Paradigma de Objetos

Departamento de Computación, FCEyN, UBA

2016

“f u cn rd ths, u cn gt a gd jb n cmptr prgrmmng” Anónimo

f u cn rd ths, u cn gt a gd jb n cmptr prgrmmng

"Un médico, un ingeniero civil y un computador científico discutían sobre cual era la profesión más antigua del mundo. El médico remarcó, -Bueno, en la Biblia, se dice que Dios creó a Eva de una costilla tomada de Adán. Esto, claramente, precisa cirugía, por lo que puedo clamar que la mía es la más antigua profesión del mundo.-
El ingeniero civil interrumpió y dijo -Pero aún antes en el libro de Génesis, se dice que Dios creó el orden de los cielos y la tierra a partir del caos. Esa fue la primera y ciertamente más espectacular aplicación de la ingeniería civil. Por lo tanto, mi querido doctor, usted está equivocado: la mía es la más antigua profesión en el mundo.-
El computador científico se reclinó en su silla, sonrió, y entonces dijo con confianza, -Ah, pero ¿saben QUIÉN creó el caos?-" Anónimo

f u cn rd ths, u cn gt a gd jb n cmptr prgrmmng

Crisis del Software y problemas del paradigma tradicional

Los problemas de ser imperativo...

- No se delega
- Todo en la cabeza
- Uno decide por otros
- No hay confianza

Código duplicado

La cohesión entre las librerías o funciones es por asociación o temática, no por objetivo ni por delegación.

Modelo diseccionado

Los elementos modelados están dispersos

"If you can't write it down in English, you can't code it." *PeterHalpern*

Software Orientado a Objetos, una definición posible...

La construcción de software orientado a objetos es un método de desarrollo de software que basa la arquitectura de cualquier sistema de software en módulos deducidos de objetos del dominio, en vez de funciones o de funcionalidades que el sistema supone brindar.

Otra...

... is to provide a natural and straight-forward way to describe real-world concepts, allowing the flexibility of expression necessary to capture teh variable nature of the world being modeled, and the dynamic ability to represent chaging situations...[1]

[1] http://web.media.mit.edu/ lieber/Publications/Treaty-of-Orlando-Chapter.pdf

Fundamentos

- Programa: Modelo computable
- Modelo: Simulación
- Cómo: Envío de mensajes
- Objeto: Representación de un ente[2] del dominio del problema
- Mensaje: Conjunto de colaboraciones entre objetos
- Diálogo: Al modelar los conceptos en relación con la realidad, es natural el diálogo que se da entre los objetos y trasciende el ámbito informático

Resultado

Objetos que **colaboran** entre sí enviandose **mensajes**
Definición del paradigma axiomática (minimalista) que permite construirlo sin influencias (ni injerencia) de otros paradigmas.

---

[2]ente: elementos tangibles o intangibles que identifico lo largo del proceso de aprendizaje.

Motivación

El objetivo de su práctica en un grupo de trabajo es

Hacia adentro

- Productividad
- Reuso
- Calidad
- Costo
- Tasa de Errores

Hacia afuera

- Unificar criterios
- Integración con otros grupos
- Lenguaje común

Motivación, Mitos

Si les preguntara cuales son las ventajas del paradigma...?

- La ventaja esta en el lenguaje.
- Es mas fácil de programar
- Se escribe menos código

... todos equivocados

Motivación

Todo es un... modelo

Que todo sea un objeto es una consecuencia, no un objetivo.

El paradigma busca...

- Qué: un modelo
- Cómo: con mensajes entre objetos[3]

Por qué?

- Nos permite engañarnos y ser minimales en el alcance
- Nos permite engañarnos y ser tan simples como se quiera

Ejemplos de modelos

- Física: Núcleo atómico y electrones orbitales
- Matemática: Funciones

exitosas mentiras con las que todos convivimos... felices

[3] Nótese que 'objeto' es el último término en aparecer

Desarrollo de Software

Se tiene un modelo...
El modelo falla...
Se reformula el modelo...

Proceso de Aprendizaje

- Iterativo
- Incremental
- El conocimientos se genera contrastando con la realidad
- El nuevo conocimiento se incorpora al anterior en forma organizada

Mensajes

Los mensajes definen al objeto

- Se sacude como perro
- Hace ruidos como de perro
- Huele a perro

Entonces tenemos un ...

- Responde a la invocación de su nombre.

Mensajes

Los mensajes dicen **qué** saben hacer los objetos.
Mediante estos mensajes que saben responder los objetos definen su comportamiento, su esencia.

Esencia

Conjunto de mensajes minimales (al sacar uno de ellos, el objeto deja de representar al ente del dominio del problema)

Y por qué no darle más responsabilidad?

- Acoplamiento (ej, cuenta bancaria con msg asXml)
- Desfavorece el reuso
- Lo hace menos cohesivo (deja de representar solo una cosa)
- Lo hace más difícil de entender
- ...

$$+\text{Responsabilidad} \equiv +\text{Incumbencias}$$

Mensajes

Qué es un mensaje?

Un objeto.

Bueh, **Cómo** es un mensaje?

Es una lista de colaboraciones entre objetos mediante el envío de mensajes.

Ufa!, Qué pinta tiene **un** mensaje?

Un Método es lo que llamamos a una lista concreta de colaboraciones [4] asociado a un mensaje que un objeto dado sabe responder.

