2011 Third International Conference on Communications and Mobile Computing

Improving Data Transmission in Web Applications via the Translation between XML and JSON

Guanhua Wang
School of Computer Science & Engineering
Southeast University
Nanjing, China
wghyy333@126.com

***Abstract*—This paper analyzes the form of two data serializing approaches used in web applications, XML and JSON. Considering that though both widely used, highly-efficient data transmission between these two methods is still a problem in application development. The features of these two data objects were analyzed and it was point out that how to translate correctly between these two objects. A recursive algorithm used to translate between these two types of data serializing forms was given based on the multi-tree data structure of XML and JSON objects. The efficiency of this algorithm was proved by translation experiments. When applied in web applications, this paper gives how to add the translation method in system structures of the applications. It gives how this method can improve system structure and the performance of the web applications.**

Keywords: XML; JSON; data structure; recursive; translate; Web Service; AJAX

I. INTRODUCTION

As two different approaches of data serializing, XML and JSON have been used widely in web applications. These two approaches are applied in data transmission between different layers of one web application and between different web applications. Two typical ones are the application in AJAX (Asynchronous JavaScript and XML) and Web Service. At the same time, because of the differences between XML and JSON in their features, developers often choose different serializing approaches based on specified situations while using these two approaches. Except for the application backgrounds used specially for data transmission, while develop the background of a common web application, in order to ensure unity and readability, often only one data serializing approach is used in development. Such choice makes it easy for programmers to develop data interfaces for data transmission based on features of the application. But if data transmission between a local web application and a third-party application is needed, the problem of data transmission between differently serialized data comes up. At this time, translation between XML and JSON becomes necessary.

Currently, when faced with the problem of translation between XML and JSON, web application developers do not give an independent and integral solution. Instead, data are translated based on their specific content format in different function models which are related with data translation. In this way, the problem of translation between XML and JSON can be solved by this function. But to the entire web application project, such solution splits an independent feature and puts it in different child models of a web application. This is not conductive to the analysis and management of system architecture of the web application. Meanwhile, redundant codes with duplicated functions will be produced, which will make the program bloated and less efficient.

In this situation, a universal and practical integrated translation solution will improve efficiency of data transmission of a web application, and optimize the web application from the level of system architecture, which will improve efficiency。

II. COMPARISON BETWEEN XML AND JSON

A. Introduction of XML and JSON

XML is full-called as eXtensible Markup Language. In the use of XML, according to prescriptive grammar, all data are stored in the tag closed strictly with the data indexed by labels. The specification of XML language ensures that the form of serialized data of XML has good security. At the same time, Tags scalability of XML language allows developers to configure depending on the characteristics of Web Service applications, so that Web Service applications can be adapted to different situations. When AJAX technology was first promoted, XML language was used in a wide range of AJAX development. However, with the popularity of AJAX technology, the inadequacies of XML when applied to interactive page is exposed. XML language has good specification, but its structure is bloated, leading to low efficiency of its analysis. Also, XML objects are analyzed as DOMs (Document Object Mode), which takes a long time. If the developers extensively use XML in lightweight AJAX applications for data exchange, the efficiency of the page will be significantly reduced.

The full name of JSON is JavaScript Object Notation. JSON is a lightweight data exchange format. Relative to the XML, it is easy for machines to parse and generate. Its format is different from the XML which uses closed tags. JSON uses a text format that is completely independent of

978-0-7695-4357-4/11 $26.00 © 2011 IEEE
DOI 10.1109/CMC.2011.25

IEEE computer society

the language. Unlike the DOM-style XML objects, JSON objects are analyzed as string arrays. This can be much faster. Compared to XML, with the above characteristics, JSON has obviously higher parsing efficiency and the advantages of easy preparation. As lightweight AJAX applications have less demanding on the security while requirements for efficiency are high and have good support of JavaScript, JSON, as an alternative to XML, has been largely applied.

