The Data Exchange on XML and JSP Based E-Commerce Platform

Yuechuan Luo[1] **Jiufeng Yu**[1+]

[1]*College of Computer Science and technology,ShanDong Economic University, Jinan,China,250014*

Email: *luoyaochuan@sina.com*

Abstract: XML is one of the core technologies of Web applications for the next generation. How XML technology applications to Web information integration system, The paper will introduce a XML and JSP based on Web application platform, and discuss this calculation model.

Keywords: XML,JSP,Web application

1 Introduction

The rapid development of Internet, made it become a huge information container. With the popularization of it's application, information resources Internet accumulated have expanded sharply, and it has developed from the original information publishing platform to information exchanging platform. Such a change will make vast resources of Internet integrate into the traditional information processing system, therefore Web computing model has also developed from the early B/S two-tier to 3 layers even n layers model. However, Internet technology which takes HTTP and HTML as the main means can't satisfy this integration demand. On the other hand, because of the heterogeneity of Internet, the data formats of each side are not exactly identical, this requires to solve the question of data exchanging effectively among a variety of data sources.

The paper introduced a Web information integration model based on XML and JSP; describing a variety of data sources on Web by using XML document and an unified data model. Retrieved and operated a variety of different data sources with cross-platform JSP technology, met the requirement that Web information integration system using different data sources and dynamic contents. (For example, one company used this model in develop it's logistics management system, achieved Web information integration system based on a XML and JSP.)

2 Web calculation model based on XML

2.1 Web architecture based on XML and JSP

With the combination of Web and Distributed object technology, Traditional C/S master-slave structure gradually evolved into a flexible multi-level distributed Web. Web architecture based on XML and JSP can be divided into three layers: Presentation layer (Client layer), Functional layer (Application server layer) and Data layer (Database Server Layer).

From all layers of the network , the specific implementation of three-tier C/S model based on XML and JSP are summarized as following:

Data layer—it achieved data integration , XML data generated in a variety of data sources, but expressed and transmitted with a unified XML format.

Functional layer—it achieved data delivery and treatment, the middle layer using JavaBean technology[1], plays a pivotal role. On the one hand, it put query request came from interface into a language that can be identified by database, on the other hand, according to the request of interface, it connects with database, getting the result, and sending result to Presentation layer; application sever can also handle XML data through DOM.

Presentation layer—it achieved data show, XML data can have a variety of manifestations, and can be visited, edited or converted by exterior, it can also be used by other systems.

In the three-tier C/S model based on XML and JSP,XML is treated as a way of structured information exchanging, taking charge of all communications with the data sources,collecting and organizing data accessed from multiple remote database server according to the final user's demand, and passing the messages data sources have returned to the client machine with the form of XML interactively.

The advantage of XML is mainly reflected in its characteristics of description of the data. This is a significant difference between XML and HTML. Using some related technologies such as XML and format DTD, Extensible Style Language (XSL) as the data description tools and conversion tools of the integration layer, not only can adapt to the needs of Web development, but also greatly simplify the implementation of Web Data Integration System[2]. XML document structure nesting can be complicated to any degree, it can express object-oriented hierarchy. XML has a great advantage in

Supported by Shandong province science and technology project (J05G17)

application of Electronic Commerce for its data description and its language characteristics of structured data.

Because XML can self-define the file type, it is conductive to the expression of information and structured organizations, and it can format and send data with a consistent way. Using JSP to server for XML, that is using XML document to store data on Web server, displaying data with JSP, thereby finding a channel which meet Web system using different data sources and dynamic contents. Putting XML and JSP into three-tier C/S model will bring a huge advantage of network applications[3].

The following will analyze application process of XML and JSP based three-tier C/S model.

2.2 Data storage of XML document on Web server

XML is a original language that allow user to define his markup language. It is no longer a fixed markup, but a information structure allows nesting and define unlimited number of Tags for describing data in document. HTML is a common method that Web display data, while XML provides a common method processes data directly. HTML mainly describes the display format of Web pages, while XML describes the contents of Web pages.

