

View Source: Design Patterns in the Wild

Patterns in Struts + Code Clinic

Patterns for Today

- Singleton
- Model-View-Controller
- Timemap Code Clinic with Nick!

Singleton



Singleton

```
-instance: Singleton  
+getInstance(): Singleton  
-Singleton()
```

ServletContextSingleton



ServletContextSingleton

```
- singleton: ServletContextSingleton  
- servletContext: ServletContext  
+ getInstance(): ServletContextSingleton  
- ServletContextSingleton()  
+ getServletContext(): ServletContext  
+ setServletContext(context: ServletContext)
```

Double Checked Locking

```
public Singleton getInstance() {  
    if (instance == null) {  
        instance = new Singleton();  
    }  
    return instance;  
}
```

...or...

```
public synchronized Singleton  
    getInstance() { ... }
```

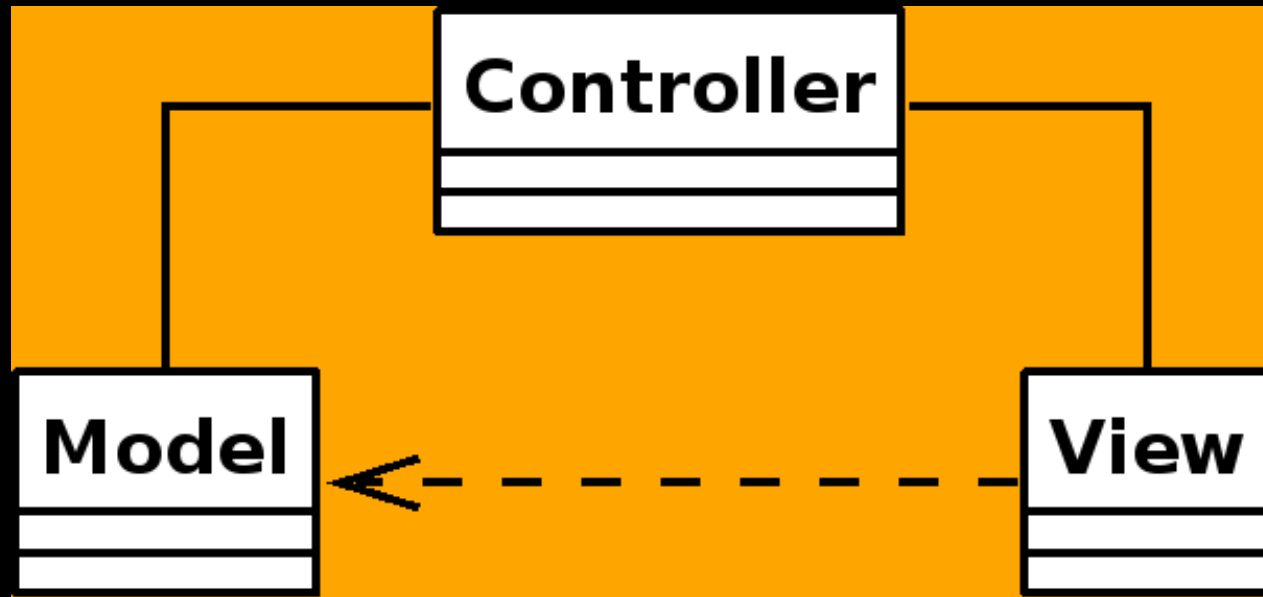
...or...

```
public Singleton getInstance() {  
    if (instance == null) {  
        synchronized {  
            if (instance == null)  
                instance = new Singleton();  
        }  
    }  
    return instance;  
}
```

Using Singleton

- Ensure a class has one, and only one, instance
- Provide a single global method for access
- Lazy load, or just-in-time initialization of instance, on demand
- Warnings!
 - Protect initialization against concurrent access, if necessary; remember DCL is broken in Java
 - Be wary of performance issues with locking
 - As a final option, control initialization explicitly

