



Svigruppo

Monga

Separation of concern

Esempio

AOP

AspectJ

Esempio

Sviluppo software in gruppi di lavoro complessi¹

Mattia Monga

Dip. di Informatica
Università degli Studi di Milano, Italia
mattia.monga@unimi.it

Anno accademico 2017/18, I semestre



Svigruppo

Monga

Separation of
concern

Esempio

AOP

AspectJ

Esempio

Lezione VII: *Separation of concerns* e divisione del lavoro



Svigruppo

Monga

Separation of
concern

Esempio

AOP

AspectJ

Esempio

Soluzione di un problema:

- Scomposizione in elementi piú semplici (**analisi**)
- Ricomposizione della soluzione finale (**sintesi**)

Secondo precetto: dividere ciascuna delle difficoltà da esaminare nelle parti di cui è suscettibile e di cui c'è bisogno per meglio risolverla. [Cartesio, "Discorso sul Metodo"]



Svigruppo

Monga

Separation of
concern

Esempio

AOP

AspectJ

Esempio

Nello sviluppo del software il principio viene applicato in tutte le fasi:

- nell'analisi dei requisiti, nel progetto, nella realizzazione, nella convalida, nella messa in opera.
- Fondamentale per la suddivisione del lavoro e delle responsabilità

Ogni persona coinvolta a qualche titolo nel sistema (*stakeholder*: dagli investitori fino agli utenti finali) ragiona attraverso l'uso di una opportuna astrazione, secondo i propri obiettivi.

- Tutt'altro che indipendenti (*ortogonali*)
- Non sempre aggregabili gerarchicamente
- Sovrapposizioni e incongruenze



Svigruppo

Monga

Separation of concern

Esempio

AOP

AspectJ

Esempio

Tradizionalmente:

- decomposizione secondo le funzionalità del sistema
- Nell'*object-oriented programming*: **incapsulamento**
 - metodologia *Class-Responsibility-Collaboration* (Beck & Cunningham, 1989)

Spaghetti programming?



Svigruppo

Monga

Separation of concern

Esempio

AOP

AspectJ

Esempio

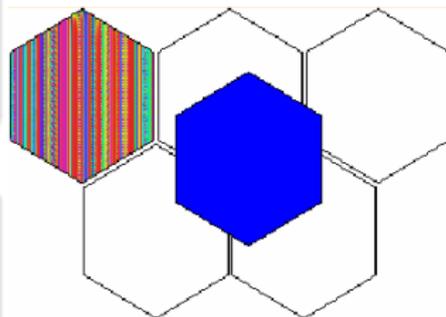
Dal punto di vista del programmatore ci sono due fenomeni:

Code Tangling

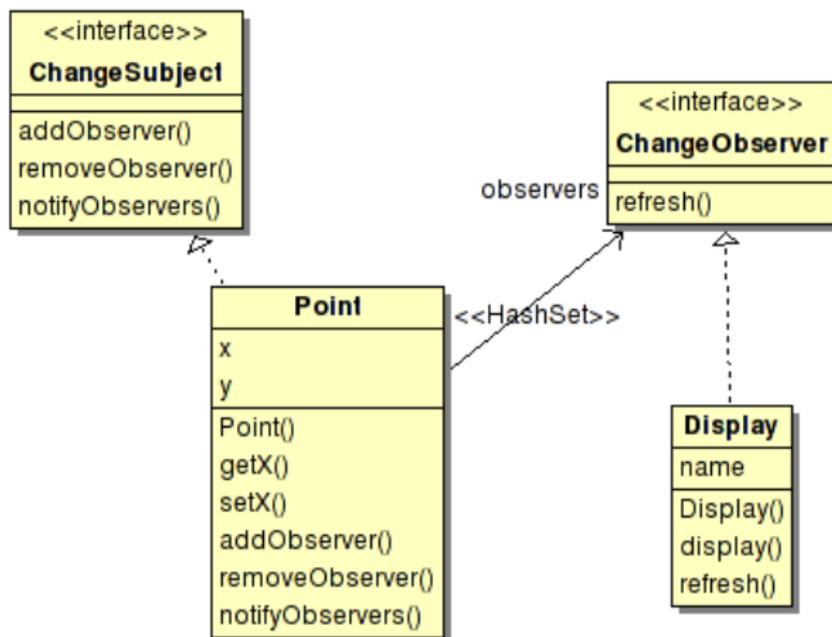
Un componente contiene codice che si riferisce a diverse problematiche.

