

Design pattern – V

Stefano Mizzaro

Dipartimento di matematica e informatica
Università di Udine
<http://www.dimi.uniud.it/~mizzaro>
mizzaro@dimi.uniud.it
PAOO, Lezione 13
3/5/2004

© S. Mizzaro - Design pattern - 5

3

Riassunto

- I 7 pattern strutturali: Adapter, Façade, Composite, Decorator, Bridge, Proxy, Flyweight
- I 5 pattern creazionali: Factory Method, Abstract Factory, Singleton, Prototype, Builder
- I primi 4 pattern comportamentali: Template Method, Strategy, State, Command

© S. Mizzaro - Design pattern - 5

2

Scaletta

- Seminario
 - eXtreme Programming, di Gianluca Demartini
- Un altro pattern comportamentale
 - Observer

© S. Mizzaro - Design pattern - 5

3

Gli 11 pattern comportamentali

13. Template Method (Metodo sagoma)
14. Strategy (Strategia)
15. State (Stato)
16. Command (Comando)
17. Observer (Osservatore)
18. Mediator (Mediatore)
19. Memento (Memento, Ricordo)
20. Iterator (Iteratore)
21. Visitor (Visitatore)
22. Chain of Responsibility (Catena di responsabilità)
23. Interpreter (Interprete)

© S. Mizzaro - Design pattern - 5

4

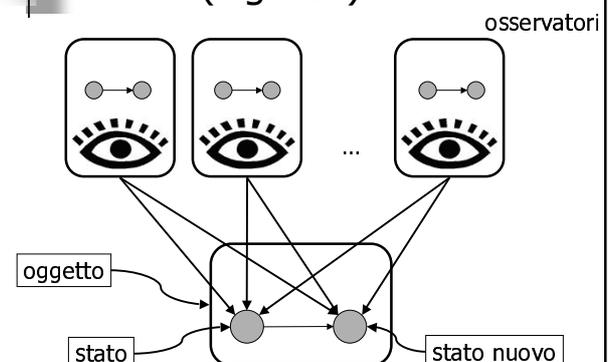
17. Observer (Osservatore)

- Scopo
 - Avere oggetti che "osservano" lo stato di un oggetto...
 - ... e sono notificati e aggiornati immediatamente quando lo stato cambia
- È un brutto nome (l'osservatore non osserva, rimane in attesa che gli venga detto di andare a guardare...)

© S. Mizzaro - Design pattern - 5

5

L'idea? (ingenua)



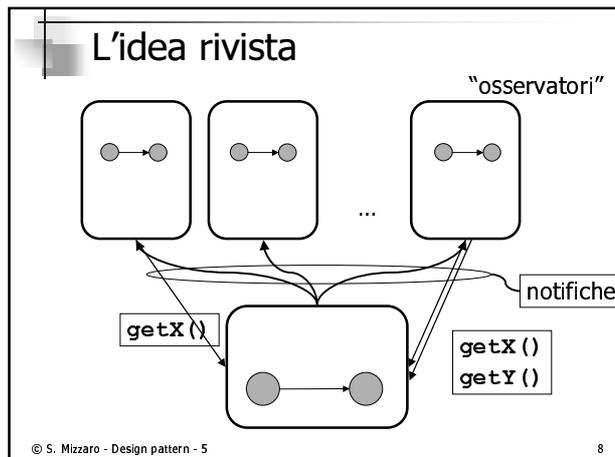
© S. Mizzaro - Design pattern - 5

6

Uhm...

- Attributi pubblici? Scartato!
- Gli osservatori invocano continuamente un metodo **cambiato** dell'oggetto osservato?
- Problemi! L'osservatore potrebbe:
 - "rompere" troppo all'osservato
 - scoprire la variazione troppo tardi
 - "perdersi" la prima di due variazioni "veloci"
 - E se ho tanti osservatori? Poco scalabile!
 - L'osservato passa il tempo a rispondere (magari senza nessuna variazione)

© S. Mizzaro - Design pattern - 5 7



Quindi

- (prerequisito: Gli osservatori si registrano presso l'oggetto osservato)
- Quando l'oggetto osservato cambia stato, notifica tutti gli osservatori (ne invoca un metodo)
- Quando notificato, ogni osservatore decide cosa fare
 - Niente
 - Richiedere all'osservato informazioni sullo stato
 - ...