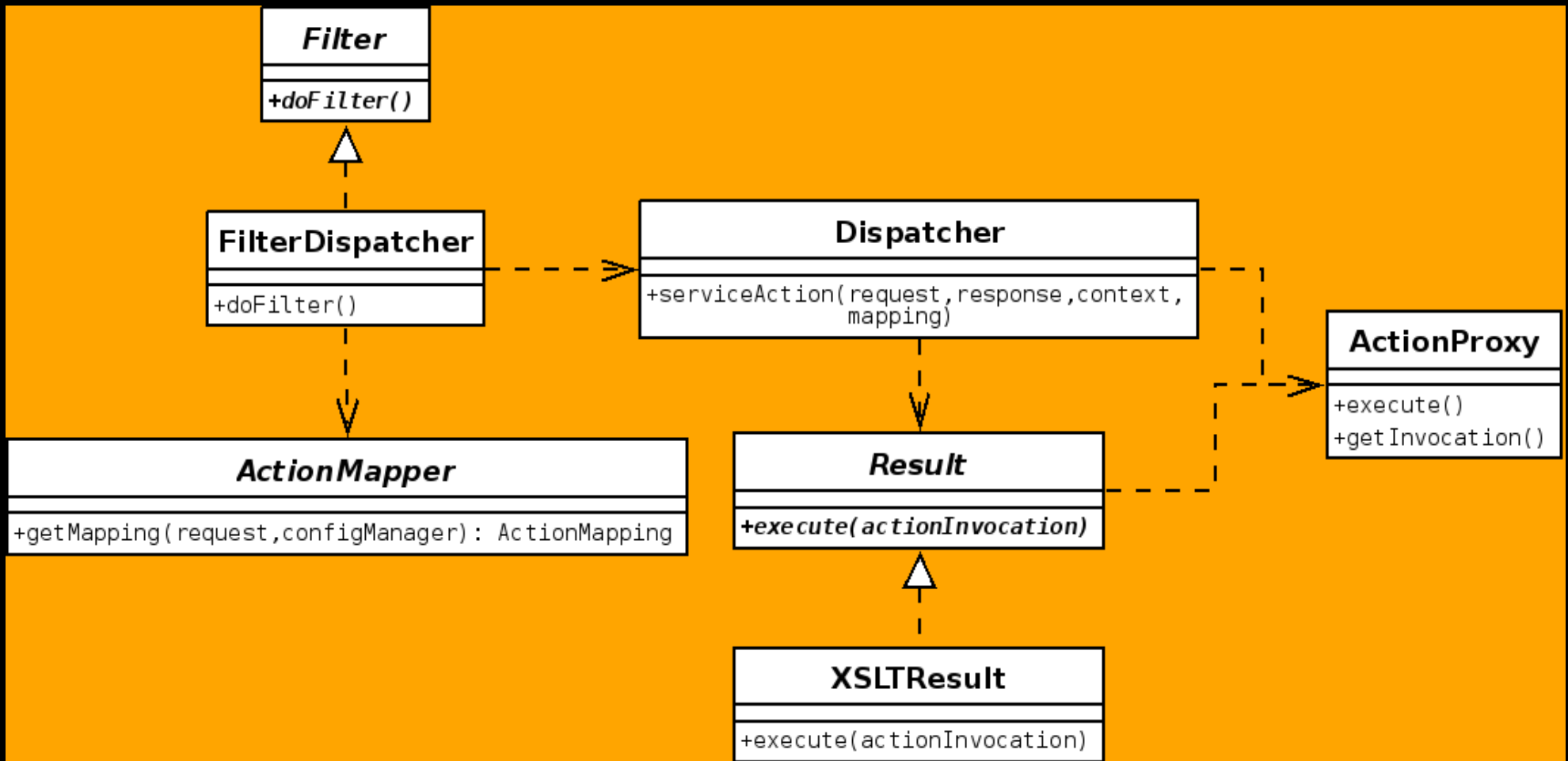
Model-View-Controller (MVC)