B. Data structure of XML and JSON

XML language specification requires that all data are stored in closed tags. Labels' attributes can be extended. The data are analyzed marked by labels, while data can be tagged for the label, which allows nested tags. The following is a typical XML object.

```
<?xml version="1.0" encoding="UTF-8"?>
<contacts>
    <contact>
        <name>
            Jack Sparrow
        </name>
        <phone>
            123-532-5392
        </phone>
        <work>
            actor
        </work>
    </contact>
    <contact>
        …
    </contact>
    …
</contacts>
```

Note that XML syntax allows the existence of two labels with the same name at the same level as indicated in the example <contact> label. In XML-based HTML language, this is a common application, but when XML serialization is used for data transmission, the same label name is likely to cause the receiver of the data analysis data indexing confusion. When translating it to JSON, JSON objects do not support the same array subscript in one layer. It will cause differences between JSON result in translation and the original XML structure, this structure differences will be discussed below.

JSON objects have two forms: one for collection of "name / value" pairs. In different languages, it is realized as an object, record, structure, dictionary, hash table, keyed list, or associative array; two to the ordered list of values, in most languages it is interpreted as an array. The same with XML, JSON object value can also be another JSON object, allowing for nesting. Therefore, during the transmission of data serialization, JSON objects are often transmitted as a string, as a non-strict standard to handle an array of objects. Because JSON objects have the features of an array, so the JSON objects do not allow two or more of the same array index in the same level, which is the structure of JSON and XML is clearly different. Such as the XML object mentioned above, when it is translated into JSON form as "name / value" pairs, the duplicate <contact> label must be removed (the other way is to handle the duplicated tags with ordered list of values, will be discussed below), resulting JSON object structure.

```
{
    "contacts": {
        "contact": {
            "name": "Jack Sparrow",
            "phone": "123-532-5392"
            "work": "actor"
        }
    }
}
```

When this JSON object is transmitted, it is compressed into a string {"contacts":{"contact":{"name":" Jack Sparrow", "phone":" 123-532-5392", "work": "actor"}}}, and this string is sent as a data object instead of the XML form of DOM.

Through the above XML and JSON data structure analysis, we can see that, in considering the application of serializing in the "name / value" pairs form, when the structure of the duplicated XML tags is removed, both JSON and XML structure can be abstracted as a multi-branch tree structure, the example data of the above can be abstracted as a tree shown in figure 1.

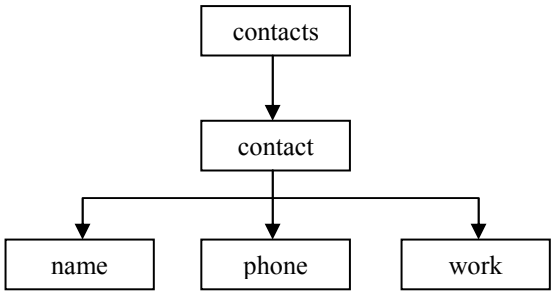

Figure 1. Data structure of "name/value" pairs

Corresponding ordered list of values in JSON, data objects can also be abstracted into multi-branch tree structure. As for the indicated XML object, if it is rewritten as ordered list of values in JSON, its repeated <contact > tags can be combined into an array, its subscript, which is an integer starting from 0, is hidden. The result of such abstraction is the following JSON object structure with array values include.

```
{
  "contacts": {
    "contact":[ {
      "name": "Jack Sparrow",
      "phone": "123-532-5392"
      "work": "actor"
    },
    {
      …
    }]
```

```
    }
}
```

The data structure of this JSON object is shown in figure 2.

