

# The Application of XML Parsing Technology in E-Government

XUE Gang	ZHANG Li-li	YANG Ru-jin	LI Hao	YAO Shao-wen
Lab of Network Intelligent Computing, Yunnan University, Kunming ,China	Lab of Network Intelligent Computing, Yunnan University, Kunming , China	Yunnan Telecom NETIT Group, Kunming,China	Lab of Network Intelligent Computing, Yunnan University, Kunming , China	Lab of Network Intelligent Computing, Yunnan University, Kunming , China

## Abstract

*XML, which can be used in Internet directly and applied in various application programs, is widely used in E-government construction. The thesis discussed related technologies about XML and some kinks of XML parsing technologies that implemented by Java. Besides, it analyzed the format of the national E-document based on XML. Meanwhile SAX is suggested as the technology for E-document parsing tool, the realization and application of the tool were also represented.*

**Keywords:** XML parsing; E-government; E-document; SAX; DOM

## 1. Introduction

The informational construction of government, which uses E-government as its core, is the key of improving the informational process of our national economy. The construction of E-government refers to many ingredients. So, we can not copy ready-made criterions or technologies abroad. In the forepart construction of government information system, the lack of unified criterions often led to bad results such as obstructed system interconnect, poor sharing and hidden security troubles<sup>[1]</sup>. For these reasons, National Standardization Administration and Information Office established “National E-government Standardization Group”

and have constituted E-government related standard protocols<sup>[1]</sup>.

As a markup language which can express any kinds of data contents, XML have become one of the leading technologies. Many industries use XML to tailor to their own characteristic data exchange criterions. With the scale of our E-government construction and application enlarging, the format standard of XML E-document which has Chinese characteristic is certainly needed. *The XML Based E-document Format Criterions (preview edition)*, which will be abbreviated as “E-document Format Criterions”, is one of the six criterions<sup>[1]</sup> which were firstly made by National Standardization Administration.

As a pivotal technology, XML parsing is the precondition of XML application in E-government. According to National E-document Format Criterion, a special parsing tool is certainly needed. For users, this tool should shield the technology details and provide simple methods to operate documents.

There are many XML parsing technologies, so several technologies of them are analyzed here at first. Considering performance, SAX was chosen and introduced. In the section of “The Analysis of Use Case”, we introduced how to use it according to the needs of E-government application.

## 2. Related Technologies

### 2.1. XML Introduction

XML (eXtensible Markup Language) is a highly structural markup language. Developers can use XML to define tags and attributes. Its criterion<sup>0</sup> includes: XSL (eXtensible Style Language), XML Linking Language (XPath, Xlink, Xpointer included), XML Namespace, XML Schema etc. XSL implements XML style; XML Schema provides restrictions about XML documents. Besides, XML technology criterion includes other burgeoning technologies as well and these technologies are developing continuously.

### 2.2. XML Parsing

XML parsing technology, which is the bridge between application programs and XML documents, provides convenient measures to access and operate XML documents. At present, there are many XML parsing technologies. Among them, SAX, DOM, JAXP and JDOM are widely used.

**2.2.1. SAX (Simple API for XML):** The core interfaces of SAX are: ContentHandler, ErrorHandler, DTDHandler and EntityResolver. ContentHandler defines important methods in parsing time; ErrorHandler defines the methods to deal with errors; DTDHandler is used to deal with DTD documents; EntityResolver is used to deal with entities. When SAX starts to work, the creator class implemented by factory pattern<sup>0</sup> creates parser instance includes XML file reader. Different processors register in the reader. And the reader can also use interfaces to replace different processors.

While SAX working, a series of events are triggered and event handling functions are activated. Application programs can access to XML documents by using them. So, SAX interface is called event-driven interface. It checks the byte stream of XML documents sequentially, judges which section of XML grammar the current byte

belongs to. Then, corresponding events will be triggered.

**2.2.2. DOM (Document Object Model):** DOM is supported by W3C and used to create and process XML documents. It is an object-oriented standard interface which is not related to language and platform. It defines the logical structure of HTML and XML documents and provides the methods to access and operate elements. Developers can use DOM to create documents, read, add, delete and modify the content of the documents etc.