Struts MVC Configuration

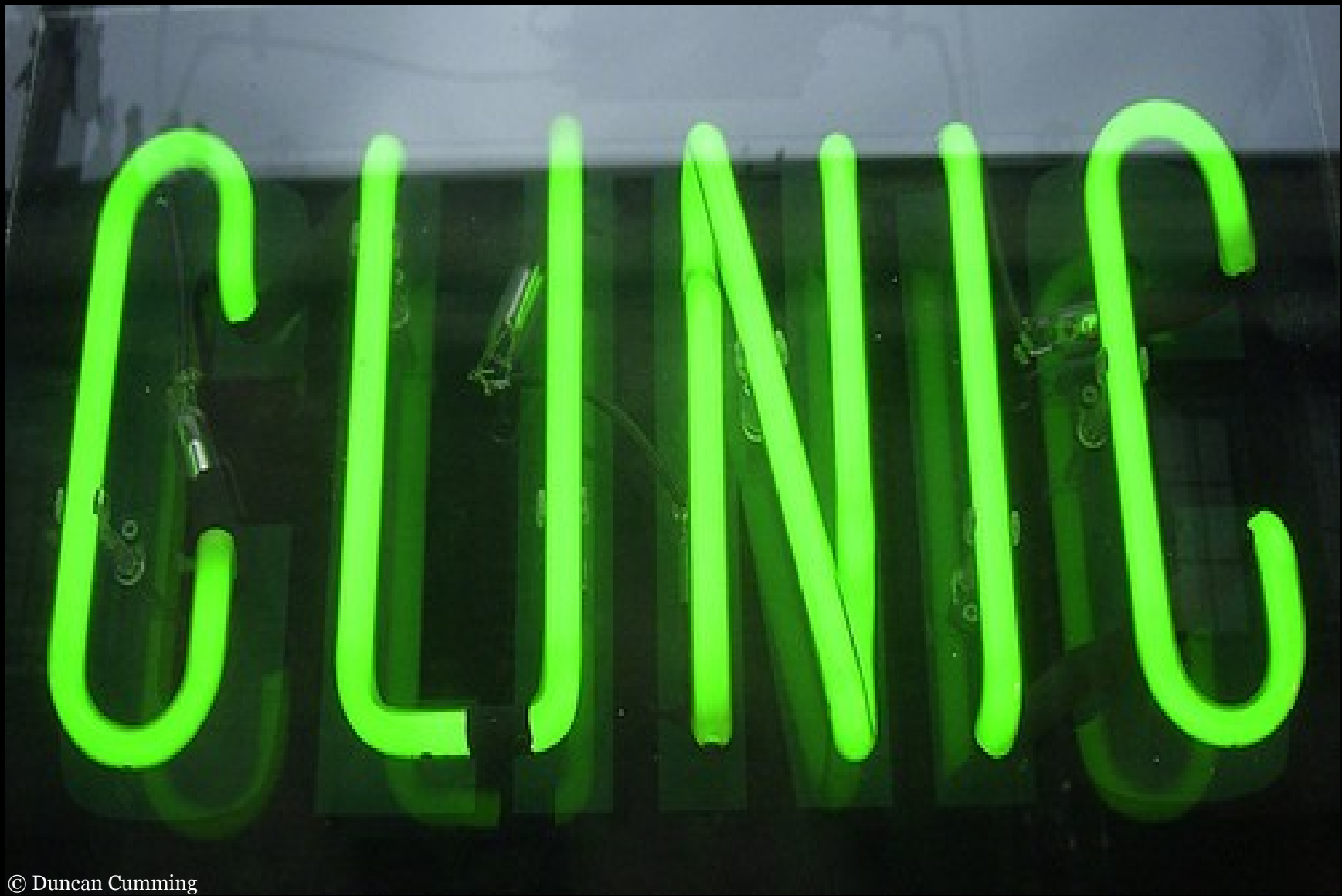
```
<action name="Login" class="LoginAction">  
  <result name="success">/Main.jsp</result>  
  <result name="failure">/Login.jsp</result>  
</action>
```