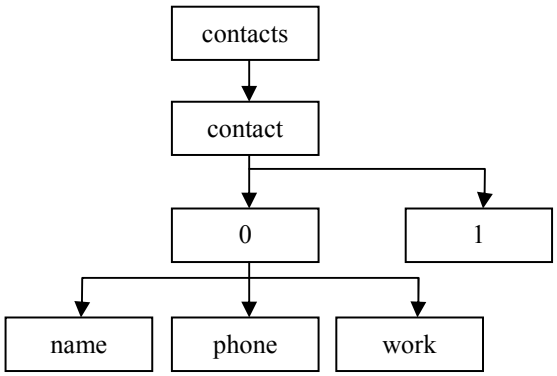

Figure 2. Data structure of ordered list of values

Comparing this data structure of ordered list of values with the original XML structure, it can be found that after abstraction, an array subscript layer is added to the structure. If this structure is restored to a XML object, all <contact> labels will be combined and <0>, <1> tags will be increased in their lower layer. It can be seen that the structure of this abstract way has destroyed the original structure of the object. The object cannot be restored after translation from the original structure. In this way, the translation of data accuracy will decrease, which should be avoided. The solution to this problem is to split and send the same name tag or manually rename them with different labels names. Through this way, a data structure of ordered list of values is changed to the data structure of "name/value" pair.

It can be seen from above that XML objects with no same labels in one layer and JSON objects in the structure of "name/value" pair can be translated to each other perfectly. In other forms, the translation can also be applied, but it will cause data mismatches.

III. TRANSLATION ALGORITHM AND EFFICIENCY

It can be seen from the above that XML and JSON data objects are multi-branch tree structure. It can be parsed and translated by recursive traversal of its tree parsing. Specific traversal algorithm is:

a. Get the root of the tree object
b. Get a value from the node of a, if the value is a number of child nodes, then traverse all child nodes, do operation a on all child trees whose roots are these child nodes, otherwise return the value of the node

By using the above algorithm, test the translation efficiency. The following data are tested using the PHP platform. All the data objects have a three-layer structure. The program runs only with a translation process and the page load operations, the page calls the implementation of the six outputs.

TABLE I. TRANSLATION SPEED TEST

Objects per page	Translation time/ms		Page load time/ms
	XML > JSON	*JSON > XML*	
1	6	2	49
3	14	7	57
5	27	9	65
8	37	14	82
10	50	20	120
100	370	180	580
1000	3433	1766	5230

As can be seen from table 1, considering the situation that running status of the server has an impact on page response and load time, with the recursive traversal algorithm of the translation, the translation time of XML and JSON linear correlation with the number of data objects. Compared to this page's simple feature, the total page load time translation spends is less than 50% occupancy rate. This low occupancy rate guarantees that the translation process will not take too many resources. And this ensures the efficiency of other parts of the program. In actual developments, serialized data objects usually have a flat tree structure. That is, no more than four layers. And the number of data requests sent in one page load is commonly less than 10. Therefore, in the situation of the actual development of complex network applications, the recursive traversal algorithm is practical.

IV. TRANSLATION IN WEB APPLICATIONS

A. Application in Web Service

Considering the standards of the services and security of Web Services, XML serialization is mostly used in this situation, especially in third-party Web Service. When the local Web Service calls third-party applications, XML objects are transmitted, but as for the local data interactive Web Service, in order to ensure the speed, developers will choose to use JSON serialization. In this case, the system architecture of the application should be added a translation system. This added system translates third-party data and then transmits the translated data into local post-processing applications, ensuring application of internal data flow. Application system is shown in Figure 3.

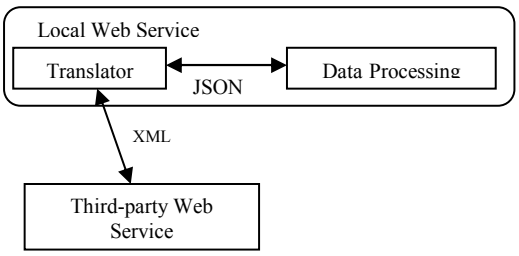

Figure 3. Application in Web Service

In this application mechanism, when it comes to the data translation problem, each processing function of the data processing layer of the local Web Service does not directly make the connection with third-party Web Service. Instead, a translation layer is added in the local Web Service, the data output by the data processing layer are translated integrated, and then transferred to a third party Web Service. Similarly, as for third-party Web Service, its output data are not connected with the data processing layer of local Web Service directly. Instead, translated by the translator layer and then sent to data processing layer. In this mechanism, data need to be translated are processed integrated, which ensures that the data processing layer can analyze and process data in a more efficient way.

