

PA1459/PA1460 Example

Mikael Svahnberg

March 1, 2021

Contents

1	Systembeskrivning	1
2	Klassdiagram	2
3	Beskrivning av Klassdiagrammet	3
3.1	Storage	3
4	Pseudokod	3
4.1	Observable::addObserver()	3
4.2	Observable::notify()	4
4.3	ContentModel::addContent()	4
4.4	t.ex. ComicsSearcher::notify()	4
5	Diskussion om GRASP-mönster	4
6	Användning av GRASP-mönster	4
	file:///Users/msv/Documents/Teaching/PA1415_software_design/Material/ 202102-HomeExam-Example-I.org pattern Observer Pattern GRASP1 Information Expert GRASP2 Con- troller	

1 Systembeskrivning

Ett system som letar efter nyckelord på en social mediaplattform (t.ex. ett diskussionsforum), och när vissa nyckelord hittas så skall bestämda handlingar utföras. Observer-mönstret används för att sära på letandet av nyckelord och agerandet utifrån dessa nyckelord.

2 Klassdiagram

```
package Scraper {
  ' not done here -- too big and not part of the task or the Observer pattern
}

package Storage {

  class ContentModel {
    +addContent(String newContent)
  }

  class ContentAtom

  ContentModel -- "*" ContentAtom
}

Scraper -> Storage : insert >

package ObserverPattern {
  class Observable {
    -List<Observer*> myObservers
    +addObserver(Observer* newObserver)
    -notify()
  }

  abstract class Observer {
    +notify(Observable* source, String newContent)
  }

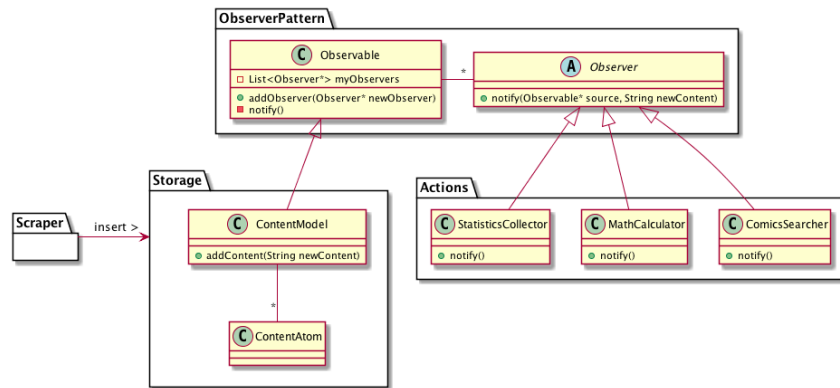
  Observable - "*" Observer
}

package Actions {
  Observer <|-- StatisticsCollector
  Observer <|-- MathCalculator
  Observer <|-- ComicsSearcher

  StatisticsCollector : +notify()
  MathCalculator : +notify()
}
```

```
ComicsSearcher : +notify()
}
```

```
Observable <|-- ContentModel
```



3 Beskrivning av Klassdiagrammet

Klassdiagrammet har ett antal paket:

Scraper samlar data från websidor och skickar till **Storage**.

Storage skapar ContentAtoms av nytt innehåll och lagrar dessa. Meddelar sedan **Actions** via sitt Observer pattern att det finns nytt innehåll.

Actions Innehåller olika sätt att reagera på innehåll.

ObserverPattern De klasser som behövs för ett generiskt Observer pattern

3.1 Storage

ContentModel

ContentAtom

4 Pseudokod

4.1 Observable::addObserver()

```
void Observable::addObserver(Observer* newObserver) {
    myObservers.append(newObserver);
}
```

4.2 Observable::notify()

```
void Observable::notify() {
    myObservers.forEach( function(o) {
        o.notify(this, newContent); // newContent is magically available
    });
}
```

4.3 ContentModel::addContent()

```
void ContentModel::addContent(String newContent) {
    ContentAtom atom = new ContentAtom(newContent);
    DBHandler::store(atom);
    this->notify(newContent); // Här använder vi Observer-mönstret
}
```

4.4 t.ex. ComicsSearcher::notify()

```
void ComicsSearcher::notify(Observable* source, String newContent) {
    String key = newContent.split()[0];
    if(myKeywords.find(key)) {
        // Do relevant action based on keyword
    }
}
```

5 Diskussion om GRASP-mönster

Observable är *information expert* på vilka observers som finns. Den är också en *controller* som delegerar ut ansvar till var och en av sina observers för att genomföra sin handling.

Observer (eller sub-klasserna till observer) är information expert på vilka nyckelord de skall reagera på. De är också information experts på att genomföra vad de nu skall göra.

Var och en av klasserna i **Actions** kan vara en controller för att dirigera det arbetsflöde som behövs för att lösa sin uppgift.

6 Användning av GRASP-mönster

- ContentModel är en Information Expert på hur nytt innehåll skall lagras. Den är också (via Observable) är den också information expert

på vilka som skall meddelas när det finns nytt innehåll.

- ContentAtom är information expert på en specifik bit av innehåll.
- Observable och Observer – se tidigare.
- [StatisticsCollector, MathCalculator, ComicsSearcher] är information expert på hur respektive uppgift skall utföras, och vilka nyckelord som skall sätta igång handlingen.
- [StatisticsCollector, MathCalculator, ComicsSearcher] kan vara controllers för att dirigera ett större arbetsflöde för att lösa sin respektive uppgift.