

# An Efficient Approach of Building a New Software Web Framework for Better Software Solutions with DAO Layer

**Digambar Y. Yedage**

*Department of Computer Engineering  
SKN Sinhgad Insitute of Technology and Science,  
Kusgaon, Lonavala.  
University of Pune, Maharashtra, India*

**Anand Bone**

*Department of Computer Engineering  
SKN Sinhgad Insitute of Technology and Science,  
Kusgaon, Lonavala.  
University of Pune, Maharashtra, India*

**Abstract**—The Object Relational Mapping is considered as an important way of implementing the data persistent layer. Many of the application development pass through the software development life cycle. This paper focuses on accessing data persistent layer very efficiently in order to build a new software web framework. This new software web framework will have focus on easy access of data access layer in other layer of the software. Whenever a new technology gets introduced the migration of existing code into the new one takes more time and requires careful attention. With the help of newly developed software web framework the migration of existing code becomes in a very easier way. The new software frameworks are developed day by day in order to provide the better engineering solutions to common recurring problems. This paper describes Object Relational mapping framework for data persistent layer which is built on Java Server Pages Tag Library. The JSP tag libraries will also help in order to have prototyping of any of the software solutions in a faster and efficient way.

**Keywords**— Object Relational Mapping, Software Web Framework, design, custom tags, Prototyping, JSP Tags.

## I. INTRODUCTION

Translating a data access object between different types of databases was very difficult in earlier days. Most of the database related access use to happen with the help of JDBC. In order to have an efficient access of these data access objects in to your presentation layer different frameworks are used. There are many ways in the software systems wherein the presentation layer will have communication with the controller and then will perform the service related activities in to the service layer [1]. The service layer will send the properly formatted inputs received from the presentation layer to the DAO layer. This DAO layer will have the responsibilities of sending all the required information to the presentation layer. With the help of such type of access to data persistent objects between the presentation layers to DAO layer with the help of service layer requires significant amount of coding.

In order to access the data access objects very efficiently into the JSP requires use of custom tags. JSP is the combination of HTML and Java technology [2]. There are different ways to differentiate between business layer and presentation layer. Business layer is basically used to give the core business related functionality. But the presentation

layer only concentrations only on the look and feel of the application.

The Object Relational Mapping is being considered as most important way to implement the data persistent layer. Here in ORM the data is efficiently being hold in the databases. This method helps the developer to map the java objects to the relations from the database. These java objects are POJOs and the relations are database schemas [3]. These mapping between object and relations is shown in fig.1. To perform any of the database related operations these objects can be used.

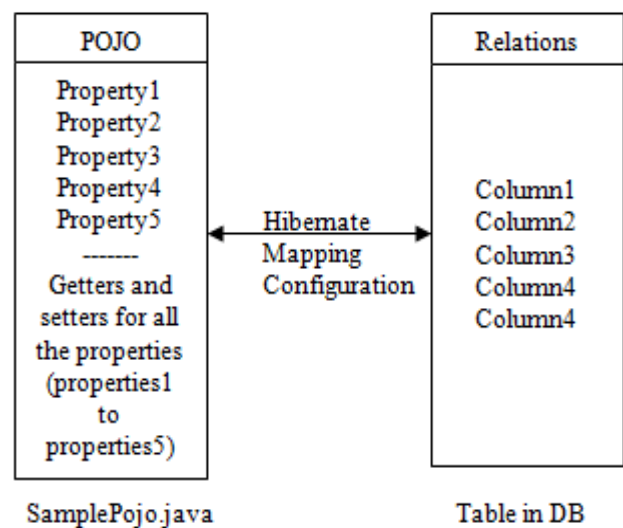


Fig.1 Object Relational Mapping

From the POJO class all the properties are being mapped to the specific columns in the relations schema of database.

## II. BACKGROUND AND RELATED WORK

Java Server Pages is one of the most powerful technology used for building of software web application which serves the dynamic contents. In earlier days web applications developed with the help of JSP consists of Java code embedded into the HTML code. The JSP Scriptlets are used to embed java code into HTML [4].