B. Application in AJAX

Unlike Web Services, AJAX applications are aimed at enhancing the user experience so that data transmission speed is very important. JavaScript and JSON have good compatibility. The quick-analyzing feature of JSON will ensure efficiency in the application of AJAX technology. Compared to XML, JSON can greatly speed up the page speed of AJAX application, so it is promoted as a better alternative to large-scale use of XML. However, when using third-party applications through the AJAX technology, in order to ensure the specifications, safety, many third-party applications use XML for transmission. If third-party XML data is translated through the translation mechanism for local applications to JSON and then transferred as internal data, the user's browser AJAX running efficiency can be improved, and in this way, the user experience can be enhanced. Also, transferring data in the form of JSON instead of XML can speed up the server data transfer efficiency. Here the application of mechanism is shown in Figure 4.

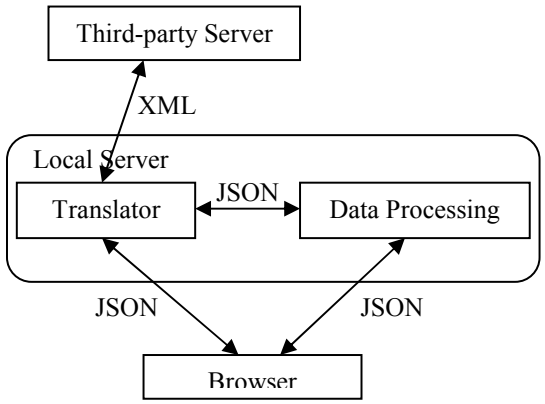

Figure 4. Application in AJAX

In AJAX application system, because the analyzing efficiency of the same data content of JSON by JavaScript is significantly higher than that of XML, local servers tend to interact with user browsers via JSON rather than XML to improve running efficiency. However, for the consideration of data security of the third-party server, when applying functions of a third-party server via AJAX, data transmission is provided in the XML serializing approach. Such data need to be translated before they are processed by local servers and user browsers. Through the translation layer, XML data provided by third-party servers are translated to JSON and then sent to background of local servers for data processing, or output directly to user browsers for analysis and display. Similarly, JSON-formed data requests sent by user browsers and local data processing layer are output to third-party server as XML objects via translation layer. In this way, ensuring the efficiency of JSON-based AJAX applications, secure connection with third-party servers can be applied through the translation mechanism.

V. CONCLUSION

XML and JSON serialization have been widely used in the actual development of web applications. These two methods have similar functions, but used in different application situations. Through the way of translation between these two data serializing approaches, these two forms of data can be interchangeable. This allows for better communication between web applications. By translating between each other, the data transmission of web applications can be secure, powerful in the XML serializing approach and fast and convenient via using the JSON serializing approach. In this way, the efficiency of web applications can be optimized, making applications more flexible and convenient. Web applications will have better extensibility and adaptability in the aspect of data transmission, Allowing for transferring with different third-party web applications. And the use of translator layer structure can ensure that the translation works will have less impact on the speed of a web application, making this translation between XML and JSON more efficient. Also, such independent translator layer will improve the program structure of an application and improve the utilization of codes.

REFERENCES

[1] Convert XML to JSON in PHP[EB/OL], http:// http://www.ibm.com/developerworks/xml/library/x-xml2jsonphp/.

[2] Introducing JSON[EB/OL], http://www.json.org

[3] SUN Baojun, Comparison and Research of JSON and XML[J], Inner Mongolia Science Technology & Economy, 2009, 12, P122-126(Ch)

[4] HU Xiaofeng, Application Analysis of JSON and XML on Networked Data Transmission[J], COMPUTER PROGRAMMING SKILLS & MAINTENANCE , 2010, 10, P77-78(Ch)

[5] CUI Can, NI Hong, Optimization Simulation of XML Performance Based on JSON[J], Communications Technology, 2009, 8, P108-114(Ch)

[6] WAN Hongli, Non-recursive Preorder Algorithm of SOAP Message Parsing[J], SOFTWARE ENGINEERING, 2009, 11, P52-P53(Ch)