

TADP – Trabajo Práctico # 2

Framework de Aspectos

Diseño de solución

Grupo # 3

Docentes

Demian Alonso

Ayudantes

Axel Bergh

Leandro Shokida

Alumnos

Guillermo Grillo

Federico Jaite

Hernan Ceballos

Javier Joaquin

Santiago Manopella

La solución que pensamos para la implementación de este motor de aspectos fue básicamente dividir las tareas entre el motor y los join-point y así poder delegar responsabilidades.

En nuestro diseño, el encargado de buscar todas las clases y métodos que hay en el sistema es el motor de aspectos mediante un método privado llamado “`todos_los_metodos`”. Este método va a cargar un hash con el nombre de la clase como key y todos sus métodos (privados y públicos; de clase y de instancia) como valores.

Por otro lado, los join-point van a estar encargados de filtrar los métodos que apliquen una condición en particular. Cada join-point va a tener que tener definido un método llamado `filtra_metodo?(clase,metodo)` en donde va a decir si ese método o clase está siendo afectado por el filtro. Todos los join-point heredan de una clase `JoinPoint` y esto permite que si se quiere agregar un nuevo join-point solamente hay que crear una nueva clase que herede de `JoinPoint` y que implemente el método `filtra_metodo?(clase,metodo)` para poder utilizarlo.

Los point-cuts están implementados como un Mixin el cual separa los operadores unarios de los binarios. El operador unario es “no” y los binarios son “y” y “o”. Para crear los point-cuts simplemente se operan a uno o más join-point con estos operadores y simplemente se le pregunta al point-cut si puede filtrar un método, de la misma manera que se hace con un join-point en particular. Los point-cuts terminan siendo arboles de join-point operados con sus operadores.

El motor de aspectos también va a ser el encargado de filtrar todos los métodos del sistema (y más adelante de aplicarle los advices) a través de un point-cut el cual se puede llamar con el método `aspecto(point_cut,advice)`. Para lograr esto, el motor va a enviarle al point-cut definido el mensaje `filtra_metodo?(clase,metodo)` por cada método que haya en el sistema y va a guardar los resultados en un array para luego aplicarle el advice.

El diagrama de clases se encuentra en la página siguiente.

