sabe donde está el barco, y lo único que se le muestra es un radio de posibles casillas alrededor de su tirada en la que el barco puede estar localizado. Para aumentar la dificultad el barco se mueve aleatoriamente siguiendo los movimientos del rey en el juego del ajedrez (en un radio de 1 casilla puede posicionarse en cualquiera de ellas). La interfaz debe aportar al usuario una forma sencilla e intuitiva de jugar de tal forma que pueda visualizar:

- el tablero con el radio de posibles casillas.
- el radio de acción bajo el cual puede estar situado el barco
- el número de tiradas
- opción de visualizar el barco (facilidad en el juego)

El juego debe ser configurable de tal forma que se puedan leer el número de casillas del tablero tanto horizontales como verticales y el número de tiradas. El applet debe ser visible desde cualquier navegador con soporte de Java.

Se deja libertad al alumno para emplear todas las posibilidades gráficas de AWT para el diseño de la interfaz. La Figura 1 muestra la pantalla de interacción con el usuario del juego Caza Barco, configurado con 10 casillas por 10 casillas y 10 tiradas

Aclaraciones:

Para el cálculo del radio nos basamos en colorear todas aquellas casillas que están a igual o menor distancia que el barco respecto a la tirada. (distancia Manhattan) Considerando distancia como:

- Si la posición del barco es (x,y)
 - Si la posición de la tirada es $(x1,y1)$
 - Distancia = $\text{abs}(x-x1) + \text{abs}(y-y1)$
- Se colorean todas aquellas casilla (i,j) donde:
- $$\text{abs}(i-x1) + \text{abs}(j-y1) \leq \text{Distancia}.$$

No se colorean casillas fuera del tablero.

La Figura 1 muestra una instantánea de interacción del juego donde se observan distintos resultados ejecutando la opción de visualizar el barco activada:

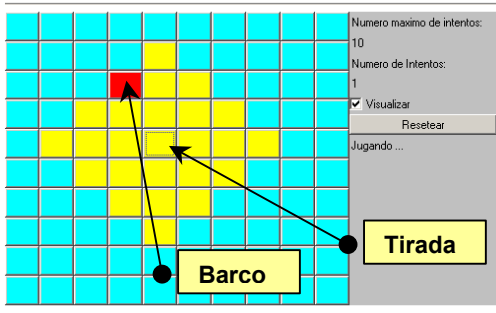


Figura 1 Instantánea de interacción del juego

En cada tirada el barco se mueve aleatoriamente con los siguientes parámetros

- en el eje x con valor +1, 0 y -1 (con igual probabilidad)
- en el eje y con valor +1, 0 y -1 (con igual probabilidad)

Por lo tanto se puede mover a cualquiera de las ocho casillas circundantes, lo cual aumenta la dificultad del juego, incluso se puede dar la posibilidad de no moverse si los dos desplazamientos son cero (ver Figura 2).

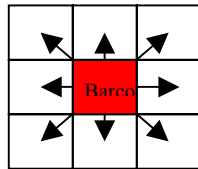


Figura 2 Posibles movimientos del barco

Sin entrar en detalle respecto a las especificaciones técnicas de la prácticas, por no afectar de manera directa al plan estratégico, se pasa a la siguiente fase.

Actividad 4: Realización por parte de los alumnos de la práctica propuesta. El diagrama de clases de la Figura 3 se corresponde con un mapeo estructural de la práctica propuesta por un grupo de alumnos.

Comentar que casi la totalidad de las prácticas funcionaba correctamente pero ningún diseño se adaptaba a la solución dictada por el patrón mediador a pesar de haber introducido este patrón en la asignatura de Análisis e Ingeniería del Software.

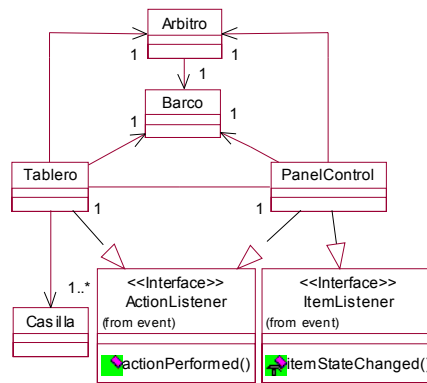


Figura 3 Artefacto actividad 4. Diagrama de clases

Los alumnos identifican claramente las abstracciones necesarias para construir software propuesto, pero se encuentran con dificultades a la hora de poder representar las dependencias existentes entre las abstracciones. El diagrama de clases de la Figura 3 presenta un mayor acoplamiento y una menor cohesión en sus clases que la solución propuesta por el patrón.

