# Slicing Java Server Pages Application

Madhusmita Sahu
*Department of MCA*
*C V Raman Computer Academy*
*Bhubaneswar-752054, India*
*madhu_sahu@yahoo.com*

Durga Prasad Mohapatra
*Department of CSE*
*National Institute of Technology*
*Rourkela-769008, India*
*durga@nitrkl.ac.in*

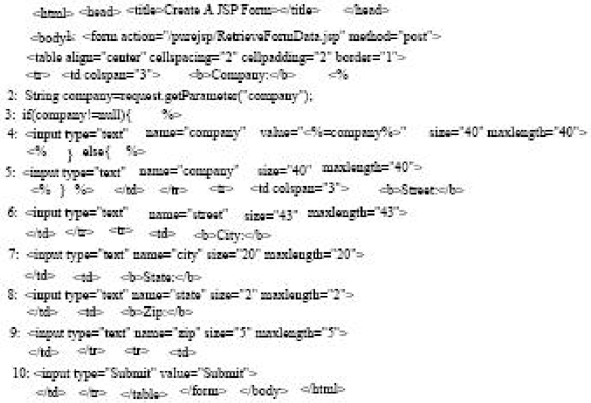
## Abstract

*We propose an efficient technique for slicing web applications. First we construct the system dependence graph for a web application and then perform backward slicing on that graph corresponding to a given slicing criterion. We use java server pages for the web application.*

## 1. Introduction

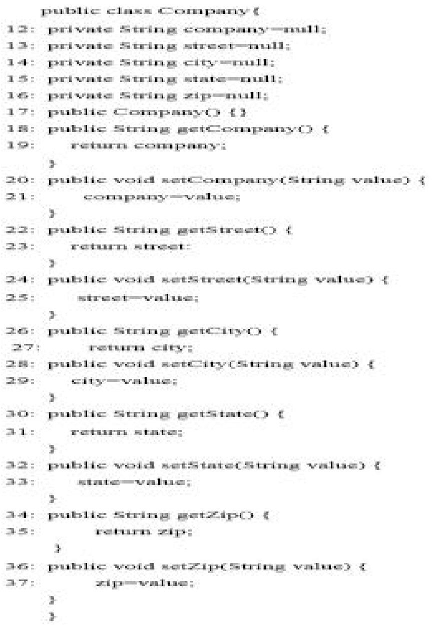
Due to the quick development of Internet, there is a need for the improvement of the quality of web pages and web applications. The Program Slicing technique is used for easy understanding, testing and maintenance of web applications.

In this paper, we present an approach to slicing the web applications using Java Server Pages (JSP) [2]. The rest of the paper is organized as follows. Section 2, describes our proposed slicing algorithm using System Dependence Graph (SDG). Section 3 concludes the paper.



```
    <html> <head> <title>Create A JSP Form</title>     </head>
    <body>  <form action="/purejsp/RetrieveFormData.jsp" method="post">
   <table align="center" cellspacing="2" cellpadding="2" border="1">
   <tr>  <td colspan="3">      <b>Company:</b>       <%
2:  String company=request.getParameter("company");
3:  if(company!=null){        %>
4:  <input type="text"    name="company"   value="<%=company%>"    size="40" maxlength="40">
    <%     }  else{    %>
5:  <input type="text"   name="company"    size="40"  maxlength="40">
     <%  }  %>    </td>    </tr>     <tr>    <td colspan="3">       <b>Street:</b>
6:  <input type="text"     name="street"    size="43"  maxlength="43">
    </td>  </tr>    <tr>    <td>     <b>City:</b>
7:  <input type="text" name="city" size="20" maxlength="20">
    </td>    <td>     <b>State:</b>
8:  <input type="text" name="state" size="2" maxlength="2">
    </td>    <td>    <b>Zip:</b>
9:  <input type="text" name="zip" size="5" maxlength="5">
    </td>    </tr>    <tr>    <td>
10: <input type="Submit" value="Submit">
     </td>   </tr>   </table>  </form>   </body>  </html>
```