MVC in Struts



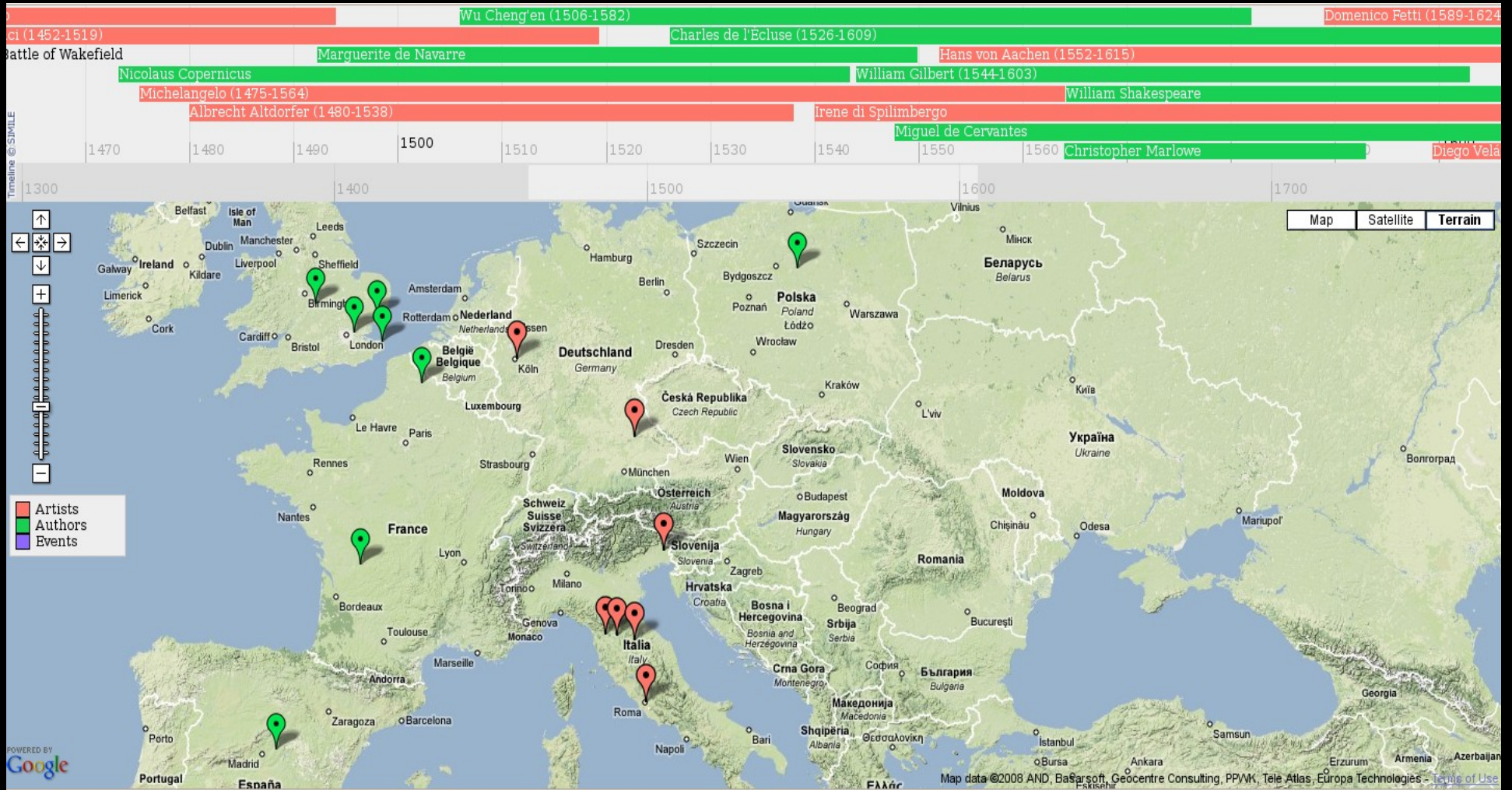
Using MVC

- Decouples application logic from presentation
- Controller: manages application behaviour
 - Maps user actions to model invocations
 - Selects view for response
- Model: encapsulates application logic and state
- View: Renders models

