

Problemas y soluciones en el aprendizaje del desarrollo de software orientado a objetos

Nelson Medinilla Martínez
Departamento de Lenguajes, Sistemas Informáticos e Ingeniería del Software
Facultad de Informática
Universidad Politécnica de Madrid

Resumen

Se analizan los problemas generales que presenta el paradigma orientado a objetos para su aprendizaje y las dificultades particulares que se derivan del contexto actual donde la tecnología todavía está en coacción y el paradigma procedural es hegemónico. A partir de estos antecedentes se exponen las soluciones que se adoptaron en cursos concretos y los resultados alcanzados.

Introducción

Por diversas razones el paradigma de desarrollo de software orientado a objetos se está extendiendo. Los centros de enseñanza, en consecuencia, tratan de aumentar el peso que le dedican a este paradigma en la formación de los profesionales del software. Pero se encuentran con dos dificultades fundamentales. Hay poca experiencia general respecto a la docencia del desarrollo de software orientado a objetos y además, esta forma de desarrollo de software aún está madurando.

El presente trabajo analiza los problemas del aprendizaje del paradigma orientado a objetos, las soluciones que se adoptaron durante los cursos 95/96 y 96/97 de las asignaturas de Sistemas de Información e Ingeniería del Software de cuarto y sexto año respectivamente, de la Facultad de Informática de la Universidad Politécnica de Madrid, y los resultados alcanzados.

Condiciones de la experiencia

Ambas asignaturas dedicaron alrededor de la cuarta parte de un semestre al paradigma de orientación a objetos y además se dio el mismo contenido. En las dos asignaturas se partió de la premisa que los alumnos desconocían el tema de objetos, aunque en Ingeniería de Software una parte de los alumnos habían cursado una asignatura optativa que lo trata. Esa premisa y el tiempo reducido de clases obligaron a cursos introductorios de orientación a objetos, donde se debían ver los conceptos básicos y la tecnología.

Problemas generales

- **Parece fácil pero no lo es**

A primera vista, el aprendizaje del paradigma orientado a objetos puede parecer fácil porque se apoya en una idea simple. “Los elementos conceptuales básicos del modelo de objetos son decepcionantemente simples: cualquier cosa es tratada como un objeto y los objetos se comunican entre ellos intercambiando mensajes” [2]. Se confunde simple con sencillo. Se requiere mucho tesón para lograr un buen sistema informático a partir de esa idea simple [3]. A continuación se discute por qué.

- **Es una representación con una visión fuertemente unipolar**

El paradigma orientado a objetos es una forma de representar sistemas informáticos, que tiene como base esa idea simple: la “*objetificación*”¹. Hay que ver cualquier cosa en el sistema como un empaquetamiento de estado y procesos, y realizar toda la computación sólo a través de la comunicación entre esos paquetes [4]. No hay más recursos para expresar el sistema que objetos y comunicación entre ellos.

Si bien “todo” se puede expresar en tales términos, en aras de su informatización, también es cierto que muchas veces resulta un modelo poco natural. La diversidad del universo se resiste a ser modelado de esa forma. Para Rosson esta no naturalidad es una cualidad favorable de la orientación a objetos porque obliga a profundizar en el problema [4]. Para el autor del presente trabajo, la no naturalidad es una fuente de oscuridad semántica del modelo.

No sólo se trata de ver los elementos del sistema en términos de objetos, sino que hay que ver también la computación como la interacción de elementos. Y esta noción es totalmente distinta de la idea de computación como transformación de información de entrada en otra de salida. Se debe pensar en un equipo de fútbol, en un panal de abejas..., que es una noción alejada del modelo clásico de computación y también de la percepción cotidiana de solución de problemas.

El paradigma orientado a objetos es una forma cualitativamente distinta de pensar, que requiere una interpretación diferente de la solución informática de los problemas, aunque comparta elementos comunes con el paradigma procedural: procedimientos, modularidad, encapsulación. De la misma manera que la visión geocéntrica difería profundamente de la visión heliocéntrica aunque coincidían los elementos: Tierra, Sol, estrellas, planetas [3].