Figure 1. An Example HTML Form using JSP: *CreateForm.jsp*



```
     public class Company{
12:  private String company=null;
13:  private String street=null;
14:  private String city=null;
15:  private String state=null;
16:  private String zip=null;
17:  public Company() {}
18:  public String getCompany() {
19:     return company;
     }
20:  public void setCompany(String value) {
21:        company=value;
     }
22:  public String getStreet() {
23:     return street:
     }
24:  public void setStreet(String value) {
25:      street=value;
     }
26:  public String getCity() {
 27:       return city;
28:  public void setCity(String value) {
29:     city=value;
     }
30:  public String getState() {
31:     return state;
     }
32:  public void setState(String value) {
33:      state=value;
     }
34:  public String getZip() {
35:         return zip;
      }
36:  public void setZip(String value) {
37:         zip=value;
     }
     }
```

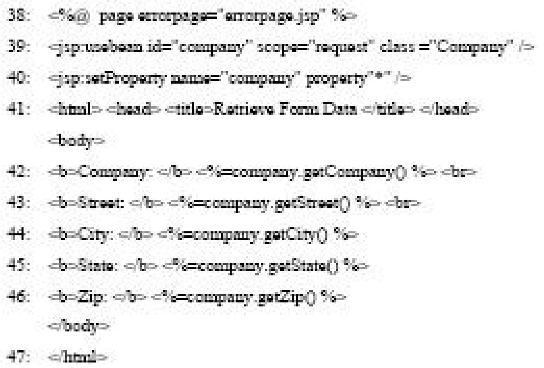
Figure 2. A Java Bean used to encapsulate simple comapny data: *Company.java*

## 2. Construction of System Dependence Graph (SDG) and Slicing Algorithm

Program slices are computed with respect to a slicing criterion. The slicing criterion for a web page is the information of interest displayed in the page. The slices are computed using a System Dependence Graph (SDG).

Figure 1 gives an example of creating a form to gather a user's company's name and shipping address. Figure 2 gives the source for creating a Java bean. Figure 3 contains the source for the JSP used to retrieve the form data. Figure 4 shows the SDG.

978-0-7695-3513-5/08 $25.00 © 2008 IEEE
DOI 10.1109/ICIT.2008.15



```
38:  <%@  page errorpage="errorpage.jsp" %>
39:  <jsp:usebean id="company" scope="request" class ="Company" />
40:  <jsp:setProperty name="company" property"*" />
41:  <html> <head> <title>Retrieve Form Data </title> </head>
     <body>
42:  <b>Company: </b> <%=company.getCompany() %> <br>
43:  <b>Street: </b> <%=company.getStreet() %> <br>
44:  <b>City: </b> <%=company.getCity() %>
45:  <b>State: </b> <%=company.getState() %>
46:  <b>Zip: </b> <%=company.getZip() %>
     </body>
47:  </html>
```

Figure 3. Source code for JSP used to retrieve form data: *RetrieveFormData.jsp*

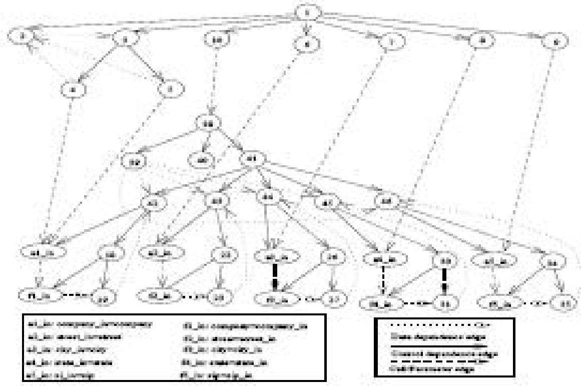


Figure 4. System dependence graph

**Algorithm: Construction of SDG**

```
1. For each JSP page, do the following
      (a) For each predicate i, compute control
          dependence for i
      (b) For each <form> tag or <html> tag f,
          compute control and data dependence
          for f
2. If <input> tag contains a "Submit" button, add call
   dependence edge to the corresponding JSP page
3. For each method m used in JSP Page, do the
   following
      (a) Copy procedure dependence graph for m
      (b) Add parameter edges
4. For each <input> tag except buttons, add control
   dependence edge to corresponding parameters
```

The following algorithm is used to perform slicing on JSP pages.