© S. Mizzaro - Design pattern - 5 9

Quindi / 2

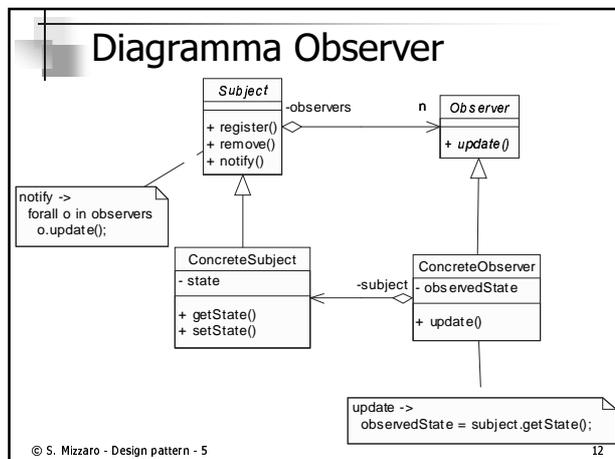
- Gli osservatori potrebbero essere varie "viste" sullo stesso oggetto
 - Dati di un foglio elettronico e loro rappresentazioni tabellari, grafiche, ...
- L'oggetto osservato non conosce l'identità precisa degli osservatori
 - Altrimenti, "cambiare trasmettitore radio ogni volta che un'autoradio entra in città..."
- Gli osservatori si aggiungono/tolgono al runtime

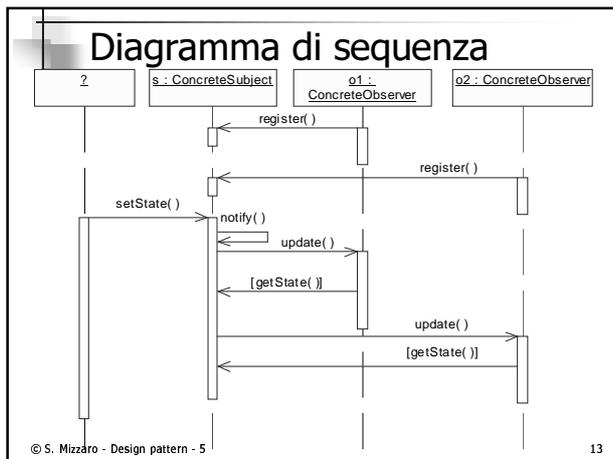
© S. Mizzaro - Design pattern - 5 10

Il pattern

- Gli osservatori devono potersi registrare
 - L'oggetto osservato deve fornire un'interfaccia standard per la registrazione
 - **register()** e **remove()**
- L'oggetto osservato deve poter notificare
 - Gli osservatori devono fornire un'interfaccia standard per la notifica
 - **update()** (e l'oggetto osservato ha **notify()** che chiama tutti gli update dei registrati)

© S. Mizzaro - Design pattern - 5 11





Commenti

- Chi invoca **setState()**?
 - Chiunque
 - Uno degli osservatori
- Chi invoca **notify()**?
 - Ogni metodo dell'oggetto osservato che modifica lo stato
 - Il cliente, dopo che ha terminato una sequenza di modifiche

© S. Mizzaro - Design pattern - 5

14

Varianti

- E se un osservatore osserva più oggetti, come fa a sapere chi ha variato lo stato?
 - Parametro di `update()` (ad es., `this`)
- Push vs. Pull
 - Pull: `update()` non passa nessuna info sullo stato, l'osservatore usa `getState()`
 - Push: `update()` passa anche lo stato
- ...

© S. Mizzaro - Design pattern - 5

15

Observer in Java

- In `java.util`
- Interfaccia **Observer**
 - `void update(Observable o, Object arg)`
- Classe **Observable**
 - `addObserver(Observer o)`
 - `deleteObserver(Observer o)`
 - `notifyObservers(Observer o)`
- Nomi di metodi differenti

© S. Mizzaro - Design pattern - 5

16

Riassunto

- Seminario
 - eXtreme Programming, di Gianluca Demartini
- Un altro pattern comportamentale
 - Observer

© S. Mizzaro - Design pattern - 5

17