Use of these scriptlets introduces the complexity in the software development. In order to prototype of the software

system use of scriptlets and HTML code is generally not preferred because it requires the knowledge of Java technology along with the UI technology [5]. These scripting elements help in adding piece of java code into the JSP. These scripting elements are of three types these are scriptless, expression and declarations. These scripting elements in JSP will be executed each and every time for every server request for that specific JSP. In order to avoid this it is better to move the scriptlet code into embedded java instead of writing in JSP. The scriptlet code begins with `<%` and ends with `%>`. To access the JDBC code inside the JSP all the JDBC related code is written in JSP inside the scriptlets.

```

<html><body>

<%
// JDBC Code
Class.forName("oracle.jdbc.OracleDriver");
Connection con =
DriverManager.getConnection(CONNECTION_STRING);
Statement stmt = connection.createStatement();
ResultSet resultset = stmt.executeQuery("select * from
Table1);
// iterate the result set
%>

</body></html>

```

Fig. 2 JDBC in Scriptlet

The fig. 2 gives an idea of accessing the data access code into the JSP with embedded JDBC code into the HTML tags. This has many disadvantages in order to provide the software solutions. The UI developer will have knowledge of UI technology only. The UI developer works with different types of tags to develop prototype of any of the software system. In order to access the data access object directly with the help of custom tags it is required to have own custom tags for accessing database related functionality into the presentation layer [7]. The DAO layer can directly be accessed into the presentation layer with its own set of JSP custom tags. This approach of accessing DAO layer improves the coding significantly.

### III. IMPORTANCE OF NEW FRAMEWORK

The intention of developing a framework is to provide an efficient access to the DAO layer into the presentation layer. The development of pilot project is done at the early stage of the actual development of the software systems. The purpose of the developing a prototype is to get to know in depth of the way final product will look like [8]. The requirements will be made very clear by implementing the customer feedback at the early stage of the development of the software systems. Initially the prototype is developed with the help of the current software requirement specifications and then this prototype will be sent to the customer for the review purpose. Once it is being reviewed and if there are any of the customer review comments if provided by the customer then needs to be implemented by

the UI developer. The UI developer will implement all the review comments in the existing prototype and again will be sent to the customer for review. If there any review comments for the updated prototype then will be implemented else it will be sent for the design phase of the software development life cycle.

The development of prototype of software system will be very slow if developed using the software architecture. This will also takes more time to write presentation layer, service layer then database layer. If everything is written in the presentation layer then also it takes more time to develop the software systems. To access the DAO layer in the presentation layer also takes more time as it needs to write the JDBC code in the scriptlets.

The custom tags will help to access the DAO layer with Hibernate framework into the presentation layer will help to manageable and easily accessible for the UI developer.

### IV. IMPLEMENTATION DETAILS

The DAO layer will be accessed with hibernate framework in the presentation layer will need to have its JSP custom tags. These custom tags will have direct interaction with any of the databases [9]. As initially it's not decided which framework or technology will be used to develop the entire system it is better to have any of the databases which can be changed as per the system requirements.

#### A. Creating a POJO class:

A sample java class with attributes and setter as well as getters for all the attributes is given below. This is an Object written in Java. This will be considered in ORM mapping in Hibernate.

```

public class Products {

    private int productId;
    private String productName;

    public void setProductId(int productId){
        this.productId = productId;
    }
    public int getProductId(){
        return productId;
    }
    public String getProductName(){
        return productName;
    }

    public void setProductName(String productName) {
        this.productName = productName;
    }

}

```

#### B. Object Relational mapping configurations:

The ORM configuration file consists of mapping between object and the relational from the database. All the column and attributes are being mapped in this configuration xml. This xml will also have following

structure. The name of this file is given as \*.hbm.xml. This file needs to be referred from the hibernate configuration xml file as a mapping resource attribute.

```

<hibernate-mapping>

<class name="orm.Products" table="PRODUCTS">

<id name="productId" column="pId">
  <generator class="assigned" />
</id>

  <property          name="productName"
column="pname" />

</class>

</hibernate-mapping>
    
```

**C. Writing of tag handler class:**

The tag handler class extends the TagSupport class and overrides its required methods.

```

public class SampleTagHandler extends TagSupport{
    private String table;
    private String column;

    public int doStartTag() throws JspException {
        JspWriter out = pageContext.getOut();
        SessionFactory factory =
        HibernateUtil.getSessionFactory();
        Session session =
        factory.openSession();
    }

    public String getTable() {
        return table;
    }
    public void setTable(String table) {
        this.table = table;
    }

    public String getColumn() {
        return column;
    }
    public void setColumn(String column) {
        this.column = column;
    }
}
    
```