- **No basta con comprender los principios hay que dominar la tecnología, pero esa tecnología todavía está en cocción**

Para que un modelo orientado a objetos sea el más adecuado, en una situación de ingeniería, debe aventajar a los otros modelos en cualidades de ingeniería. El hecho de que sea un modelo orientado a objetos no es una cualidad en sí misma, como a menudo se piensa. Conseguir esas ventajas requiere mucha pericia, específica para este tipo particular de modelización [5], [3]. Por tanto, no basta con comprender los principios teóricos, hay que desarrollar habilidades para lograr un modelo realmente ventajoso. Y éstas habilidades hay que desarrollarlas desde el comienzo del aprendizaje o al menos tenerlas presente.

¹ Palabra empleada por Rosson y Alpert para caracterizar la orientación a objetos y distinguirla de la abstracción de datos [4].

Las habilidades que se tengan en otras formas de modelar son muy de difíciles de transferir a la forma orientada a objetos dadas las profundas diferencias cognitivas entre unas y otras.

La pericia se hace como el camino, andando, pero se ofertan tantas rutas (metodologías) que es difícil escoger una definitivamente, porque además, como son productos de mercado, siguen llegando. Booch en su libro "Object Analysis and Design" se declara desarrollador de software, no "metodologista" y le pregunta a los "metodologistas" si usan sus propias metodologías para desarrollar software [1].

- **La representación procedural interfiere a la representación orientada a objetos**

En general, una forma arraigada de ver algo dificulta visión desde otra perspectiva. Los que llegan desde paradigma procedural tienden a tomar la orientación a objetos como un asunto de sintaxis [3], como otra manera de expresar lo mismo que ya saben. Se resisten a cambiar. Sin embargo, el problema va más allá.

Booch advierte que la forma algorítmica de modelar y la forma orientada a objetos son ortogonales. No es posible construir un sistema de ambas formas simultáneamente [1]. La presencia de una forma de modelado oscurece la otra. Por tanto, se aumenta la dificultad para acomodarse a la nueva visión.

- **Hegemonía actual del paradigma procedural**

La interferencia del paradigma procedural sobre el orientado a objetos y la confusión entre ambos es motivo de discusiones sobre por cuál comenzar los estudios informáticos.

Pero, en la práctica, no existe alternativa en muchos casos. La hegemonía actual del paradigma procedural provoca que la mayoría de los alumnos lleguen con esta forma de pensar arraigada a la primera clase de orientación a objetos.

- **Situación de la mayoría de las universidades**

Las universidades asimilan los avances con un desfase. La orientación a objetos no escapa a este proceso. En la actualidad, gran parte de las universidades, todavía le dedican poco tiempo al paradigma orientado a objetos y lo relegan a años superiores [5]. Y, aunque el panorama está cambiando favorablemente, estas condiciones son una parte importante del problema al que hay que enfrentarse hoy en día.

Formas de abordar los problemas mencionados

Las soluciones se apoyaron sobre varias ideas básicas.

Se prefirió ofrecer una aproximación simple y sencilla del paradigma en vez de un panorama exhaustivo y riguroso. " Si se quiere engendrar en los discípulos, verdadero y exacto, conocimiento de las cosas, hay que procurar que la enseñanza toda sea por medio de la propia intuición y la demostración sensual" [6]. "No puedo más que preferir un conocimiento intuitivo y difuso de algo, a una proposición clara y limpia, sin una intuición que la sostenga [7]. Booch, ya en el terreno de los objetos, refleja una elevada valoración del papel docente de la intuición, en el certero uso y diseño de las "simpáticas" figuras de su libro, ya citado.

Se aplicó la experiencia de un proyecto de la Facultad e IBM para la educación de software a través de ejemplos [8], que sigue la heurística: “ninguna lengua se aprenda por la gramática, sino mediante el uso de autores adecuados” [6], Coad utiliza un enfoque semejante en su libro, que define como el primer libro en el mundo que enseña un método software mediante ejemplos [9].

Se estimuló el pensamiento concreto. “Es el pensamiento concreto el que en la mayoría de los casos, si no en todos, merece la descripción de <lo auténtico> en el funcionamiento de la mente, mientras que los principios abstractos funcionan como herramientas al servicio de ese pensamiento concreto” [7].

