A-12

Developing a JSP parser with ANTLR

Phommalath Daovalath†　Megumi Kimura‡　Taku Fujii‡　Yukikazu Nakamoto†

1. Introduction

Web applications commonly are used for webmail, online sales, social networking website and many other functions. Web applications may consist of a simple HTML pages to complex applications (Java, JavaScript, Flash and other program languages). Web applications can be categorized into static web applications and dynamic web applications. JSP (JavaServer Pages) [1] is one type of dynamic web content technology. Web applications have high maintenance costs for two reasons. First, the applications comprise of heterogeneous technologies and have complicated structures. Second, web applications evolving over time add to their maintenance costs. Web application evolution [2] is primarily driven by two factors:

1) The law of continuing change: A program used in the real world must change over time or eventually it will become less useful in the changing world.
2) The law of increasing complexity: As a program evolves, it becomes more complex and extra resources are needed to preserve and simplify its structure.

Heterogeneous technologies such as Java, XML, JavaScript, HTML and other technologies may be used in one web application; speed of business logic change and technology advancement make maintenance costly.

JSP presents a convenient solution to web application development by using heterogeneous technologies to generate dynamic code. However, using JSP often results in complex code structures, which are difficult to modify or maintain. In order to improve JSP code usability and maintainability, it is necessary to first devise methods to measure the complexity of the code. We have chosen ANother Tool for Language Recognition (ANTLR) [3] as a JSP code analysis tool, since it is easy to integrate heterogeneous language parsers into ANTLR. In this research, we develop a parser for measuring complexity of JSP web applications through the ANTLR parser generation tool.

2. Background

Before considering the development of a parser for JSP web applications with the ANTLR parser generation tool, first we describe a model of web applications and JSP.

†Graduate School of Applied Informatics University of Hyogo

‡ OGIS-RI Co.,Ltd

2.1 Web application system model

Figure 2 shows the structure of a web application model.

(1) *A Client system* sends requests to the web server system. After the requests are processed, the client interprets HTML documents and displays them as web pages. The client is also known as a "*browser*".

(2) *Web application server system* outputs dynamical HTML documents which are generated by a program when requested by a client system. "Apache web server" [4] is currently the most popular HTTP server software for web applications such as Perl, Python, Tcl and PHP. JSP uses "Apache Tomcat (or Jakarta Tomcat or simply Tomcat)" [5], which is a servlet container developed by the Apache software [6].

(3) *A Web application system (Web application) a group of programs or configuration files* using dynamic web content technology such as ASP.NET, ASP, CGI, ColdFusion, JSP/Java, PHP, Perl, Python, Ruby on Rails or Struts2, in whose programs application logics are executed. The web application system queries and updates databases, generates dynamic HTML pages as a response, and forwards them to the client.

(4) *Database* is a system for storage of data.

2.2 JavaServer Pages (JSP)

```
1 <html>↓
2 <head>↓
3 <title>Hello, World!!</title>↓
4 </head>↓
5 <body>↓
6 <% out.println( "<p>Hello, World!!</p>" ); %>↓
7 </body>↓
8 </html>↓
```

Figure 1 HelloWord.jsp sample of simply JSP page

JavaServer Pages (JSP) [1] are a server-side programming technology; JSPs are an extension to the Java Servlet technology developed by Sun Microsystems and accomplished by embedding Java code in HTML, XML, DHTML or other document types (See Figure 1). Java Servlet is a Java class that runs on web application servers (See Figure 2); JSP syntax includes the following (Table 1) element tags:

Table 1 JSP element tags

Tag name	syntax
HTML tag	<…>
JSP custom tag	<prefix:tag name …>
Jsp comment	<%-- PCDATA --%>
Scriptlet	<% … %>
SP formula	<%= … %>