**D. Tag library definitions:**

The tag library definitions will have set of tag names which will be referred in the JSP. This tag has the name and attributes associated with each tag. In the JSP only attribute names needs to be mentioned.

```

<taglib>

<uri>aggregations.SampleTagHandler</uri>

<tag>
  <name>avgOf</name>
  <tagclass>aggregations.
SampleTagHandler TagHandler
</tagclass>

  <info>Average Details</info>
  <attribute>
    <name>table</name>
    <required>true</required>
  </attribute>
  <attribute>
    <name>column</name>
    <required>true</required>
  </attribute>

</tag>
.....
.....
</taglib>
    
```

**V. RESULTS**

Use of tag library will help in improving the coding approach. The UI developer needs to access the DAO layer in the presentation layer with the help of following tag in the JSP.

```

<aggregations:avgOf column="price" table="Products">
</aggregations:avgOf>
    
```

For sample test purpose mysql database with below list of columns with data is being used. The product table has the price column where the aggregation functions can be applied on the table.

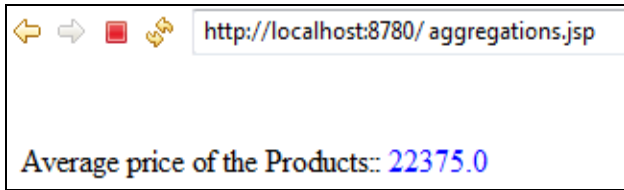
pId	pname	price
101	Television	45000
102	Laptop	35000
103	Chair	500
104	Microphone	9000

When the JSP with the aggregation function custom tag is written then it will have aggregation result.

The average of the price column is applied directly by the UI developer with very simple tag.

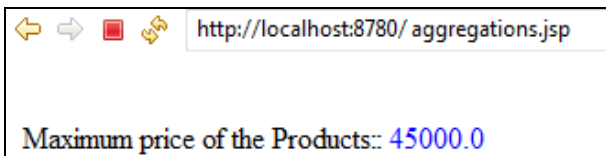
```

<aggregations:avgOf column="price" table="Products">
</aggregations:avgOf>
    
```



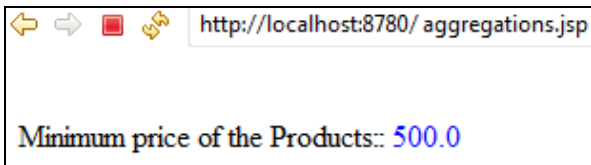
The maximum value of price column in the product table will be written by

```
<aggregations:maxOf column="price" table="Products">
</aggregations:maxOf>
```



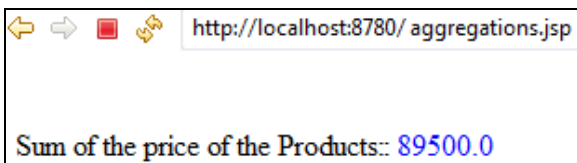
The minimum price value will be directly accessed by the UI developer with the help of below tag.

```
<aggregations:minOf column="price" table="Products">
</aggregations:minOf>
```



The sum of the price value will be directly accessed by the UI developer with the help of below tag.

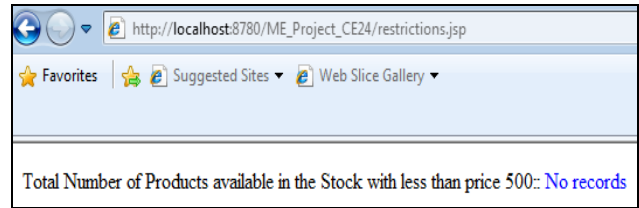
```
<aggregations:sumOf column="price" table="Products">
</aggregations:sumOf>
```



Also there are other similar operations are being performed which are related to the aggregation operation.