• Qué ejemplos y cómo desarrollarlos

Puesto que el método educativo se basa en ejemplos es muy importante su selección. Al comenzar, el ejemplo debe estar muy próximo a la orientación a objetos y lejano del paradigma procedural, para reducir la interferencia. El propósito es trabajar en un terreno que difícilmente pueda ser interpretado desde un punto de vista procedural para minimizar sus evocaciones. El ejemplo debe expresar lo simple de la idea y además, ser familiar a cualquier persona.



Entonces, se inventó la modelización de una historia frecuente en los dibujos animados: un gato maúlla, despierta a un durmiente que enfurecido, lanza un zapato al gato y lo pone a “dormir”. Un antecedente de la descripción de historias como recurso docente en informática se encuentra en el origen de las tarjetas CRC (Class/ Responsibilities/ Collaborators) usadas inicialmente como una herramienta para enseñar programación orientada a objetos [1].

A través del desarrollo, exhaustivo e interactivo, del ejemplo se intenta que el alumno comience a crear su visión personal del paradigma. La pequeña distancia intelectual que hay entre el ejemplo y el modelo de objetos, así como su lejanía con la imagen procedural, facilitan la apropiación del paradigma. Una vez encontrado el ejemplo se ha usado de forma sistemática en todos los cursos.

El ejemplo del gato se desarrolla desde el análisis hasta el código porque conviene ver el paradigma en su conjunto, con el código incluido, puesto que es la forma tradicionalmente “concreta” de pensar de los informáticos.

Hay una razón adicional para tratar el código desde el principio en el paradigma orientado a objetos: el código está desde el principio en el modelo, describiendo objetos, atributos y responsabilidades, aunque sin desplegar todo su detalle. Una mejora significativa de la tecnología orientada a objetos es la capacidad para usar sólo un modelo desde el dominio del problema hasta el código [9].

Sin embargo, no hay que excederse en el código para tratar los principios básicos, porque la cultura del código puede desviar la atención hacia detalles innecesarios en esos momentos [10]. En esta experiencia, fue suficiente que los alumnos comprendiesen el código de implementación del ejemplo. Pero, es mejor que los alumnos lo lleven a la máquina y lo prueben allí, como se hizo, en su momento, con el paradigma procedural, lógico o funcional.

Los ejemplos dedicados a la tecnología de construcción de los modelos orientados a objetos toman una dimensión, que hace evidentemente secundario el código. Por tanto, no hay que preocuparse de su interferencia en esa fase.

Para seguir la línea de reducir las confusiones entre lo viejo y lo nuevo, se puede ver el código de un lenguaje puro de orientación a objetos. Pero en los cursos, para las asignaturas de Sistemas de Información e Ingeniería del Software, se prefirió un lenguaje híbrido, más cercano a lo que los alumnos consideran útil.

Detrás del ejemplo del gato, se continúa con otros ejemplos sobre la organización de una empresa y sobre el dibujo de figuras geométricas, que complementan la visión de los principios básicos de la orientación a objetos. Aunque la herencia y el polimorfismo se analizan desde el comienzo, se tratan con detalle en el ejemplo de las figuras geométricas.

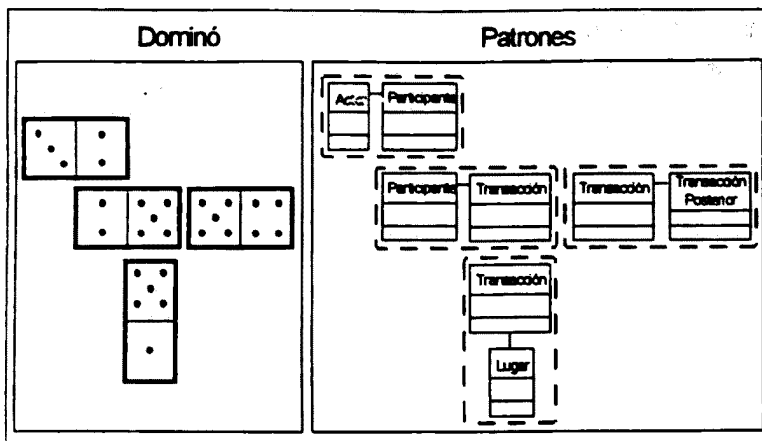
De manera reciente se ha incorporado como ejemplo previo al del gato, la pantalla de un procesador de textos para comenzar por un ejemplo informático donde se percibe mejor la imagen de elementos que colaboran, que la imagen tradicional de “entrada, proceso y salida”. La reacción de los elementos de la interfaz (botones, ventanas...) contribuyen a ofrecer una imagen de elementos “vivos” capaces de colaborar entre sí. Este ejemplo, sin embargo puede acarrear confusiones entre la programación basada en eventos y el paradigma orientado a objetos. Por tal razón y la complejidad del ejemplo, sólo se utiliza de introducción.