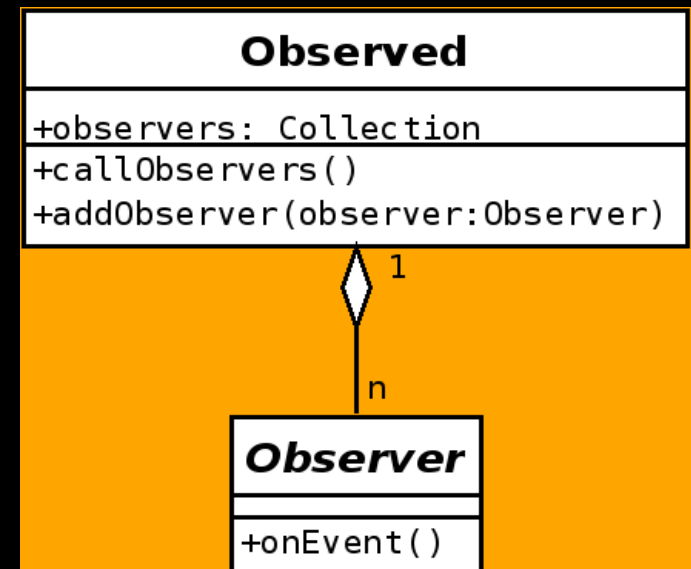
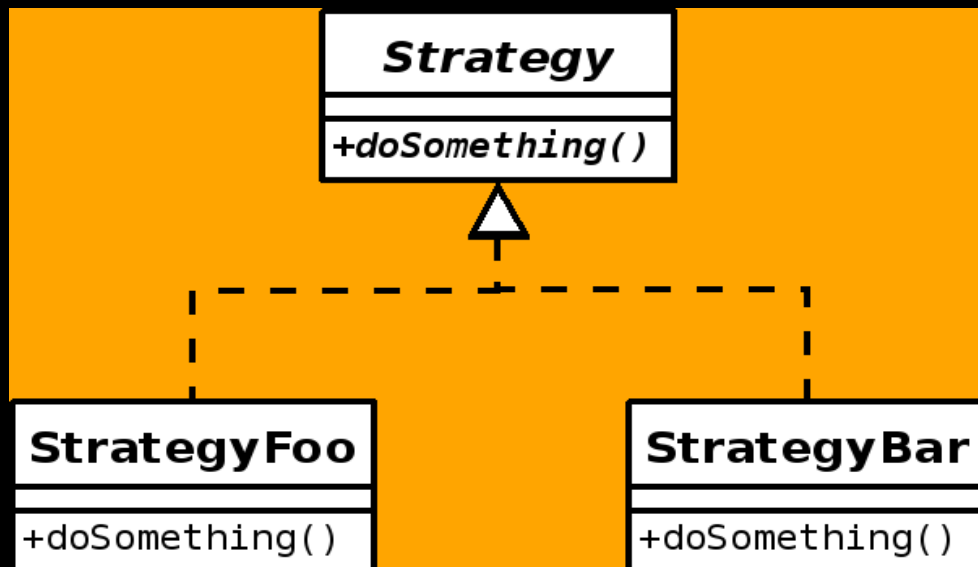
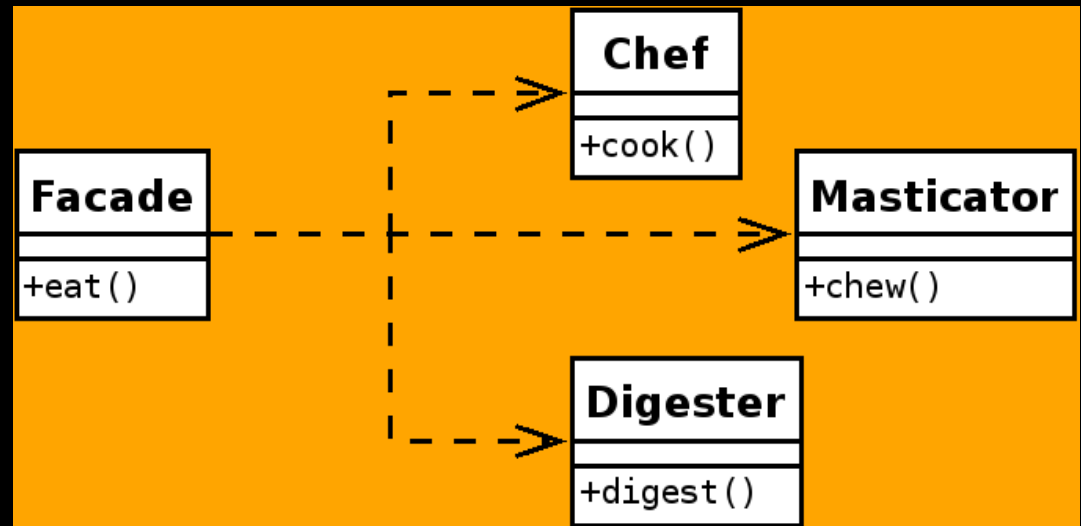
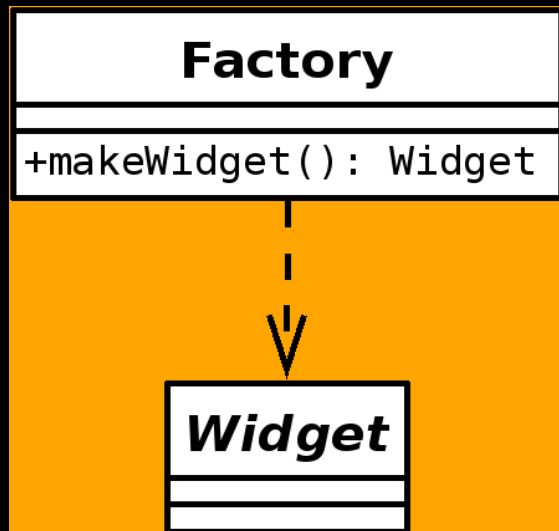