The principle of DOM is like that: DOM converts XML documents to object instance and organizes them as tree in memory. With the method provided by DOM, the document operation was done towards the objects in memory. When it began to organize output, DOM made the objects became sequential and wrote them into XML document.

The differences between SAX and DOM are: The sequential process model provided by SAX is not allowed stochastic access to XML documents; The transverse movement among elements is difficult. Compared with SAX, DOM consumes more resources and has a lower efficiency; however, DOM supports XML pattern better than SAX; and the implementation of DOM based on SAX.

**2.2.3. Other Parsing Technologies:** JDOM (JAVA Document Object Model) is similar to DOM. However, JDOM has made some optimizations. It is specialized for JAVA platform, not layered and driven by classes.

JAXP (Java API for XML Parsing) is an abstractive layer whose bottom layer includes SAX and DOM. It does not provide new XML parsing methods. It uses DOM and SAX application programming interfaces with a method which is not related to vendors.

## 3. XML-based E-document

The criterion of E-document format pointed out

that: XML-based E-document should satisfy basic needs of dealing with E-document in government<sup>[2]</sup>. Despite the main body, the composition should include information which records business process. The basic elements of E-document are: document body, document style, document transaction, document security, document exchange and filing<sup>[2]</sup>. On the premise of not changing the basic elements, it is possible to do some extension.

The explanation section of the criterion<sup>[2]</sup> made classification and analysis about the present documents. At the same time, it made structural description by using XML. The stipulation in the second section (official document body) of the criterion<sup>[2]</sup> is: the official document body consisted of two parts, the basic composing elements and the extensional composing elements. And former includes: the serial number, secrecy deadline, release number, receiver, title, sending department, text, attachment, signature of release department, written date, annotations, key words, printed edition logo and so on; the latter can be added in the body by departments according to their business needs.

The composing structure of official documents is like this: the root element is “E-document” (see DTD definition<sup>[2]</sup> in figure 1), below the root element, “sending department” and “official document body” are included.

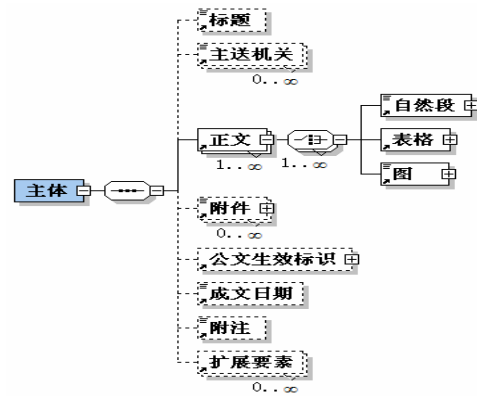
```

<?xml version="1.0" encoding="GB2312"?>
<ELEMENT 电子公文 (发文机关+, any*)>
<ATTLIST 电子公文
  xmlns CDATA #FIXED "http://www.egs.org.cn/eGovDoc"
  公文标识 CDATA #REQUIRED
  版本号 CDATA #FIXED "1.0"
  公文类别 CDATA #REQUIRED
  公文种类 CDATA #REQUIRED >
<ELEMENT 发文机关 (#PCDATA)>
<ATTLIST 发文机关 组织机构代码 CDATA #REQUIRED 办理类型 (主办 | 协办) #REQUIRED>
<ELEMENT 公文体 (眉首, 主体, 版记)>
<ATTLIST 公文体 xmlns CDATA #FIXED "http://www.egs.org.cn/eGovDoc/body">

```

**Figure 1 the XML DTD of “E-document” element**

The composing elements of the official document body are: “the page header”, “main body”, “edition logo”. They can include plenty of child elements (see figure 2):



**Figure 2 the XML structure of the element “main body”**

According to the analysis about the basic composing elements, we know that: 1. the elements in documents can be divided into these categories: the document elements without child elements and attributes, the document elements with son elements and attributes, the document elements with child elements but not child elements, the document elements with child elements but not attributes; 2. the relationship of elements is clear tree structure; 3. if document elements include textual information, it won’t include other child elements, and vice versa; 4. the same document elements will occur in different positions; 5. there are no unanalyzed entity information.

## 4.The Implementation of Parsing Tools

To national XML official documents, it is necessary to keep high efficiency of processing while catering the basic need. When designing parsing tools, we could deduce the attributes and methods in the classes of DOM or define new classes to express the information and use another tree structure to organize these class instances so as to take less resources and process documents more efficiently.

