A Framework For Extracting Information From Web Using VTD-XML's XPath

C. Subhashini
Education & Research
Infosys
Mysore, India.
Subhashini_c@infosys.com

Dr.Arti Arya
Master of Computer Applications
PES School of Engineering, Bangalore
Bangalore, India
artiarya@pes.edu

Abstract — **The exponential growth of WWW (World Wide Web) is the cause for vast pool of information as well as several challenges posed by it, such as extracting potentially useful and unknown information from WWW. Many websites are built with HTML, because of its unstructured layout, it is difficult to obtain effective and precise data from web using HTML. The advent of XML (Extensible Markup Language) proposes a better solution to extract useful knowledge from WWW. Web Data Extraction based on XML Technology solves this problem because XML is a general purpose specification for exchanging data over the Web. In this paper, a framework is suggested to extract the data from the web. Here the semi-structured data in the web page is transformed into well-structured data using standard XML technologies and the new parsing technique called extended VTD-XML (Virtual Token Descriptor for XML) along with Xpath implementation has been used to extract data from the well-structured XML document.**

Keywords- *VTD-XML, Web Content Mining, Web Data extraction, Web Data Mining, XML, Xpath.*

I. INTRODUCTION

World Wide Web is the comprehensive information pool, but the data available on internet is unstructured or semi-structured and there is a need for extracting useful information from it. World Wide Web is too huge and structures of web pages are complex and it is tough to find the essential data and information. This poses a great challenge how to extract useful information from the web, mostly which is in the form of semi-structured. Moreover, extracting useful information from World Wide Web is necessary, which can lead to the best decision-making [1]. Web information is usually described in the form of HTML. Since it has unstructured layout and it is not suitable for database application [3], it is difficult to process HTML document for extracting data, so it is better to take the full advantages of XML for analyzing and processing the data on the web [2] and XML also separates data structure from layout which gives more suitable data representation [3]. In this paper, a framework is suggested to extract useful information from the web based on XML technologies.

The components of the proposed framework include data acquisition, data preprocessing, data conversion, data integration, data extraction and data storage. The paper is organized as follows: Section 2 provides the literature review of all possible data extraction based on XML, Section 3 focuses the related technologies for web data extraction; Section 4 gives the overview of the proposed framework. Section 5 presents the final conclusion and future scope of the proposed framework.

II. LITERATURE SURVEY

Web Data extraction is usually carried out in a documents that is made up of markup language such as HTML or XML. This document represents their inner structure. Web data extraction methods mainly focus on

the text representation or document tree structure. Web data extraction can be divided into two categories, i. Pattern matching, ii. Structure matching [3].

In Pattern matching document is accessed as a text and text based approaches (pattern matching, regular expression) are used to access the documents. It accesses only the individual lines of the document not the whole document.

Structure matching access the documents as a tree like structure and uses path and relation based approaches between the nodes. This approach access the document as a whole or individual sub-tree, which is in interest of extraction [3].

Many researches are available in the literature how extract useful information from the HTML page, which is in Semi-Structured format.

Yan Hu. et. al [16] have proposed a generic XML-based Web information extraction solution. This method proposes two key technologies: the XML-based Web data conversion technology which converts the HTML into XHTML document according to XML grammars, builds the XMLDOM tree and DOM-based XPath [18] generation algorithm is developed to generate XPath expression for the desired information nodes when the user marks the information points. Then XSLT template rules are applied to extract the user's interested information from the XHTML documents and the extraction results are expressed in XML.

Hanyang Luo et.al [17] has proposed a wrapper based on XBRL (eXtensible Business Reporting Language)-GL taxonomy to extract financial data from the web. In this, the user extracts information by using XPath as extraction rules and then the information collected are attached with tags using XBRL to generate XBRL instance document that enable the further data mining.

Siti Z.Z. Abidin et.al [4] has proposed a prototype tool to extract and classify unstructured data. The prototype architecture includes six important components such as Web - a collection of web pages, Generator that is used to request web services from the target web and also to retrieve data from the web, User Specifies input data to the generator and classifies the results of the data extraction , Converter that converts data from a XML documents to a Multimedia database or from a generator to XML document, XML document act as structured storage for data classification and Multimedia data bases stores different types of data like text, audio and video.