Meanwhile, XML is suitable for storing data and documents (XML documents), they can be XML plain text documents, they may also stem from relational database and other types of data sources. The former store data in the text files make it easy to be accessed or display the style sheet in a browser, or connected to the other applications through DOM interface programming; while because data of the latter come from a variety of dynamic applications, it is good to manage the data through database system, then use server-side applications (such as JSP) for dynamic access[4].

The combination of database technology and Web technology is the trend of MIS application system, but now Web technology has a very obvious defect: (at the present time) there are a lot of texts, pictures, sound and images on Web, they are usually stored in HTML document, without strict definition of structure and type, and can bring some problem such as the search engines often return a great deal of garbage data after the user input the key words. However, they have a maximum of applications on data manipulation and storage.

As a hierarchical format, XML can design data structure with a direct way to meet the requirement, now it is not necessary to use a Entity-Relationship editor and Chart Standardization. As long as there is an element contains another element, you can express it in the format directly without using the correlation of table. So we can convert the original Web database which was written by HTML into the database of XML format, use XML to construct the Web (all XML documents) based Data Warehouse can address these deficiencies. In addition, XML adapts to the exchange of object-oriented database, besides that, it adapts to information exchange of the traditional relational database.

With this idea, we will make a two-way data conversion between XML documents and a variety of database, put XML documents into Web server for users' access, thereby complete the MIS system's operation to a variety of data sources.

Database structure is mapped to the XML document:

- Create a XML document element for the object in database;
- Express each simple data field contained by objects with the property of elements;
- Use the above steps to generate a new element for each sub-object field, the new generated element is treated as sub-element of the current element; Until all of the domain has been completed.

The XML document is treated as an object tree, XML document structure is mapped to the database structure:

- Search from the leaves to root according to the principle of limited depth;
- Each encountered an internal node (except leaf nodes and root node), examine the corresponding abstract data type in database to exist or not, if not, create corresponding abstract data type, the abstract data type should treat all direct child nodes of this node as its domain; do the above job repeatedly as far as the direct child nodes of root node(or sub-root node);
- Create an corresponding sub-root node based on object of abstract data type, this object corresponds to the root node;
- Fill each element's property of XML documents into the corresponding domain of the object.

We also can use this technology and treat the XML document as middleware to achieve all exchanges of database information.

Although the XML document adapts to data exchanging and reading among applications, it can't be showed on Web page, it shows through the external style sheet, the external style sheet can be CSS file or XSL file. Now the XML document using a command to explain the style sheet: <? xml : stylesheet type="text/css" href="file name.css"?> explains that displaying XML document with CSS style sheet, while "type="text/xsl" href="name.xsl" means that displaying XML document

978-1-935068-18-1 © 2010 SciRes.

with XSL style sheet.

2.3 Interaction of JSP and XML

JSP page can use XML with three methods: direct use of XML, use JavaBeans to implement XML and use XML through the tag library.

2.3.1 Direct use of XML

Direct use of XML in JSP page can be divided into three categories:

- JSP can read the XML document and then implement some actions base on these data. For instance, an application program can read the XML document whose data contains some specific structure.
- JSP can create the XML document to send data to client programs or other application programs. JSP can convert the XML document, this conversion can be disposed by XSLT and treat JSP as a controller, or can be finished through non-XSTL solution. In both cases, the role of JSP is to read the XML document, convert it and generate an output.
- Directly call an analytical procedure to read/write XML data. But this is a very unreasonable method, because such data and code logic will not be able to be well separated.

2.3.2 Use JavaBeans

JSP can integrate with JavaBeans tightly with < jsp : usebean> tag. The following procedure demonstrated how to set and obtain property with a JavaBean in JSP page.

