Chain of Responsibility

Propósito

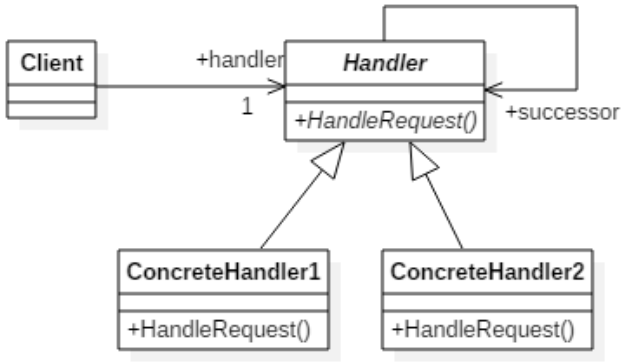
Evita acoplar el emisor de una petición con su receptor dando la oportunidad, a más de un objeto, de gestionar (o manejar) la petición. Encadena a los objetos receptores y pasa la petición a lo largo de esa cadena hasta encontrar un objeto que la atienda.

Aplicabilidad

El patrón Chain of Responsibility se utiliza cuando:

- Más de un objeto puede manejar una petición y no se conoce a priori quién es el que la manejará. El manejador se decide automáticamente
- Se desea emitir una petición a más de un objeto sin especificar explícitamente el receptor
- El conjunto de objetos que puede manejar una petición se debe especificar dinámicamente

Estructura



Participantes

- Handler: define la interface para manejar peticiones. Opcionalmente podría implementar el enlace sucesor.
- ConcreteHandler:
  - Maneja las peticiones de las que se responsabiliza
  - Puede acceder a su sucesor
  - Si puede manejar la petición, lo hace. En otro caso la reenvía al sucesor
- Client: inicia la petición dirigida a un objeto ConcreteHandler de la cadena

Colaboraciones

- Cuando un cliente emite una petición, ésta se propaga por la cadena hasta que un objeto ConcreteHandler asume la responsabilidad de manejarla

Consecuencias

El patrón Chain of Responsability tiene los siguientes beneficios y desventajas:

1. Reduce el acoplamiento. El patrón libera a un objeto de tener que conocer qué objeto maneja una petición. El objeto sólo tiene que saber que la petición será manejada adecuadamente. Receptor y emisor no se conocen explícitamente, y un objeto de la cadena no conoce la estructura de la misma.
   Como resultado, el Chain of Responsability puede simplificar la interacción entre objetos. En vez de tener que mantener referencias a todos los candidatos receptores, cada uno mantiene una referencia al sucesor
2. Agrega flexibilidad al asignar responsabilidades a los objetos. El Chain of Responsibility agrega flexibilidad al distribuir las responsabilidades entre los objetos. Se puede agregar o cambiar responsabilidades de manejar una petición mediante la agregación o modificación de la cadena en tiempo de ejecución. Se puede combinar este aspecto con la subclasificación para especializar los manejadores estáticamente.
3. No se garantiza que exista un receptor. Como la petición no tiene un receptor explícitamente declarado, no se garantiza que la petición sea gestionada (podría alcanzar el final de la cadena sin que ningún objeto la atienda). Una petición podría quedar sin gestionar si la cadena no está adecuadamente configurada.