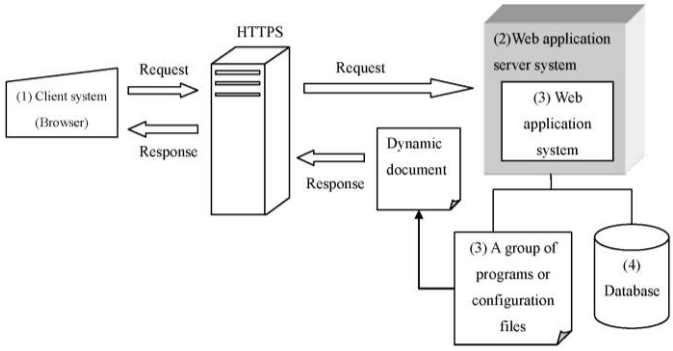

Figure 2 A generic web application model

2.3 ANTLR tool

ANother Tool for Language Recognition (ANTLR) [3] is a sophisticated parser generator, using the predicated-LL(*) [7] parsing mechanism. ANTLR can be use for generating parsers and interpreters for domain-specific languages. ANTLR supports multiple target languages such as Java, C, C#, C++, Objective-C, Ruby and Python.

3. JSP parser

First, we consider a case of a simple JSP which is in HTML form in figure 2. In this case we can use an XML parser or an HTML parser to parse this page. If we use an HTML parser, the parser only parses HTML tags and directive tags like `<%@ foo x="..." %>`.

To develop a JSP parser, we must consider the following features of JSP:

(1) JSP defines a custom tag which has a prefix name and followed by ":" with a tag name such as XML name space like `<x:foo>`.

(2) Java Script programs or Java programs written in JSP, a scriplet tag and a JSP expression tag can be used, such as `int i = <%= formname %>`

(3) JSP also contain sections written in a form of ${expr} Expression language (EL). For example:

`<foo ${condition: 'x="..."' ? ''}>` is written in EL.

(4) A JSP tag can be nested such as

```
<foo name="x" value="<%= y %>"/>
```

We must develop a parser which is capable of parsing such complex tag structures in JSP. We utilize the dynamic scoped attributed of ANTLR to implement a JSP parser. The parser is a simple recursive tree walker that can change, delete and add nodes during the walk.

4. Conclusions and future work

Web applications are becoming increasingly common and continue to grow in complexity and size. One of the largest challenges in web application development is the high maintenance cost associated with their complexity. Previously, web application structure has been studied in terms of metrics [8] for understandability, reusability, and maintainability. In this research, we developed a JSP parser to evaluate web application understandability. Future work shall include the development of a metric for measuring JSP pages' complexity. As a conclusion, web applications are growing faster and become more complexity. An important thing is that there is their high maintenance cost. Many web applications' metrics are studied for understandably, reusability and maintainability for reduce web application maintenance cost. We develop JSP parser for well-form and understand JSP structure. Our future work develops a metric tool to measure JSP's complexity [9].

References

[1] M. Hall and L. Brown: Core Servlets and Javaserver Pages: Core Technologies, Vol. 1 (2nd Edition), PRENTICE HALL, 2004.

[2] E. Ghosheh, J. Qaddour, M. Kuofie and S. Black: A Comparative Analysis of Maintainability Approaches for Web Applications, *Proc* IEEE International Conference on Computer System and Applications, pp.1155-1158, 2006.

[3] T. Parr, The Definitive ANTLR Reference: Building Domain-Specific Languages, The Pragmatic Bookshelf, May 2007.

[4] http://httpd.apache.org/.

[5] http://tomcat.apache.org/.

[6] http://apache.org/.

[7] http://www.antlr.org/blog/antlr3/lookahead.

[8] E. Mendes, N. Mosley and S. Counsell: Web Metrics - Estimating Design and Authoring Effort, *proc* IEEE Multimedia, pp. 50-57, 2001.

[9] W. Jung, E. Lee, K. Kim and C. Wu: A Complexity Metric for Web Applications Base on the Entropy Theory, *proc* IEEE computer society, pp. 511-518, 2008.