<jsp : useBean id= "cb" scope= "session" class = "xmlrep. Customer"/>

<jsp:setPropertv name="cb" property="id" value="45"/>

<B>First Name is:</B>

<%=cb.getFname()%>

<B>Last Name is:.</B>

<%= cb.getLname()%>

JSP and JavaBeans integrated feature can automatically translate form elements of HTML into JavaBean. If there is a HTML form, and we want it to submit form contents for JavaBean, we can implement it with following codes: <jsp:setProperty name="cb" property="'?" value="?"/>

Property of name contains the value that JSP page has cited Bean, like the above <jsp:urseBean> tag, it has set the name of "cd", set "Property" as its property, set "value" as its value. Different from setting a separate Bean property, we can mark all properties with "*": <jsp:setProperty name="cb" property="*"/ >, JSP page will automatically map HTML form value to a homonymic Bean property. If it read each HTML form element, then call the Bean method of corresponding property, the result is the same.

2.3.3 Implement interaction of JSP and XML through the tag library

In JSP, a tag seems like a standard HTML tag, its logic isn't implemented on client but on server as a part of Servlet converted from JSP. Each tag is packaged in a separate category, its name and parameters are displayed in a document of special configuration descriptor with "tld" extensions. In JSP, Users can customize a variety of tag libraries for using.It is the most important and flexible method to implement interaction of JSP and XML through tag library.

Tag library can be defined in JSP page as a user-defined label of class XML elements. We can associate a specific Java code with each tag. For example, to assume that a user can access a database of weather conditions, and will need to output the current weather conditions. So he can insert JDBC codes in JSP to query database, package these codes in a tag library, add the following code in JSP page:

<%@taglib uri="the TLD file" prefix="foo"%>

Current weather is<foo:Weather/>

In the tag library, there is a descriptor file associated with XML document, named Tag Library Descriptor (TLD). In TLD, each tag has an entry contains its name, edition and other optional information. In JSP, page designers can designate TLD document through "<%@_ tag lib prefix = "foo"%> ". The "prefix" was used for specifying a prefix, citing any tag with a particular library in JSP. That's why to use tag of <foo: Weather/> rather than just <Weather/>. The exact location of TLD document depends on the application server which is being used.

A tag in tag library can replace the corresponding Java program code which complete the program logic. Each tag is equal to a Java class of the same name. This class must implement the interface of TagSupport and contain the capture flip-flop case method as a treatment for this page JSP engine. When it met the tag for the first time, the engine would call the method doStartTag (). We can empty this method or implement the application logic when we need it. When this method has returned SKIP_ BODY, the engine would skip this tag body. When it returned EVAI_BODY_INCLUDE, the engine would dispose this tag and its subtags. Similarly, JSP engine will call the method doEndTag after it analyzed the end tag. Method of doAfterBody () allows you to implement some actions after the engine has disposed the

element body, but you must do it before the method doEndTag(). The following is a Weather class sample code to implement weather conditions:

```
import javax.servlet.jsp.*;
import javax.servlet.jap.tagext.*;
import java.io.*;
public class Weather extends TagSupport{
public int doStartTag(){
        try{
            JspWriter out=pageContext.get0ut();
            out.print(" sunny and cloudy mixed with rain and sunshine.");    }
        catch (IOException e){
            System.out.println("Error"+e);    }
        return (SKIP_BODY);    }
}
```

When the engine met the tag "<somePrefix: Weather/>", it would search a class of the same name. If the method doStartTag() implemented(like this example), it would be called. This makes the string contain the corresponding displayed weather conditions. Because the method has returned SKIP_BODY, the JSP reader moved to the end of tag.

3 Concluding remarks

Web-oriented information integration is a complex technology, this paper is trying to do some researches on the interactive technology of XML and JSP, XML is the basic standard of data exchange, data sharing and data storage, the only way to implement heterogeneous database sharing. JSP is an effective tool to create applications and server-side programs, it has a nature of cross-platform. The combination of JSP and XML is the best method to develop Web programs, it will have a broad prospect in the new generation of Web model.

References

[1] Zheng Y F. Web-based heterogeneous information source integration middleware. 2005, 25(1):81-84.

[2] Feng S R Constructing program generator based on XML and JAVA.2005, 22(1):57-60

[3] Gao J Q. Designed technique and applications of general query subsystem based on XML. 2006,27(1):165-168.

[4] Li Z Q. Semi-Structured Web Information Extraction Process Based on XML. 2007,(1):66-68

978-1-935068-18-1 © 2010 SciRes.