Cheng Zheng et. al [2] has used XML technology to convert the HTML pages into XML through XSL transformations[20]. An XSL transformation (XSLT) is a language for transforming XML documents to HTML or XHTML documents. Then these converted XML documents are integrated and then the data extracted from the integrated XML documents are stored to the database through Virtual Token Descriptor (VTD).

Jussi Myllymaki [5] has proposed ANDES (A Nifty Data Extraction System), a crawler-based web data extraction framework. It uses XML technologies such as XHTML, XSLT for data extraction and also provides deep web access. It extracts data from the targeted web page as well as navigational web page with the help of manual navigation and extraction rules.

Jussi Myllyamaki et. al [6] described a methodology for creating Xpath expressions to extract data virtually from any HTML page. They also specified categories of extraction rules based on their dependence on content, structural or formatting features.

In this paper a new extended edition of VTD-XML combining with 64-bit JVM(Java Virtual Machine), which supports XPath-based XML processing has been proposed to extract data from XML documents. This extended edition of VTD-XML makes possible to process giant XML documents (up to 256 GB) in size [10].

III. Related Technologies

A. Tidy

Tidy is a HTML syntax checker and pretty printer [14]. It is a freely available product and it corrects common mistakes in HTML documents and produces equivalent documents that are well-formed. Tidy can also be used to render these documents in XHTML (Extensible Hypertext Markup Language), a subset of XML [13].

B. XSLT, XPath and XML

XSLT (Extensible Style Sheet Language Transformation) provides the mechanism for converting one data structure into another. This is achieved by applying an XSLT style sheet to the XML document. The style sheet specifies the conversion rules for accessing and transforming the input XML document to a different output format. An XSLT processor is applies the rules defined in the style sheet to the input XML document [15].

The XML Path Language (XPath) is a standard for creating expressions and the expression can be used to find specific pieces of information within an XML document [15]. XPath uses path expressions to navigate in XML documents [18].

XML is the standard way for exchanging data over the Internet. The motive for choosing XML technology is that, it is most widely adopted technology for information representation and exchange on the WWW [7].

C. XML Parsing Theory Based On VTD-XML Model

VTD-XML is a new open source XML processing model. It centers on a "non-extractive" XML processing technique called "Virtual Token Descriptor" [8]. "Non-extractive" means, the original XML document is read into the memory in binary way and then analyzes the position of every element in this byte array and records some information; the followed traversal operation will be on these records. This VTD-XML is contrast to "**extractive**" parsing such as DOM, SAX and other old XML processing, which extracts part of the original document, and then creates objects in memory. DOM parses each event using these objects and builds the structure [9].

VTD-XML parses the XML document and creates 64-bit binary format VTD record (token) for each event. Through the list of VTD records, the application program may access any desired element. VTD-XML provides higher performance and requires lower resource compared with DOM (VTD-XML only need memory of about 1.3~1.5 times of the original XML document size, compared with DOM's 5~10 times of that) [11]. VTD-XML provides random access capability and also performs rapid analysis and traversal [9]. The extended edition of VTD-XML combines with 64-bit JVM (Java Virtual Machine) to support XPath-based XML processing to extract data from large XML documents (up to 256 GB) in size [10].

IV. PROPOSED FRAMEWORK

The algorithm used in the proposed framework is as follows.

WDE-XML (Web Data Extraction based on XML):

Step 1: Data Acquisition: Identify the data source.

Step 2: Data Preprocessing: Map it to XHTML through Tidy tool.

Step 3: Data Conversion: Find the reference points within the XHTML document and Map the data to XML through XSLT

Step 4: Data Integration: Merge the results through XSLT

Step 5:Data Extraction: Extract the data using VTD-XML and Xpath's implementation.

In this paper, only for the last step implementation is provided with the sample xml file.

The proposed framework is depicted in Fig 1.

