



Web Application DSL

Ali Ibrahim



Motivation

- J2EE and most web application frameworks do not check types or parameters between logic and presentation layers.
- Better understand web form semantics. Which computations may be performed twice? Which computations can be bookmarked? How do we store state for future computations?



Approach

- Extend JSP pages to allow for type checking of parameters for creating the page. Add parameter checking for invocation of computations, i.e. Check form parameters against expected parameters for the computation we are performing.
- Create an explicit notion of a web computation using a variation on continuations. Here I took some ideas from a paper by Krishnamurthi.



Why Extend Java?

- J2EE is a popular java based web application framework.
- Java has limited native reflection.
- Java has many tools which provide compile time introspection (such as Open Java) and run-time code generation.



Java Web Application(JSP)

```
<html>
<head> <title>Second parameter for addition</title> </head>
<body>
  <h2>Adding two numbers (second step)</h2>
  <h3>First number entered: ${requestScope.param1}</h3>
  <form action="/Add" method="get">
    <input type="hidden" name="param1" value="${requestScope.param1}"/>
    <table>
      <tr><th>Second addition parameter</th><td><input type="text" name="param2"
        size="5"/></td></tr>
      <tr><td><input type="submit" value="Add"/></td></tr>
    </table>
  </form>
</body>
</html>
```



Java Web Application (JSP*)

```
<!-- SecondParameterPage(String param1) -->
<html>
<head> <title>Second parameter for addition</title> </head>
<body>
  <h2>Adding two numbers (second step)</h2>
  <h3>First number entered: ${param1}</h3>
  <form action="/Add" method="get">
    <table>
      <tr><th>Second addition parameter</th><td><input type="text" name="param2"
        size="5"/></td></tr>
      <tr><td><input type="submit" value="Add"/></td></tr>
    </table>
  </form>
</body>
</html>
```



Parameter and Type Checking

- I used source to source translation to convert my modified JSP format to a JSP file and Page class which handle marshalling and unmarshalling parameters and ensure type correctness.



Web computations

- We need to keep track of state for future computations which is not required to generate the current view.
- We need to think about which computations we would like the user to be able to bookmark and which ones need some dynamic state.



Types of computations

- Static computations are independent of system or session state. These computations may be safely bookmarked by the user.
- Dynamic computations depend on some state. These computations may be restricted to running once per instance.



Example 2 Step Computation

```
public class SaveParameterComputation extends StaticComputation {  
  
    protected Page performComputation(int firstParameter) {  
        return new SecondParameterPage(firstParameter, new AddComputation(firstParameter));  
    }  
}  
  
public class AddComputation extends StaticComputation {  
    private int firstParameter;  
  
    public AddComputation(int firstParameter) {  
        this.firstParameter = firstParameter;  
    }  
  
    protected Page performComputation(int secondParameter) {  
        int result = firstParameter + secondParameter;  
        return new ResultPage(result);  
    }  
}
```



References

- Krishnamurthi, S. "The CONTINUE Server".
- Matthews, J. and Findler R. and Graunke, P. and Krishnamurthi, S. and Felleisen, M. "Automatically Restructuring Programs for the Web".
- Graunke, P. and Findler, R. and Krishnamurthi, S. and Felleisen, M. "Modeling Web Interactions".