

3815ICT-Software Engineering Workshop 7

Activity 1

Study the entry in Wikipedia for the “Command pattern” (en.wikipedia.org/wiki/Command_pattern).

Review the following piece of code. Explain why this represents an anti-pattern:

```
if (op==1) {  
    User u = new User( request.getAttribute("user"), request.getAttribute("pass"));  
} else if (op==2) {  
    ...  
}  
...
```

What are the disadvantages of the template suggested by the anti-pattern above?

Response 1

The command design pattern is intended to create encapsulated environments in which a user defines a list of commands which may be executed at a later time. The reason the above code represents an anti-pattern in that respect is because a command should be a straightforward task to accomplish and should not rely on any outside factor. The use of conditionals signals an anti-pattern for command design as the command should be invoked with conditions outside of the command itself. The command should simply accomplish the given task.

Activity 2

Examine the code below that defines an abstract class. What part of the Command pattern does this achieve? What does the constructor achieve?

```
//Action.java  
  
public abstract class Action  
{  
    protected Model model;  
    public Action(Model model) {  
        this.model = model;  
    }  
    public abstract String getName();  
    public abstract Object perform(HttpServletRequest req);  
}
```

Response 2

This abstract class functions as the receiver of the command pattern. The constructor instantiates the instance of the class with a model variable in the form of a Model object.

Activity 3

Examine the code below that defines a concrete class. What part of the Command pattern does this achieve? What does the constructor achieve? Explain the method getName().

// CreateUserAction.java:

```
public class CreateUserAction extends Action {  
    public CreateUserAction(Model model) {  
        super(model);  
    }  
    public String getName() {  
        return "createUser";  
    }  
    public Object perform(HttpServletRequest req) {  
        return model.createUser(req.getAttribute("user"), req.getAttribute("pass"));  
    }  
}
```

Response 3

In terms of the command, receiver, invoker and client, this piece of code refers to the commands as these commands are to be run. The constructor here simply refers to the superclass of Action and imparts the model behaviours of the superclass. The getName() method is called to return a string which is 'createUser'.

Activity 4

Examine the code below that defines a concrete class. What part of the Command pattern does this achieve? Is this linked to some part of MVC?

```
public class ControllerServlet extends HttpServlet {  
    private HashMap actions;  
    public void init() throws ServletException {  
        actions = new HashMap();  
        CreateUserAction que = new CreateUserAction(model);  
        actions.put(que.getName(), que);  
    }  
}
```

```

        //... create and add more actions
    }

    public void doPost(HttpServletRequest req, HttpServletResponse resp) throws IOException,
    ServletException {

        // First identify operation "op" from URL.
        // method getOperation() is defined elsewhere.
        String op = getOperation(req.getRequestURL());
        // Then find and execute corresponding Action
        Action action = (Action)actions.get(op);
        Object result = null;
        try {
            result = action.perform(req);
        }
        catch (NullPointerException npx) {
            ...
        }

        //determine next view result to determine next view
    }..

```

Response 4

The above piece of code defines a controller in the MVC framework. In terms of its use in the command pattern, this code refers to the invoker as it is the program which invokes the commands.

Activity 5

Study the short entry in Wikipedia for the “Stateless protocol” (en.wikipedia.org/wiki/Stateless_protocol).

Research the concept of session and discuss what are the implications that the HTTP protocol of the Web is stateless with respect to sessions?

Response 5

A session is essentially a browsing session in which the user opens a ‘session’ with the client/server that he/she is connected to. A stateless protocol will not keep information about current state. When a request is made to the server, the server does not remember any information about the transaction by default. That is a stateless protocol. For the server to retain memory of the transaction which it may use for future transactions, the session

must not be stateless. This can be accomplished with various methods, ie. Use of cookies to retain memory.

Activity 6

Frameworks for Web architectures are very varied. Even for Java Server pages there are several variants. Browse over the chapter “The Spring Web MVC Framework” (available at Learning@GU).

Identify 4 advantages that this chapter attributes to the Spring Web MVC Framework.

Response 6

1. It is easier to test while using the Spring MVC Framework. Due to JavaBeans creation, user is able to inject test data using methods of the class.
2. Separation of roles between components that make up the web framework.
3. The controllers in Spring MVC framework are adaptable and do not require a html form.
4. MVC does not require business classes to extend a special class. Business objects are as such, reusable and binding is made easier.

Activity 7

Attempt to complete the tutorial named “Introduction to Spring Web MVC” (netbeans.org/kb/docs/web/quickstart-webapps-spring.html)

This tutorial can be followed with almost no background on Java Server Pages (JSP) or java beans. Although learning about such technologies might be useful. The point of completing such exercise is for you to experience a Web architecture based around Model View Controller (MVC). In particular, several of the components are configured by setting properties in XML-files, for example.

I recommend you follow this tutorial using NetBeans. You will probably find that you should install some plug-ins so you can complete the tutorial. If you are trying to find the category "java Web" to create a project of type "Web Application" you probably are missing the “java EE Base”, and if you find that after creating some project you are missing some libraries (a message about resources missing), you probably need the "EJB and EAR" plug-in.

Similarly, as you develop the project, you will need a WEB server that support Java beans. Chose the GlassFish server, and if it is not installed, NetBeans will fetch it and install it for you.

The entire project will be made available for you in Learning@GU, but it is important you attempt to follow the tutorial yourself. Also, you should introduce debugging print statements so you can follow some of what is going on. The challenge is, how would you represent the behavior with an UML sequence diagram?

Response 7

Activity 8

Write 15 lines of a reflective report on the previous activities. Analyse and evaluate the match of the activities to the learning objectives proposed in this workshop/laboratory.

Response 8

The first learning objective was to review software patterns in the context of web architecture. This appeared to use the Command Patter as the primary software pattern to study. Aside from this, the MVC framework was used to demonstrate software patterns. All of the learning objectives revolved around discussion of issues within web architecture and to be fair, I don't really believe these have been met. I did not once consider the idea of web architecture while attempting these questions and I don't feel that I was pushed to identify particular challenges within web architecture. There was not nothing to be learned however, the command pattern exists and Spring MVC is a method of applying a Model View Controller Paradigm in a Java environment which seems useful. In terms of learning objectives however, I feel they were skewed towards web-architecture and different representations of it yet my main concern was with the concepts of Command pattern, MVC and how they relate to the specific problems at hand. To be honest I feel I did not learn a whole lot but I am now aware of the existence of these concepts at the very least and I may or may not be able to apply it to the assignment.