Information Management on Networks Using Weblog

Yoshiaki Hoshi s1090185 Supervised by Carl Vilbrandt

Abstract

We propose new tools for an emerging form of information management and communication on the Internet, the Weblog. With the development of the Internet where individuals can broadcast information, the Weblog is replacing the HTML page. The Weblog uses a browser interface and is easier to dynamically update than the HTML page. Furthermore, the Weblog allows the user to update and manage information with limited computer skills, and the user's data can be accessed and managed from any computer that is Internet and Web browser capable. Our application research project investigated methods of improving the use of Weblogs. Accordingly, we developed new tools for shared bookmarks, an easy note pad, and improved information management while preserving the ease of use and convenience of the Weblog environment. Finally, we demonstrate our new tools based on XML technology written in Java. The tools conserve the simplicity, ease of use and original constructs of bookmarks while extending them with automatic updating over multiple computers and networks.

1 Introduction

Today, it is common for people to use multiple computers. Computers on different networks at the company, the school, and the net cafe as well as the computer for home are used. Because the settings are different, multiple computer usage confuses the user. However, most computer users have learned how to use a browser to find information on the Internet. Browsers are a common application interface across all computer platforms and operating systems. Therefore, it is easy to keep individual information like bookmarks on the Web. If a computer can access the Internet, it is possible to use it to access and manipulate one's information if that information is stored on the Web, much in the same manner as downloaded data on a local hard drive. The advantage to this scenario is that one can access and use their data from any computer at any location as long as that computer is Internet capable. One of the best ways for keeping information on the Web might be a Weblog [1]. Although, the Weblog originally focused on the journal system site for the individual dispatch which sends news and impressions to many and unspecified people, it is now in fashion as a Web diary. Compared with the conventional homepage, the handiness of installation, updating, management and the ease of communication brought about its rapid popularity. If it is possible to connect to the Internet, it is possible to update the Weblog easily without using a special tool. Accordingly, the number of people with Weblogs calling Webloggers is increasing every year. Although there were 150,000 registrants of rental Weblogs in 2000, registrants exceeded 10 million people at the end of 2004. Rental Weblog service is also increasing simultaneously, and about 120 services exist in Japan. Reasons why the Weblog is suitable for simplicity in information storage and manipulation are given below:

- *Everyone can use the Weblog*: Common Web browsers, such as IE and Mozilla, are basically all that is necessary in order to use the Weblog. To acquire a new Weblob is easy by inputting individual user information. Typically, no more than ten minutes is needed to establish the Weblog.

- *Operation of the Weblog is easy*: Even a computer and Internet novice can operate most Weblog services. Moreover, posting, updating and deleting of entries can be done without knowledge of HTML such as one would need to maintain a homepage.

- *Weblog has free file space*: When registering, the Weblog service provides file space of tens of MB. This capacity varies according to the service.

- *No extra cost*: A lot of Weblog services do not charge to establish an account.

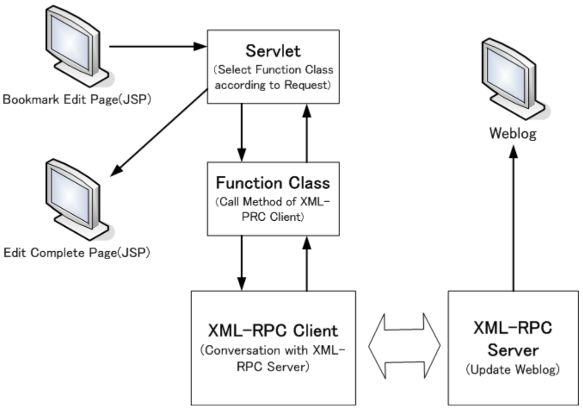
- *Easy to create an application*: Most existing Weblog services use standard Weblog API [2] of Extensible Markup Language-Remote Procedure Calling (XML-RPC [3]). Operation of most Weblogs can be developed with the client module of XML-RPC.

The goal of this research is to develop a Web application using the Weblog having information in common over multiple computers and many networks. The main functions of this application are a Web bookmark, simple memo pad, and file management. Our application research method is discussed in the following sections.