**Slicing Algorithm:**

```
1. Construct the SDG.
2. For a given slicing criterion <s, v>, mark s.
3. Traverse backward starting from s and mark the
   edges and nodes.
4. Map the marked node except parameter nodes to
   the corresponding statements in the program.
```

The algorithm is applied at statement 42 to find the slices for the value retrieved by the function *getCompany()*. The required slice can be obtained by traversing backward in the SDG and marking the nodes and edges. The resultant SDG is shown in Figure 5. The statements included in the slice are shown in Figure 6.

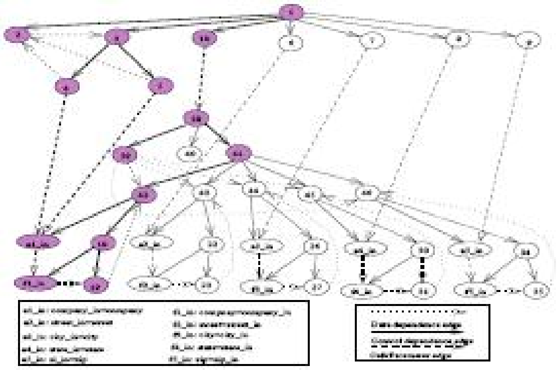
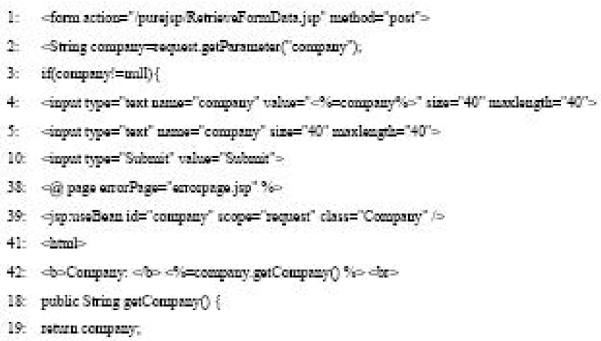


Figure 5. System dependence graph with marked nodes and edges



```
1:   <form action="/purejsp/RetrieveFormData.jsp" method="post">
2:   <String company=request.getParameter("company");
3:   if(company!=null){
4:   <input type="text name="company" value="<%=company%>" size="40" maxlength="40">
5:   <input type="text" name="company" size="40" maxlength="40">
10:  <input type="Submit" value="Submit">
38:  <@ page errorPage="errorpage.jsp" %>
39:  <jsp:useBean id="company" scope="request" class="Company" />
41:  <html>
42:  <b>Company: </b> <%=company.getCompany() %> <br>
18:  public String getCompany() {
19:  return company;
```

Figure 6. Slice computed at statement 42

## 3. Conclusion

We have proposed an algorithm for slicing web applications based on JSP. We have constructed the SDG for the web pages. The slice is computed by traversing backward in the graph and marking the edges and nodes.

## References

[1] Ricca F. and Tonella P. Web application slicing. In *Proceedings of the IEEE International Conference on Software Maintenance (ICSM'01)*, 2001.

[2] Goodwill J. *Pure Java Server Pages*. Techmedia, 2005.

[3] Wu J., Xu B. and Jiang J. Slicing web application based on hyper graph. In *Proceedings of the 2004 International Conference on Cyberworlds (CW'04)*, 2004.