Code Scattering

Il codice che si riferisce ad una problematica risulta disperso fra più componenti.



Esempio



Swigruppo

Monga

Separation of concern

Esempio

AOP

AspectJ

Esempio



```
public class Point implements ChangeSubject {
    private HashSet<ChangeObserver> observers;
    private int x, y;
    public Point(int x, int y) { this.x = x; this.y = y;
        this.observers = new HashSet<ChangeObserver>();
    }
    public int getX() { return x; }
    public void setX(int x) { this.x = x;
        notifyObservers();
    }
    public void addObserver(ChangeObserver o) { this.observers.add(o); }
    public void removeObserver(ChangeObserver o) { this.observers.remove(o); }
    public void notifyObservers() { for (ChangeObserver o : this.observers)
        { o.refresh(this); }}
```



Svigruppo

Monga

Separation of
concern

Esempio

AOP

AspectJ

Esempio

Le problematiche non facilmente incapsulabili in moduli classici vengono dette **aspetti** [Kiczales97] e **programmazione orientata agli aspetti** il filone di ricerca che cerca di identificare i meccanismi linguistici idonei ad isolare il piú possibile tali aspetti

L'idea dell'AOP



Svigruppo

Monga

Separation of concern

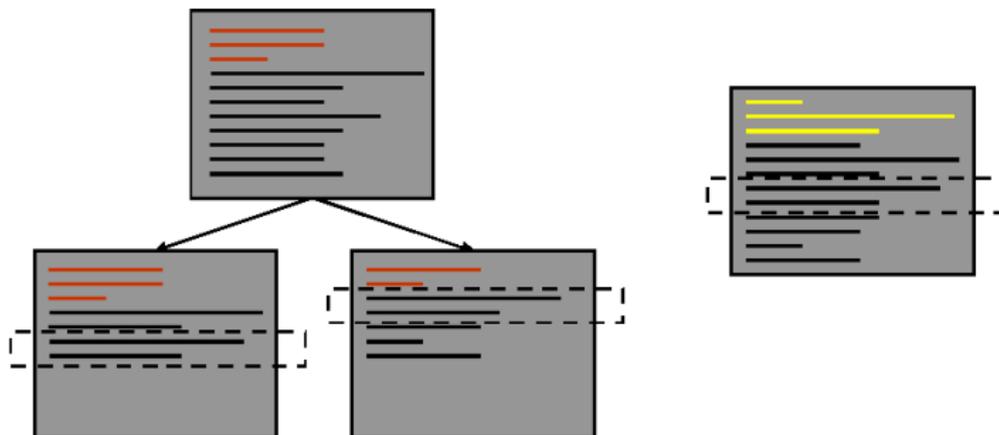
Esempio

AOP

AspectJ

Esempio





L'idea dell'AOP



Svigruppo

Monga

Separation of concern

Esempio

AOP

AspectJ

Esempio



L'idea dell'AOP



Svigruppo

Monga

Separation of concern

Esempio

AOP

AspectJ

Esempio





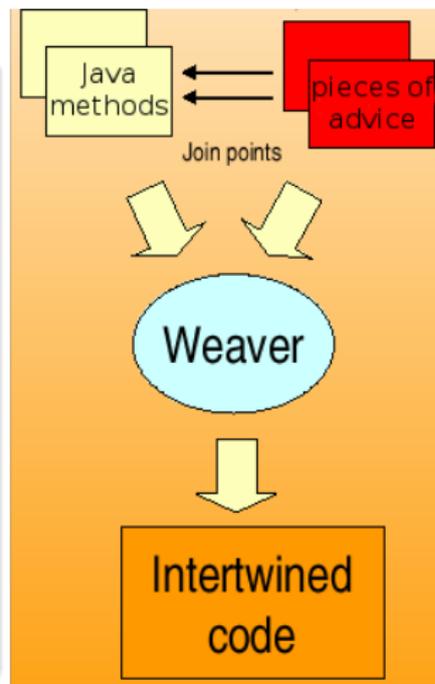
AspectJ [Xerox98] è un linguaggio orientato agli aspetti di cui Java è un sotto-insieme.