With the principle of DOM, we use SAX technology to analyze documents and organize the analysis results into specific objects. The concrete implementary method is: SAX is selected as the document analysis technology in the bottom layer;

and then, Java classes for document elements and attributes are define; tree model is used to organize these objects; relatively simple and convenient methods are provide to implement the operation to objects; supports which made objects serialized and written into XML documents are provide.

#### 4.1. The Definition of Document Information

The major information is: elements, attributes and entity information etc. According to analysis about the basic composing ingredients, we know: if document elements include text information, it won't include other child elements, and vice versa; there are no unanalyzed entity information in the basic composing ingredients. So, the modeling classes are:

- the class of element attributes (naming: ODAAttribute) includes attribute name, type, attribute information and provide simple method to get and set value;

- the class of attribute sets (naming: ODAAttributes) is used to manage multiple attribute objects included by a single document element and provides the operations of search, getting and modification to the objects in attribute sets;

- the class of document elements (naming: ODNNode) stipulates the indispensable processing methods. Document elements include child elements, the content information of elements and the attribute information. This class implements the management to attribute set etc.

Because the structure of XML documents is tree, object relationship should keep this structure so as to make the mapping easy while keeping simple object relationship. The Composite Pattern<sup>0</sup> is used to organize objects into tree structure to express the layer structure of "parts and whole".

#### 4.2. The Definition of Utility Classes

In order to unify operating method and make the management and usage of program easy, all class

files are put into a package. We select the Facade Pattern<sup>0</sup> to build an operating entrance class ODCController. It is responsible for invoking executive class of XML parsing and managing various needed resources during program runtime; for external, providing document parsing method, CRUD operating method and the method to write objects into XML files in serialization (convertToXMLFile). The class structure in the package is illustrated in figure 3.

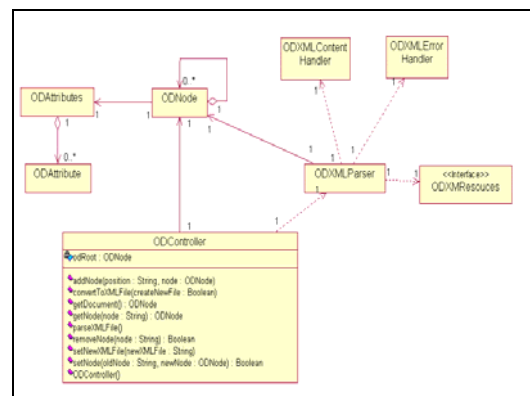


Figure 3 the UML structure diagram of parsing package

#### 4.3. The Implementation Analysis of Parsing Tools

Before using SAX to parse, document processor (ContentHandler interface implemented) and error handler (ErrorHandler interface implemented) should be defined and document reader (XMLReader) should be built. After reader respectively registers document processor and sets the mark of document parsing system, program starts to invoke bottom parser to parse XML documents. SAX analyzes XML documents and doesn't save the results. The document analyzing process is:

- When it meets the starting element, An ODNNode object and an ODAAttributes object should be initialized;

- The attributes included by XML elements should be parsed. According to the attribute information gotten by parsing, ODAAttribute object should be initialized, ODAAttribute object should be

set into ODAAttributes and then, ODAAttributes object should be set in ODNNode object;

-The processor of element content will set the content information into the attributes of the current ODNNode object, and then, put this ODNNode object into working stack;

-Using above-mentioned steps repeatedly to analyze other elements;

-When it meets the end element, get the element object out of the stack, judge the relationship between the current element object and other element objects and set the correct object relationship;

-When the analysis process ends, the root node (the object of the document root element) of object tree in memory should be saved as the entrance of the access to the whole object tree.

Basic methods provided by ODController class are: parserXMLFile (parses official document); getDocument (gets the root node object); removeNode (deletes the specific document element object); getNode (gets the specific element object); addNode (adds new element object); convertToXMLfile (writes the object tree into XML files). The principle of the parsing package is illustrated in figure 4.