[4] un cacho de código

Mensajes, Relación de Conocimiento

Comunicación entre objetos

- Dirigida: El receptor esta presente y es alcanzable (visible) al momento del envío.
- Sincrónica: El envío de un mensaje es bloqueante
- Siempre hay respuesta
- Anónima: El receptor no conoce al emisor, la respuesta no varía en función del emisor.

Mensajes, Colaboradores

Colaboradores internos

Por una cuestión de comodidad e implementación algunos lenguajes permiten definir colaboradores internos en referencias nominadas según el rol que cumple el colaborador para el objeto.

Ejemplo

Las referencias pueden ser símbolos implementadas con variables. Una implementación posible de la clase *Fraccion* podría tener en su representación interna dos variables *numerador* y *denomindaor*.

Smalltalk, Self

self

Forma en la cual un objeto se refiere a si mismo.

Pseudo-variable

- Una de las palabras reservadas.
- Siempre esta y no se puede asignar.
- Representa al objeto que recibió el mensaje y esta ejecutando el método.

Smalltalk, Mensajes - Tipos y Sintaxis

Tipos

- Unarios
- Binarios
- Keyword

Se evalúan de izquierda a derecha por pasadas: primero unarios, luego binarios y finalmente keywords

Ejemplos

- anArray **size**, 1 **factorial**, etc.
- 1 + 2, prefix**,** self name [5], etc
- anArray **at:** 1 **put:** 2, aList **add:** 2, etc

anArray at: 2*10 factorial*3 put: name size ==
(anArray at: ((2*(10 factorial))*3) put: (name size))

[5](notar que el mensaje binario es la ",")

Polimorfismo

Cúando hay polimorfismo?

Dos objetos son polimórficos **con respecto a un conjunto de mensajes** si ambos responden a estos mensajes con la misma semántica.

- $1 + 2$
- $1.0 + 2.0$
- '1' + '2' ... no, no

Polimorfismo

Cúando hay polimorfismo

Dos objetos son polimórficos **con respecto a un conjunto de mensajes** si ambos responden a estos mensajes con la misma semántica.

- El único acoplamiento es el conjunto de mensajes con el que dos objetos se relacionan entre sí.
- A acciones similares, mensajes iguales (reduce el vocabulario, refuerza la semántica)
- Delegación, no importa el receptor mientras respete la semántica.

Máximas

Se necesita más delegación cuando...

Repeated Code

- Distintos objetos, Missing Object. Forwarding[6]
- Mismo objeto, Missing Method. Delegación

Conditional code

"Real Devs Don't use IF"

- Missing Objects. Polimorfismo

[6] O 'delegación por forwarding'

Construción del conocimiento[7]

Se tiene un modelo...
El modelo falla...
Se reformula el modelo...

Cómo introducimos el nuevo conocimiento?

Mecanismo de Sharing (del conocimiento)

- Clasificación, Sharing by inheritance
- Prototipado, Sharing by delegation

[7] http://web.media.mit.edu/ lieber/Lieberary/OOP/Delegation/Delegation.html

Prototipado

- Todos los entes son de similar jearquía, no hay abstracciones.
- Los objetos pueden tener padres (o prototipos) en otros objetos
- Permite trabajar con los objetos sin necesidad de tener clases
- Permite definir comportamiento por objeto.
- Crear objetos = Clonar objetos.
- Permite modelar sin generalizar ni abstraer.

Clasificación

Clases

- Por cada entidad del problema a modelar existen una o mas instancias de esa entidad y la clase que las define.
- Exige pensar un nombre. Tipado estatico una complicacion. Tipado dinamico un feature
- Su esencia es saber crear instancias y definir el comportamiento de sus instancias.
- También administran asuntos de clase.
- Si el lenguaje es estáticamente tipado, es un problema pensar el protocolo a implementar por la clase de antemano.
- Method LookUp mas elaborado.Tipado estatico muy complicado. Tipado dinamico mucho mas flexible con varias optimizaciones

Clasificación

Si entendí bien...

Todo es un objeto, por lo tanto, una clase es un objeto, y los objetos solo existen cuando su clase existe.

- Cuál es la clase de una clase?
- Cuando envío el mensaje new a una clase. Quién responde?
- Si necesito que una clase tenga un new distinto?

Subclasificación

Platón habló de Clases, pero Aristóteles agregó:
Hay una cierta relación entre Abstracciones, por ejemplo, los Autos son Vehículos (veía el futuro).

Relación de Subclasificación

- Permite decir que hay conceptos más abstractos que otros. No suele haber realizaciones concretas de esas clases más abstractas.
- Una subclase es algo más concreto que una super clase.
- Una de las herramientas **peor** usadas del paradigma.
- La herencia es un mecanismo para construir teoría, no para ahorrar código.
- Programming by difference [8]

[8] www.cse.msu.edu/ cse870/Input/SS2002/MiniProject/Sources/DRC.pdf

Method Lookup

Cómo responden los objetos a los mensajes?

Otra pseudo-variable: Super

Cómo funciona super?

super = self ?

Closures

Permiten representar un conjunto de colaboraciones.

Bloque, Closure o Full Closure?

- Diferencias.
- Closure, son una mejora a los bloques, agregan Binding Lexicográfico a los bloques.
- Full Closure, agregan además un control apropiado de flujo.