AspectJ

- 1 costrutti speciali per isolare gli aspetti (**aspects**)
- 2 costrutti per identificare i punti di integrazione (**joinpoints**)
- 3 Aspetti vengono poi **intrecciati** da un motore opportuno (**weaving**)

AspectJ

- 1 Incapsulamento: *aspect*
- 2 *Pieces of advice*: porzioni (anonime) di codice eseguite *before*, *after*, o *around* i *join-points*
- 3 *pointcut*: insiemi di *join-points*
- 4 È possibile variare la struttura del codice
- 5 Il **weaving** avviene a *compile-time*



Svigruppo

Monga

Separation of concern

Esempio

AOP

AspectJ

Esempio



```
public class Point {
    private int x, y;
    public Point(int x, int y) {this.x=x; this.y=y}
    public int getX() {return x; }
    public void setX(int x) {this.x = x; }
}
```

```
public class Display {
    private String name;
    public Display(String s) {this.name = s; }
    public void display (String s) {
        System.out.println(name + ": " + s);
    }
}
```



```
public aspect CoordinateObserver extends ObserverProtocol{
    declare parents: Point implements ChangeSubject;
    declare parents: Display implements ChangeObserver;
    protected pointcut subjectChange(ChangeSubject s):
        (call(void Point.setX(int)) ||
         call(void Point.setY(int)) ) && target(s);
    protected void refresh(ChangeSubject s, ChangeObserver o) {
        ((Display)o).display("Screen updated ");
    }
    after(Display d): execution(Display.new(*)) && this(d){
        this.addObserver((ChangeObserver)d);
    }
}
```

Observer protocol



Svigruppo

Monga

Separation of concern

Esempio

AOP

AspectJ

Esempio

```
import java.util.HashSet;

public abstract aspect ObserverProtocol {
    protected interface ChangeSubject { }
    protected interface ChangeObserver { }
    private HashSet<ChangeObserver> observers = new HashSet();
    public void addObserver(ChangeObserver o) {
        observers.add(o);
    }
    protected abstract pointcut subjectChange(ChangeSubject s);
    after(ChangeSubject ss): subjectChange(ss) {
        for (ChangeObserver o : observers){
            refresh(ss, o);
        }
    }
    protected abstract void refresh(ChangeSubject s, ChangeObserver o);
}
```



Svigruppo

Monga

Separation of
concern

Esempio

AOP

AspectJ

Esempio

```
public class Main{
    public static final void main(final String[] args) {
        Point p = new Point(2,3);
        Display d = new Display("Test");
        p.addObserver(d);
        d.display("...");
        p.setX(2222);
    }
}
```



Pointcut language

Nei linguaggi orientati agli aspetti è importante capire il *join-point model*: quali **eventi computazionali** possono essere intercettati?

- Chiamate ed esecuzioni di metodi
- Lettura e scrittura di campi
- *Handling* di eccezioni

Si possono filtrare rispetto a proprietà statiche e dinamiche

- “*within*” una classe o un metodo
- nel *control flow* di un altro (insieme di) *join-point*
- se una condizione (a *run-time*) è vera

Svigruppo

Monga

Separation of concern

Esempio

AOP

AspectJ

Esempio



Svigruppo

Monga

Separation of concern

Esempio

AOP

AspectJ

Esempio

Che informazioni sono disponibili in un *join point*?

- In quale oggetto è il *join point*? (`this`)
- Che oggetto è coinvolto nel *join point* (`target`)
- Quali argomenti ha il *join-point*? (`args`)
- *Reflection* con `thisJoinPoint`



Svigruppo

Monga

Separation of
concern

Esempio

AOP

AspectJ

Esempio

```
pointcut setter():  
target(Point)&&(call(void setX(int))||call(void setY(int)));
```

```
pointcut setter(Point p):  
target(p)&&(call(void setX(int))||call(void setY(int)));
```

```
pointcut testEquality(Point p1,Point p2):  
target(p1)&&args(p2)&&call(boolean equals(Object));
```



Svigruppo

Monga

Separation of concern

Esempio

AOP

AspectJ

Esempio

AOP secondo Filman

Quantification Wildcard (lessicali, sintattici, semantici)

Obliviousness Il programmatore della parte decomposta
funzionalmente non sa nulla della parte aop

Domanda: quanto è “reale” la separazione?