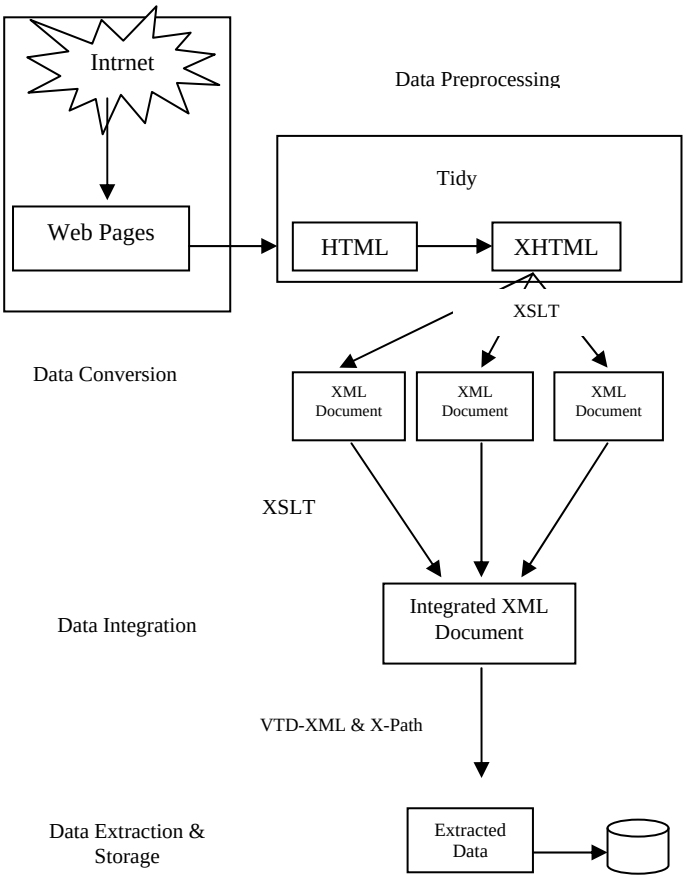

Figure. 1 An Overview of proposed framework

A. Data Acquisition

The first step is to obtain the web page for data extraction. The data source for extraction can be a data on local disk or data on the network [2].

B. Data Preprocessing

The web page in the HTML format is not well-formed because it does not conform to HTML specification. That is, in HTML ignoring closing tag will not give any error message. Therefore first it must be converted into well-structured XHTML format. Tidy is used to repair the broken syntax and produces well-formed XHTML.

C. Data Conversion

XSL (eXtensible Style Sheet language) is used to convert the XHTML to XML. The conversion is needed because of poor structural of XHTML documents. Even though it is based on the XML syntax structure, still it contains a lot of HTML vocabulary. So an XSL file has to be designed to convert the XHTML to XML. Next to extract information, it is necessary to find the reference point which contains the actual content.

D. Data Integration

Data integration allows the users to operate on data effectively. The web site may contain several pages and hyperlinks, so a merging method based on XSLT for several XML has to be designed. First, create a merge document and the sub element in the document states the name of the XML document to be merged. Next, define a style sheet for the corresponding XML document and then apply it to the merge document.

E. Data Extraction with extended VTD-XML and Xpath

Data extraction step extracts useful data form the integrated XML document. The technology of extended VTD-XML and XPath has been used for analyzing and processing the XML document. Because VTD-XML is the fastest and only XML parser that allows Xpath to process 256 GB XML document [20]. Using XPath, application can binds only the relevant data items, which avoids wasteful object creation. The XPath-based code can be understood easily and simple to write and debug [12]. But XPath can be applied to parsed tree of XML. So VTD-XML is used to build a tree-like table .

There is Java API for VTD-XML which is present at the top level and consists of three components.

- VTDGen (VTD generator) encapsulates the parsing routine that produces the internal parsed representation of XML.
- VTDNav (VTD navigator) is a cursor-based API that allows for DOM-like random access to the hierarchical structure of XML.
- Autopilot is the class that allows for document-order element traversal [19].