- **Aprendizaje de la tecnología**

Los cursos del paradigma orientado a objetos en las asignaturas de Sistemas de Información e Ingeniería de Software, deben estar dirigidos hacia la tecnología. Sin embargo, hay entre veinte y ochenta metodologías de desarrollo software, según el criterio que se tome para definir una metodología [11]. ¿Qué camino seguir en general y en particular, cuándo el tiempo es reducido ?

La decisión fue ofrecer los medios (ejemplos, discusiones y tareas) para promover el pensamiento concreto acerca de cómo desarrollar modelos orientados a objetos, utilizando, como siempre, un enfoque intuitivo. Y además, por su importancia en la transformación de la artesanía en la ingeniería, realzar mecanismos de reutilización, mediante la aplicación de patrones, en particular los propuestos en [9].

El proceso docente fue, primero, construir mediante discusión colectiva, un modelo de orientado a objetos, de un cajero automático. Después, descomponer el modelo para descubrir los patrones presentes. Y luego, construirlo otra vez, pero utilizando los patrones ya conocidos. El siguiente paso fue un ejemplo nuevo, empleando los patrones para modelizarlo, también mediante una discusión colectiva, que se dejó iniciada; no había más tiempo. Por último, los alumnos en grupos de tres, modelizaron un problema relativamente grande, y cercano a la realidad. Tuvieron que tomar diversas decisiones, entre ellas los propósitos y alcance de su modelo particular, a partir de una entrevista colectiva con un supuesto cliente (ajeno a ellos) que asistió a la clase.



En el curso se utilizaron los patrones propuestos por Coad debido a su sencillez y aplicabilidad. Se usaron, empleando la metáfora del juego de dominó, aprovechando su analogía con fichas de dos extremos, que se unen por alguno de esos extremos.

Aunque la motivación principal del empleo de los patrones fue enfatizar la reutilización, inspirados en los Lego²,

después se reflexionó sobre otros aspectos que realzan aún más la importancia de los patrones en la docencia.

Los patrones, en sentido amplio, constituyen estructuras mentales que participan en el pensamiento diario y también en el pensamiento profesional de los diseñadores, médicos, abogados, etc. La acumulación de ejemplos de diseño arquitectónico forman la casuística de los profesionales [12].

Los patrones ayudan a estimular la creación individual de esas estructuras mentales y a conformar una cultura que acepte y utilice explícitamente este modo de trabajo, como un recurso más.

Polya, citado por Papert, observó que la capacidad de los estudiantes para resolver problemas mejoraba cuando se les proponía seguir una simple regla: antes de hacer nada, pensad en otros problemas que sean parecidos a éste [7]. Los patrones son “problemas parecidos” que hay que concretar en cada caso. El pensar cómo concretarlos, en el marco que ofrece el patrón, propicia el aprendizaje. Los patrones son elementos concretos de contraste con los que se puede “discutir”, una “opinión a oír” en el oscuro proceso del análisis o el diseño.

El objetivo de los Lego es constituir un medio divertido de aprender; la posibilidad de reutilizarlos sólo es una condición secundaria: estimular la creatividad. Esta esencia matética³ de los Lego es una de las mejores cualidades de los patrones en el contexto docente.

Resultados

Los alumnos lograron, por sí solos, aplicar los conceptos básicos de la orientación a objetos y pasos tecnológicos a la resolución de problemas sencillos de gestión: de una biblioteca, de un aeródromo. Dada la base nula de los alumnos y el reducido tiempo del curso, tales resultados son satisfactorios. Además, se percibió una alta motivación de los alumnos, que en muchos casos se sintió como un compromiso individual con el tema. Las encuestas fueron favorables.