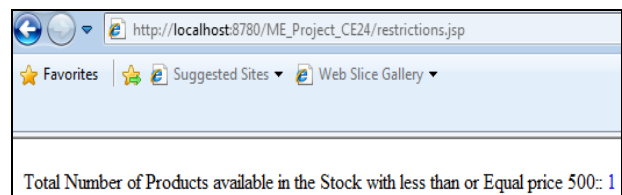
The restriction related DAO will be used by the UI developer with the help of restriction tags. The less than value of price column in the product table will be written by

```
<restrictions:lessThanOf column="price"
table="Products" value="500">
</restrictions:lessThanOf>
```



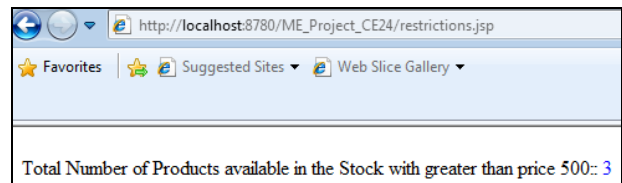
The less than or equal to operation is performed on the specific column in the given database and specified table. The UI developer will be able to fetch less than or equal to specific value for the column in the table with below syntax.

```
<restrictions:lessThanOrEqualOf column="price"
table="Products" value="500">
</restrictions:lessThanOrEqualOf>
```



The greater than of operation is performed on the specific column in the given database and specified table. The UI developer will be able to fetch greater than to specific value for the column in the table with below syntax.

```
<restrictions:greaterThanOf column="price"
table="Products" value="500">
</restrictions:greaterThanOf>
```



Similarly there are other important and related custom tags will be created for efficient access of the DAO layer for easy access of DAO layer in the presentation layer.

## VI. CONCLUSION

New way of accessing the DAO layer in the presentation layer will help in minimizing the coding task. The DAO layer will have set of custom tags developed as a part of the new framework. This does not require the knowledge of Hibernate and any other programming language for the presentation layer. This framework is the perfect solution for developing rapid application and pilot projects. Customers requirement will be finalized at the early stage of the development with the help of customer feedback approach in the prototyping approach of the software development.

#### ACKNOWLEDGMENT

Authors are thankful to Dr. Ramachandra V. Pujeri, Prof. V. D. Thombare, Prof. S. B. Sarkar and Prof. P. R. Barapatre for the continuous guidance and support.

#### REFERENCES

- [1] Chunna ZHANG, Wenxuan SHEN, "The Studies on Integrated Development Approach Based on the Architecture and MVC Design Pattern", IEEE Symposium on Electrical & Electronics Engineering (EESYM) on June 2012
- [2] Pubudu Gunawardena, 99X Research, 99X Technology, "Efficient Access to Hibernate Through JSP Powered by a New Tag Library", in Information and Automation for Sustainability (ICIAFS), IEEE 6<sup>th</sup> International conference on Sept. 2012
- [3] Yingyidu Xiong, "The design of automatically generating drop-down menu on JSP", in Electrical & Electronics Engineering IEEE, International Conference on Computer Science and Information Processing (CSIP), June 2012
- [4] Shannon Xu and Thomas Dean, "Transforming Embedded Java Code into Custom Tags", 2012
- [5] Ming Xue and Changjun Zhu, "Design and Implementation of the Hibernate Persistence Layer Data Report System Based on J2EE", IEEE, Pacific-Asia Conference on Circuits, Communications and System, May 2009
- [6] Shu-qiang Huang and Huan-ming Zhang, "Research on Improved MVC Design Pattern Based on Struts and XSL", IEEE, International Symposium on Information Science and Engineering, Dec 2008
- [7] Shannon Xu and Thomas Dean, "Modernizing JavaServer Pages by Transformation" Seventh IEEE International Symposium on Web Site Evolution, Sept. 2005
- [8] X. Guo, J.R. Cordy and T.Dean, "Unique Renaming of Java Using Source Transformation", IEEE 3rd International Workshop on Source Code Analysis and Manipulation, Amsterdam, p151-160, September 2003.
- [9] N. Synytskyy, J.R. Cordy and T.R.Dean, "Resolution of Static Clones in Dynamic Web Pages", Proc. WSE 2003, IEEE 5th International Workshop on Web Site Evolution, Amsterdam, p49-58, September 2003.
- [10] F. Ricca, P. Tonella, and Ira D. Baxter, "Web Application Transformations based on Rewrite Rules", Information and Software Technology., vol. 44, n. 13, pp. 811-825, October 2002