© Duncan Cumming

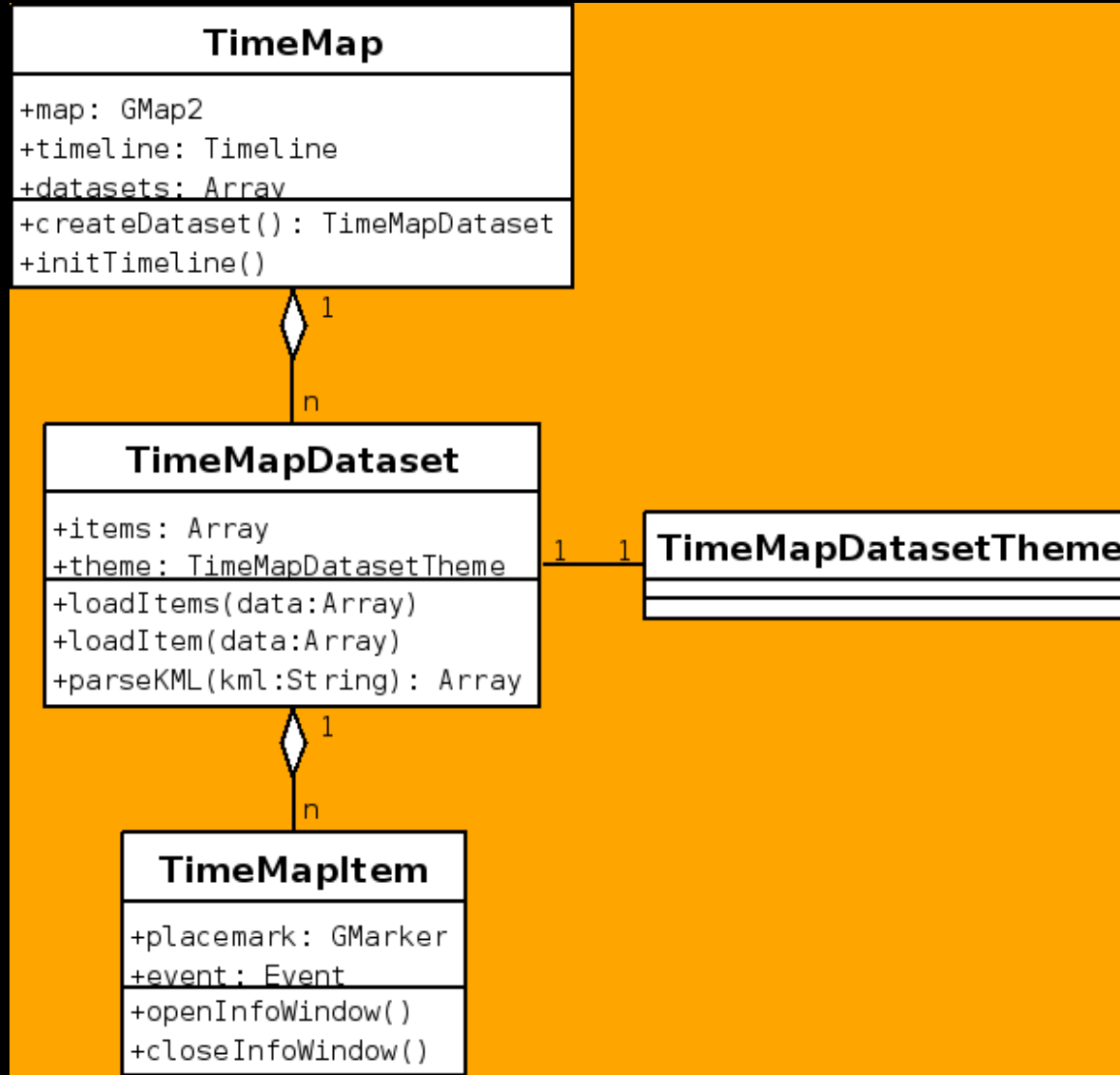
Introducing TimeMap



Patterns in TimeMap



Modeling TimeMap



Using Strategy in TimeMap

Current

```
TimeMapItem.openInfoWindow =  
  function() {  
    // omitting maximizing logic here  
    if (content is loaded) {  
      map.openInfoWindow(content)  
    } else {  
      load content with ajax from url  
      map.openInfoWindow(content)  
    }  
  }  
}
```

Using Strategy

```
loadedContentStrategy = function() {  
  map.openInfoWindow(this.content)  
}  
  
ajaxContentStrategy = function() {  
  load content with ajax from this.url  
  map.openInfoWindow(content)  
}  
  
TimeMapItem = function(strategy) {  
  //...  
  this.openInfoWindow = strategy;  
}
```