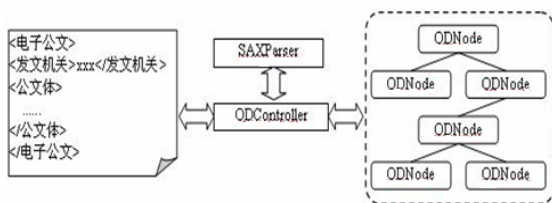


Figure 4 the principle of the parsing package

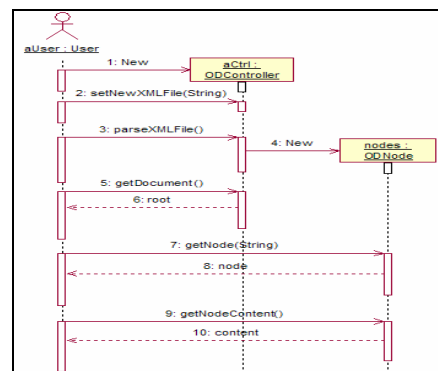
## 5. The Analysis of Use Case

Processing official documents is one of the most common business processing. Take document transmission as an example, the destination address (in the environment of network application, it is possible to establish the directory server to implement the mapping between the unit name and the address) of the document is acquired before it

should be transmitted. In order to get the information of “sending department”, application program invokes the parsing package. First, an ODController object is instantiated; the path of official document is set; then, parserXMLFile method is invoked to parse the document. During the process, various statuses can be fed back to users or application programs.

Because the same document elements will occur in different positions, destination locating string is used to locate destination node object. The writing rule of destination locating string is: specify the concrete path from the root node of object tree to destination node. By using this method, the accurate position of destination node is specified directly so as to enhance the efficiency of object searching.

After document parsing ends, getDocument method is invoked to get the root element. How to write destination locating string: “ 电子公文 . 公文文[0] . 主体[0] . 主送机关[0] ” (the number included in the string is the serial number of the needed object in the sibling nodes which have the same name, and if there is no sibling nodes with the same name, the serial number can be omitted). Destination locating string is passed into getNode method as a parameter, and then, the method returns an ODNNode object which includes the name of “sending department”. After invoking getNodeContent method, application program gets the concrete unit name. The sequence diagram is illustrated in figure 5. Then, application programs look up the corresponding address information in directory server and invoke the function of sending documents to send the documents to the destination address.



**Figure 5 the sequence of getting the element of  
“sending department“**

## 6. Summary and later work

After discussing XML and XML parsing technologies, SAX is selected as the parsing technology for E-document; Composite pattern is selected to organize element object as tree; the classes of design tools provide methods for document parsing. Through the introduction, we can conclude it is feasible to establish parsing tools for E-documents by using this method. This parsing tool not only shields the technology details for users, but also simple and convenient.

In the later work, we will strengthen the validation function of XML document in parsing tools and keep watching and researching the related criterions. Besides, we need go deep into the development of the application environment so as to make it easy to use in multiple platforms.

### Reference

[1] Standardization Administration of the People's Republic, Information Office of State Department, “the Related Criterions of E-government(preview edition)”,

Feb,2003

[2] Standardization Administration of the People's Republic, Information Office of State Department, “the XML Based E-document Format Criterions (preview edition)”, Feb,2003

[3] Standardization Administration of the People's Republic, Information Office of State Department, “the Application Guide of XML in E-government (preview edition)”, Feb,2003

[4] Brett McLaughlin, “Java & xml ,2nd Edition:Solutions to Real-World Problems”, O'Reilly, August 2001

[5] Erich Gamma et.al., “Design Patterns: Elements of Reusable Object-Oriented Software”, Addison-Wesley, 1995

[6] Ann Navarro et al., “Mastering XML”, BPB Publications, 2000

[7] W3c , “Extensible Markup Language” , <http://www.w3.org/XML/>

[8] W3c, “Extensible Markup Language (XML) 1.0 (Third Edition)”, W3C Recommendation 04 February 2004<http://www.w3.org/TR/2004/REC-xml-20040204/>

[9] Apache , “Xerces 2 API JavaDoc” , <http://xml.apache.org/xerces2-j/api.html>

[10] Apache , “Simple API for XML” , <http://xml.apache.org/xerces2-j/sax.html>

[11] W3c , “Document Object Model” , <http://www.w3.org/DOM/>