2 System Overview

The research application is divided into two parts, the XML-RPC client module to execute the XML-RPC communication processing and the Web application for the user to display, to operate, and to input. The XML-PRC client is constructed with Java [4], and the Web application is constructed with JSP/Servlet [5]. The XML-RPC client improves independence, and takes aim at packaging. The purpose of packaging is to flexibly respond to the occurrence of problems in the XML-RPC client and the Web application. Moreover, we anticipate other projects that would use this package. This application is made as a Web application in order to preserve the ease and freedom of the Weblog. When the user edits a bookmark edit from the Web browser, the class that does the processing from which the servlet is selected is called, the servlet receives the request, and calls the class that processes it. The class calls the XML-RPC client, it processes to the Weblog, and the displayed item is acquired. The acquired display item is preserved in the session. The display item is acquired in JSP, and displayed in a browser.

The concept chart of the system is displayed as below.



3 Method

This section explains requirements for application development.

3.1 XML-RPC

XML-RPC is a Remote Procedure Calling protocol that works over the Internet. An XML-RPC message is an HTTP-POST request. The body of the request is in XML. A procedure is executed on the server and the value it returns is also formatted in XML. Procedure parameters can be scalars, numbers, strings, dates, etc.; and can also be complex record and list structures.

```
1 :POST /RPC2 HTTP/1.0
 2 :User-Agent: Frontier/5.1.2 (WinNT)
 3 :Host: betty.userland.com
 4 :Content-Type: text/xml
 5 :Content-length: 181
 6 :
 7 :<?xml version=”1.0”?>
 8 :<methodCall>
 9 :      <methodName>examples.getStateName</methodName>
 10:      <params>
 11:            <param>
 12:                  <value><i4>41</i4></value>
 13:            </param>
 14:      </params>
 15:</methodCall>
```

The payload is in XML, a single <methodCall> structure. The <methodCall> must contain a <methodName> sub-item, a string, containing the name of the method to be called. This string calls both a process and the API it is going to use, as shown in line 9 of the sample code above. If the procedure call has parameters, the <methodCall> must contain a <params> sub-item. The <params> sub-item can contain any number of <param>s, each of which has a <value>. Parameters of <value> mean parameters of procedure call.

3.2 Weblog Operation Using XML-RPC

An operation is completed by transmitting XML to the path of the XML-RPC server of the Weblog. At that time, the <methodName> is designated as wanting to be executed. The XML-RPC client of Java takes charge of this processing. Details of the operation are described in the section 4.2.

3.3 Corresponding Weblog

The Weblog services using standard Weblog API are the object of this application research. Therefore, Weblog services using original Weblog API like Excite Blog are not considered by this research.

3.4 Registration of Bookmark Information

The bookmark information is registered in one article with the XML notation. The structure of the bookmark information includes books and folders. In addition, the book has book ID, title, URL, note and belonging folder ID, and the folder has folder ID, folder name and parent folder ID. With the appropriate folder ID or the parent folder ID, the class of the folder can be achieved. The sample XML data structure of the bookmarks is shown below.

```
<?xml version=”1.0” encoding=”UTF-8”?>
<Bookmark>
    <Folders>
        <Folder>
            <FolderId>1</FolderId>
            <FolderName>Portal site</FolderName>
            <ParentFolderId></ParentFolderId>
        </Folder>
        <Folder>
            <FolderId>2</FolderId>
            <FolderName>Shopping site</FolderName>
            <ParentFolderId>1</ParentFolderId>
        </Folder>
    </Folders>
    <Books>
        <Book>
            <BookId>1</BookId>
            <Title>Yahoo! Japan</Title>
            <Url>http://www.yahoo.co.jp</Url>
            <Note>Japanese Yahoo’s Homepage.</Note>
            <ParentFolderId>1</ParentFolderId>
        </Book>
        <Book>
            <BookId>2</BookId>
            <Title>Amazon.co.jp</Title>
            <Url>http://www.amazon.co.jp/</Url>
            <Note>We can buy books and CDs.</Note>
            <ParentFolderId>2</ParentFolderId>
        </Book>
    </Books>
</Bookmark>
```

3.5 Simple Memo Pad

As well as bookmarks information, the simple memo pad is registered in one article which has the items: title, text and date. The sample XML data structure of the memo pad is shown below.

```
<?xml version=”1.0” encoding=”UTF-8”?>
<MemoPad>
    <Memos>
        <Memo>
            <Title>Next appointment with proffesor</Title>
            <Text>01/12/05 Wednesday</Text>
            <EditDate>2005/01/10 12:44:32</EditDate>
        </Memo>
        <Memo>
            <Title>CD wanting it</Title>
            <Text>WHITE ROOM by YOSHII LOVINSON sale on March 9, 9 -NINE- by ULFULS sale on February 23</Text>
            <EditDate>2005/01/15 21:43:05</EditDate>
        </Memo>
    </Memos>
</MemoPad>
```

3.6 File Management

In this research, the file management function provided by most Weblog service is enhanced. The file uploading function is mounted using the package offered by Jakarta-Commons [6].