Sin embargo, como era de esperar, el nivel de habilidades para lograr las ventajas potenciales de la orientación a objetos, flexibilidad, facilidad de mantenimiento, reutilización, es insuficiente. Los alumnos tienen una visión “natural” de los objetos y todavía contaminada por el paradigma procedural. Deben desarrollar la visión informática de los objetos, que es la que aporta las ventajas.

² Marca comercial de juego de construcción.

³ Vocablo usado por Papert para designar la heurística del aprendizaje [7].

Los ejemplos que se emplearon para el aprendizaje de los conceptos básicos favorecieron la visión de los objetos “naturales” (tangibles), en detrimento de los objetos “informáticos”. En los ejemplos usados para ilustrar la tecnología se vieron objetos informáticos pero sólo sirvió para que los comprendieran. Resulta muy difícil la adquisición del concepto informático de objeto, sobre todo por su confusa relación con los objetos del mundo real.

Los sistemas informáticos que modelaron los alumnos mostraban una tendencia a la centralización común del paradigma procedural, a pesar de que se les insistía en hacerlo de manera contraria, advertidos por el trabajo de Guzdiel. Es frecuente encontrar agrupamiento de funciones en un solo objeto. La herencia y el polimorfismo quedaron en un nivel de comprensión, pero no de aplicación.

Aunque el camino para desarrollar los modelos fue de libre elección, muchos grupos utilizaron los patrones en los ejercicios de evaluación. Las soluciones, en esos casos, fueron más uniformes, pero el aporte perceptible más importante de los patrones fue su contribución a la visión informática de los objetos y a la descentralización.

Conclusiones

Es posible aprovechar las posibilidades actuales de las universidades para mejorar las capacidades de sus graduados en el área de orientación a objetos. El punto de ataque a este problema es estudiar, en la práctica, los factores que influyen en el aprendizaje particular del paradigma orientado a objetos.

En el presente trabajo se hizo un estudio teórico y práctico de algunos de esos factores. Se demostró que es posible lograr una independencia aceptable de los alumnos en cursos de carácter introductorio. Los resultados en las asignaturas de Sistemas de Información e Ingeniería del Software así lo confirman. Pero también se demostró que tales cursos son insuficientes para desarrollar las habilidades que hacen ventajosa la orientación a objetos sobre el paradigma procedural.

Bibliografía

- [1] Booch, Grady. “Object-Oriented Analysis and Design with Applications”. Second Edition, Rational, Santa Clara, California 1994.
- [2] Thierry Demay and Stephen King “BPR and OO: origins and common ground” Computer Bulletin Feb 1996, pp 8-9.
- [3] Luker, Paul A. “There’s more to OOP than Syntax”. SIGCSE BULLETIN, Vol. 26, N° 1, March 1994, pp 56-60.
- [4] Mary B. Rosson and Sherman R. Alpert “The Cognitive consequences of Object-Oriented Design Human Computer Interaction. Vol 5, 1990, pp 345-379.
- [5] West, Martin. “Object Technology: an overview”. Computer Bulletin, February 1996, pp 2-3.
- [6] Comenio, J. A. “Didáctica Magna”. Editorial Reus, Madrid 1922.
- [7] Papert, S. “La máquina de los niños” Ed. Paidós, Barcelona, 1995.
- [8] Medinilla, Nelson “Cursos IBM-FIM. Orientaciones metodológicas generales” Informe Interno. Facultad de Informática. UPM. Marzo 1994.

- [9] Coad, Peter. "Object Models. Strategies, Patterns, & Applications". Yourdon Press, Prentice Hall Building, Englewood Cliffs, New Jersey 07632, 1995.
- [10] Rout, T.P. "The Culture of Quality and Software Engineering Education". SIGCSE BULLETIN, Vol 24, N° 2, June 1992.
- [11] Henderson-sellers, B. "Convergence Is in the Air" Report on Object Analysis & Design, March-April 1996, pp. 47-49.
- [12] Shulman, L.,S. "Paradigmas y programas de investigación en el estudio de la enseñanza: una perspectiva contemporánea" en Wittrock, Merlin C. "La investigación de la Enseñanza, I", Paidós Educador, 1989.