Después de evaluar las diferentes prácticas de los alumnos, se pudo observar que se encontraban en una situación idónea para que pudiesen ver los beneficios de la solución de aplicar el patrón mediador.

Actividad 5: Refactorizar la práctica aplicando el patrón elegido. El profesor propone la reestructuración de la práctica aplicando la solución del patrón (ver Figura 4). Este patrón presenta dos tipos de clase, las clases colegas (Tablero, Barco, PanelControl) cuyas instancias presentan dependencias entre ellas, y la clase mediadora (Mediador) cuya instancia coordina las dependencias entre las instancias de las clases colegas.

La clase mediador proporciona como parte de su interface métodos (registrarBarco, registrarPanelControl, registrarTablero) que permiten a los objetos de las clases colegas registrarse, para que el objeto mediador pueda gobernar todas las dependencias entre todos los colegas registrados. De esta forma, cuando alguna instancia de las clases colegas sufre algún cambio de estado se lo notifica al mediador a través de un evento, y éste aplica la lógica de dependencias de estados.

En este diseño propuesto por el patrón se aprecia que se reduce el acoplamiento, y aumenta la conexión de las clases participantes ya que las clases colegas se liberan de la responsabilidad de controlar su dependencia con otros objetos de las clases colegas.

En el contexto específico de la práctica las dependencias existentes son:

- Dependencia entre tablero y el panel de control. Si se activa la casilla de verificación visualizar en el panel de control el tablero debe visualizar la última posición en que estuvo el barco.
- Dependencia entre tablero y posición del barco. Cada vez que el barco haga un movimiento se lo debe comunicar a tablero para poder visualizar el radio en el que se puede encontrar el barco.

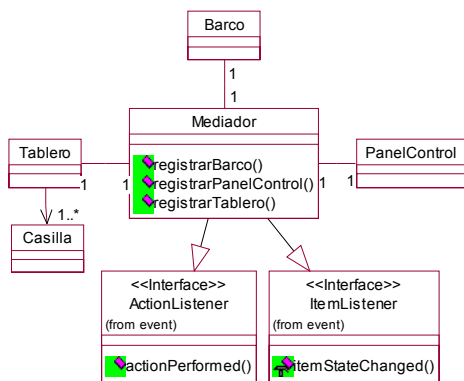


Figura 4 Artefacto actividad 5. Diagrama de clases.

3.5. Evaluación de resultados prácticos

Para poder evaluar los resultados obtenidos en la aplicación de este método se añadieron algunas preguntas en la encuesta de evaluación que se pasa a los alumnos al final del curso de la asignatura.

La Tabla 3 recoge los resultados de la encuesta sobre una muestra de 32 alumnos. En la columna de la izquierda se muestra el enunciado de las preguntas realizadas relativas a la evaluación de la asignatura que se corresponde de

alguna forma a la evaluación del tema presentado. En el resto de columnas se indica el resultado de la evaluación por parte de los alumnos indicado en tanto por ciento. Comentar que en la evaluación existían dos tipos de preguntas, unas ponderables con valores de 1 a 5 (P1, P2, P3, P4, P5), el otro tipo únicamente poseía las respuestas si, no o sin contestar y un campo textual para indicar las razones (P6).

En las preguntas ponderables con un rango de valores se añade un campo donde se indica el valor medio obtenido de la encuesta.

Ya sea con los datos estadísticos de la Tabla 3 o con su representación gráfica mostrada en la Figura 5 se aprecia la buena aceptación por parte del alumnado de esta idea. Aun siendo conscientes de que algún alumno no acepta o simplemente no está convencido, de que sea adecuado introducir este temario y por otro lado, tampoco se ha conseguido un éxito pleno en conseguir la puntuación más alta. Pero como norma general, se observa que la mayoría se mueven en una puntuación de aprobado y notable a la utilización e inclusión de los patrones.

Uno de los datos más destacables de la encuesta son los resultados obtenidos en la pregunta 6, donde se deja al alumno con una gran motivación para que pueda aplicar patrones en su trabajo fin de carrera.