4 Experimentation

This section explains how we implemented the methods and evaluates the application.

4.1 XML-RPC Client

The <methodName> and some <value>s are necessary for XML-RPC. <value>s can be scalars; type is indicated by nesting the value inside one of the tags listed in Table 1:

Tag	Type
<i4> or <int>	four-byte signed
<boolean>	0 (false) or 1 (true)
<string>	string
<double>	double-precision signed
<dateTime.iso8601>	date/time
<base64>	base64-encoded binary

Table 1: Nesting Table of Tags and Input Types

It is necessary to do the mapping in this application tag and the Java object so that it is a nest of the input type shown in Table 1. The mapping of tag and the Java object is shown in Table 2.

Tag	Java Object
<i4> or <int>	java.lang.Integer
<boolean>	java.lang.Boolean
<string>	java.lang.String
<double>	java.lang.Double
<dateTime.iso8601>	java.util.Date
<base64>	byte

Table 2: Mapping Table of Tags to Java Objects

Therefore, the String type as <methodName> and the list of the Object type that stores some different type <value>s are necessary for argument of the method of the XML-RPC client in order to get results. This is the standard XML-RPC client for Java.

4.2 Commands and Results

However, the standard XML-RPC client for Java shown in Section 4.1 above is not convenient. Therefore, we make a command class and a result class. The command class has the XML-RPC client as the parents class. The command class succeeds as a function of the XML-PRC client. And, the command class has some child classes such as postEntryCommand class and updateEntryCommand class, which function based on the class name. For instance, the postEntryCommand class means posting the Weblog article class. The command classes have the result class corresponding to oneself. The result classes contain processing result informations. We can easily execute the XML-PRC client by using the command class and the result class.

4.3 Reading and Writing of XML

The bookmark information gathered by XML is read by using Digester. Digester is simple binding tool of XML to Java provided by the Jakarta-Commons project. Reading the XML file can be achieved in Digester much more easily than by using SAX [7] and DOM [8]. Loading can be processed easily by Digester, but Digester cannot output XML from the Java object. Therefore, writing XML should use other methods. Our research created the toString method that outputs all elements and values by a character string of XML form for the class of URL and the class of Memo.

5 Conclusion

We have demonstrated the development of new tools for Weblogs. We have conserved the construct of bookmarks and extended their use to Weblogs. The addition of bookmark features to the Weblog enhances the migration to Weblogs empowering the user to become a knowledge provider rather than just an information consumer.

For future work, it is necessary to add importing and exporting of bookmarks between the Weblog and various Internet browsers, thus easing the migration from consumer to provider. The import function would register the bookmark in a browser under use at the time, and the export function would output the bookmark in a form matching the browser under use. In addition, it would be necessary to merge and update the bookmarks between the Weblog and browsers over various computers and networks.

Acknowledgements

I am very grateful to my supervisor, Prof. Carl and Jody-san, for giving advice. Also, I would like to thank Hirohisa Mitshuishi, Syogo Hamada and Daichi Nakajima for your great support. Finally, I thank the various people who have helped me complete this research.

References

[1] 宮川 達彦, 伊藤 直也 ”Blog Hacks”, O’Reilly Japan, Inc, August 2004

[2] Standard Weblog API :
blogger API
(http://xmlrpc.free-conversant.com/docs/bloggerAPI)
metaWeblog API
(http://www.xmlrpc.com/metaWeblogApi)

[3] Simon St. Laurent, Joe Johnston, Edd Dumbill ”Programming Web Services With Xml-Rpc”, Oreilly Associates Inc, June 2001

[4] Java (http://www.sun.com/)
Bruce A. Tate, Justin Gehtland, Bruce Tate ”軽快なJava―Better,Faster,Lighter Java” O’Reilly Japan, Inc, September 2004

[5] 日立ソフトウェアエンジニアリングインターネットビジネス推進部, ”JSP/Servlet 逆引きサーブレット JSP リファレンス―やりたいことがすぐわかる機能引きリファレンスの決定版 アドバンストサーバサイドプログラミング”, 技術評論社, January 2001

[6] Jakarta-Commons (http://jakarta.apache.org/commons/)
File Upload (http://jakarta.apache.org/commons/fileupload/)
Digester (http://jakarta.apache.org/commons/digester/)

[7] SAX (http://www.saxproject.org/)

[8] DOM (http://www.w3.org/DOM/)