CS 635 Advanced Object-Oriented Design & Programming
Spring Semester, 2005
Doc 11 More Observer
Contents

References

Refactoring to Patterns, Joshua Kerievsky, 2005, pp 236-246

Observer Issues
What Methods/Classes are used Here?

```
public class Subject {
    Window display;
    public void someMethod() {
        this.modifyMyStateSomeHow();
        display.addText( this.text() );
    }
}
```

How hard will it be to follow the flow of control?

What Methods/Classes are used Here?

```
public class Subject {
   ArrayList observers = new ArrayList();

   public void someMethod() {
      this.modifyMyStateSomeHow();
      changed();
   }

   private void changed() {
      Iterator needsUpdate = observers.iterator();
      while (needsUpdate.hasNext() )
         needsUpdate.next().update( this );
   }
}
```

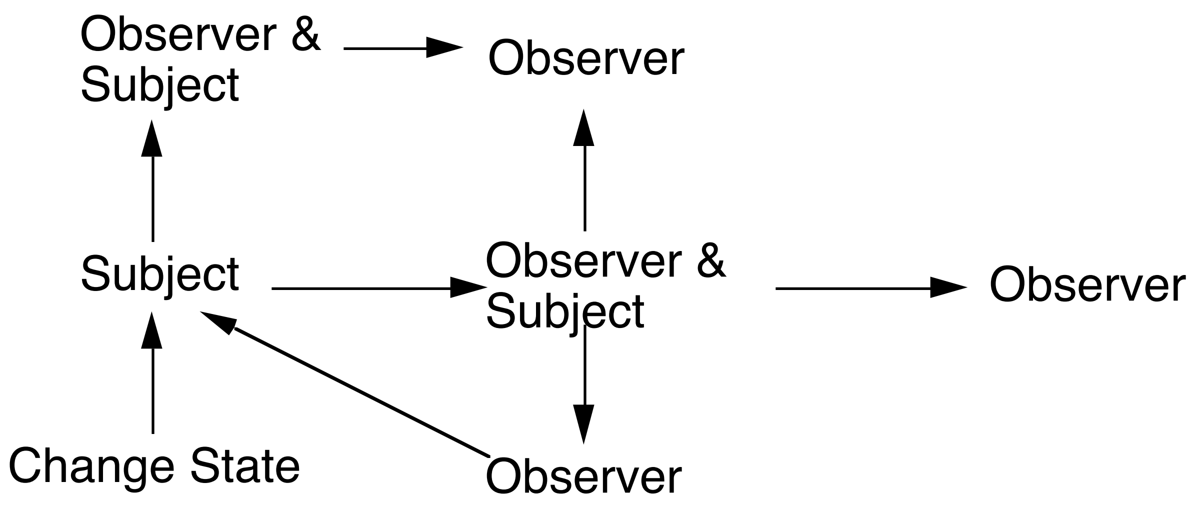
How hard will it be to follow the flow of control?

More Liabilities of Observer Pattern

More complex than hard-coded notification

Memory leaks if observers are not removed from subject

Cascading notifications are hard to follow



Words of Advice

The observer pattern is used often. Because it isn't difficult to implement, you may be tempted to use this pattern before it's actually needed. Resist that temptation.

Joshua Kerievsky

Small Subjects
ValueModel with Java-like Syntax

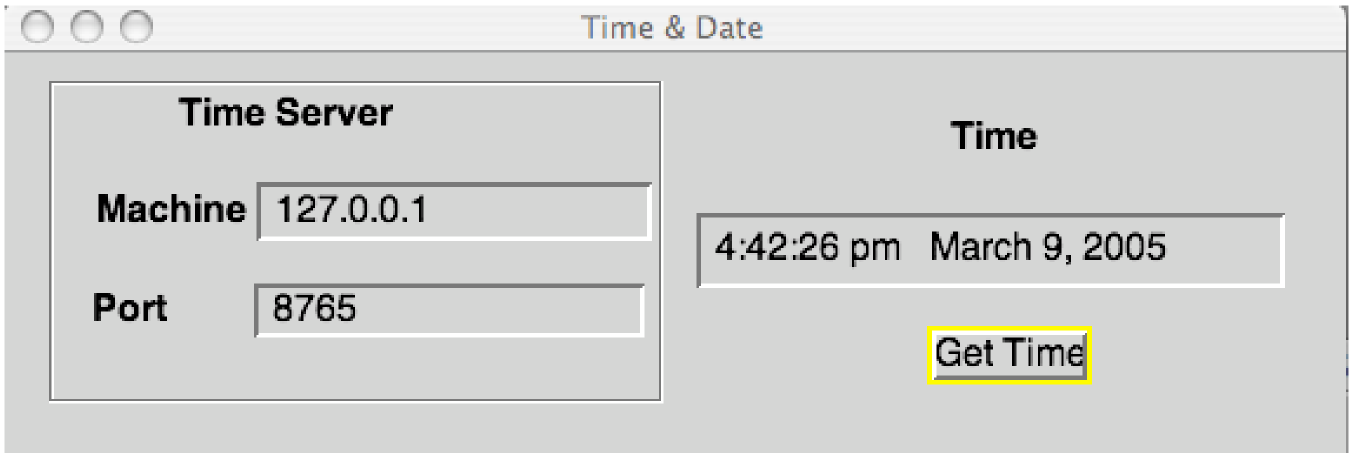
```
public class ValueModel {
   ArrayList observers = new ArrayList<Observer>();
   Object subject;

   public Object add(Observer newObserver) {
      observers.add(newObserver);
   }

   private void changed() {
      for (Observer each : observers)
         each.update();
   }

   public void value(Object newSubject) {
      subject = newSubject;
      this.changed();
   }

   public Object value() {
      return subject;
   }
}
```

Now a String can be a subject

ValueModel provides Interface for Observers

GUI widgets tend to observer:

- Strings
- Numbers
- Lists



127.0.0.1

8765

4:42:26 pm March 9, 2005

Generic GUI widgets can observe value models

What if application does not want to use ValueModels?

```
public class TimeDateClient
   {
   String server;
   int serverPort;

   public TimeDateClient(String serverNameOrIP, int port)
      {
      server = serverNameOrIP;
      serverPort = port;
      }

   public String date() throws IOException
      {
      return sendMessage("date");
      }

   public String time() throws IOException
      {
      return sendMessage("time");
      }
```