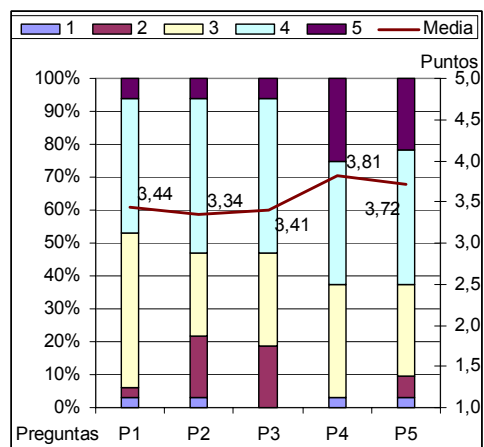


Figura 5 Gráfica de resultados de la evaluación

Enunciado preguntas	Media	1	2	3	4	5
		%	%	%	%	%
P1 Se ha incidido suficientemente en la aplicación práctica de la teoría	3,44	3,13	3,13	46,88	40,63	6,25
P2 Satisfacción del alumno con la parte práctica de la asignatura	3,34	3,13	18,75	25,00	46,88	6,25
P3 Grado de satisfacción con los ejercicios de prácticas	3,41	0,00	18,75	28,13	46,88	6,25
P4 Consideras interesante la explicación de patrones de diseño	3,81	3,13	0,00	34,38	37,50	25,00
P5 Ves útil la aplicación de patrones de diseño en prácticas	3,72	3,13	6,25	28,13	40,63	21,88
		% Si	% No	% NS		
P6 Intentarías utilizar patrones en tu trabajo final de carrera SI o, NO o no sabes ¿Por qué?		87,50	3,13	9,38		

Tabla 3 Resultados de la evaluación docente

4. Conclusiones

Los resultados obtenidos en las encuestas realizadas, confirman la consecución de los objetivos planteados en el plan estratégico. No sólo por parte de los docentes, existe una seguridad en la adecuación de los patrones de diseño como parte del temario de la asignatura de Programación Avanzada, sino que por parte de los alumnos, se logra transmitir y concienciar de dicha idea.

Los alumnos de ingeniería técnica no ven el concepto de patrones de diseño como algo puramente teórico difícil de aplicar. Elevan el nivel de abstracción, facilitando de esta forma la comunicación con el profesor a la hora de plantear y resolver nuevos problemas en los que se vislumbra la aplicación de patrones ya explicados. Se aprecian en la práctica los beneficios de la aplicación de patrones tanto a nivel de diseño como de implementación:

- Reducción del tiempo de desarrollo
- Mejora de documentación
- Aumenta la calidad del software

Además, se ha podido constatar el como llevar estas ideas más allá, en la realización de Proyectos Final de Carrera, facilita la comunicación del docente y alumno. Se maneja un vocabulario común y las soluciones planteadas por uno y realizadas por otro, son comprendidas y discutidas en un corto periodo de tiempo.

Experiencias prácticas en esta línea se confirman en la utilización de patrones observador

y mediador en el caso de proyectos con una fuerte carga de interfaz gráfica. Igualmente el uso de patrones como composite, de cara a estructuras complejas, en las que la composición es clave o patrones mucho más sencillos como singleton, que garantiza unas propiedades muy concretas, llevan a que los tiempos de realización se acorten, la documentación sea correcta y quede una buena base para que posteriores proyectos usen dichos documentos como base de trabajo. De hecho, el que las soluciones adoptadas y el vocabulario empleado en la resolución del proyecto sea común, anima aquellos alumno que han cursado la asignatura de Programación Avanzada a documentarse a partir del trabajo de sus compañeros, aumentando la reutilización en el proceso de producción de software.

Referencias

- [1] "Computing Curricula 2001. Computer Science" The Joint Task Force on Computing Curricula IEEE Computer Society. Association for Computing Machinery. Ed IEEE-CS y ACM.
- [2] Bertrand Meyer, "Object Oriented Software Construction", 2nd Edition. Ed Prentice Hall. 1997.
- [3] Erich Gamma, R. Helm, R. Johnson, Vlissides J. " Design Patterns. Elements of Reusable Object-Oriented Software". Ed Addison Wesley 1995.
- [4] Mark Grand. " Patterns in Java. Volumen 1". Ed. Wiley computer publishing. 1998.
- [5] Miguel Rebollo. " Aprendizaje activo en el aula". Actas JENUI 2001