To use Extended VTD-XML and Xpath, application needs to include **com.ximpleware.extended** and need 64-bit JVM to take full advantage of extended VTD. The code for VTD-XML's Xpath implementation is shown Fig.2.

```
import com.ximpleware.extended.*;

public class Xpath
{
public static void main(String[] args)throws Exception
{
  VTDGenHuge vg=new VTDGenHuge();
  if(vg.parseFile("D:/Sample/input.xml",
            true,VTDGenHuge.MEM_MAPPED))
 {
  VTDNavHuge vnh=vg.getNav();
  AutoPilotHuge aph=new AutoPilotHuge(vnh);
  aph.selectXPath("/Employees/Employee/Empname/text()");
  int i=0;
  System.out.println("Employee Name");
  System.out.println("=============");
  while((i=aph.evalXPath())!=-1)
  {
   System.out.println(vnh.toString(i));
   }
  }
}}
```

Figure. 2 Code for VTD-XML Xpath's Implementation

V. CONCLUSION AND FUTURE SCOPE

This paper explains how to extract data from the largest source of information World Wide Web. World Wide Web is a huge source of unstructured information. The web pages in the HTML format is converted into XHTML using Tidy which further are processed using XSLT to form well formatted XML documents. Then the XML documents are integrated based on merge method using XSLT. Finally, extended VTD-XML and Xpath has been applied to integrated XML document to extract useful data. A new extended edition of VTD-XML with 64-bit JVM, which supports Xpath-based XML processing, is used. This extended edition of VTD-XML is able to process huge XML documents up to 256 GB in size. Future work includes systematically generating the Xpath expression, when the user marks the interested node, extracting the attributes present in the xml document and storing the extracted data into the databases, which may be analyzed for decision making purposes. This is an ongoing research wherein this technique can be used in actual web pages.

REFERENCES

[1] Li L. , Rong Q., "Research of Web Mining Technology based on XML, International Conference on Networks Security, Wireless Communications and Trusted Computing, 2009, pp.653-656

[2] Cheng Z., Yong F.Y.S. : The Implementation of the Web Mining based on XML technology, International Conference on Computational Intelligence and Security,2009 Page(s):84-87

[3] Rudy AG.Gultom., "Implemeting Web Data Extraction and Making Mashup with Xtractorz.

[4] Siti Z.Z. Abidin., " Extraction and Classification of Unstructured Data in WebPages for Structured Multimedia

[5] Database via XML.

[6] Jussi Myllymaki., "Effective Web Data Extraction with Standard XML Technologies.

[7] Jussi Myllymaki., " Robust Web Data Extraction with XML Path Expressions".

[8] Yasser K. ,Katsuhiko G. ,Web Mining Applications and Techniques:, Tokyo Institute of Technology, XML Semantics, pp:169-188.

[9] VTD-XML: XML Processing for the Future (Part I),http://www.codeproject.com/KB/cs/vtd-xml_examples.aspx

[10] Lan X. , Su J., Cai J. VTD-XML-based Design and Implementation of GML Parsing Project.

[11] VTD-XML ,http://en.wikipedia.org/wiki/VTD-XML

[12] Chee C., Faisal M.Y., Azhar K. M. :RBStreX: HardwareXML Parser for Embedded System.

[13] SCHEMA LESS C#-XML DATA BINDING WITH VTD-XML, **HTTP://WWW.CODEPROJECT.COM/KB/XML/SCHEMALESS_BINDING.ASPX**

[14] WEB-BASED DATA MINING, HTTP://WWW.IBM.COM/DEVELOPERWORKS/LIBRARY/WA-WBDM/JTIDY OPEN SOURCE SOFTWARE WRITTEN IN JAVA, HTTP://WWW.ROSEINDIA.NET/OPENSOURCE/OPENSOURCESOFTWARE.PHP?ID=407

[15] XML and Web services, Unleashed. Pearson Education.

[16] Yan H., Yanyan X., Research on Web Information Extraction Based on XML, In Second Intl. Conf. on Genetic and Evolutionary Computing, Sept. 2008, pp:401-404.

[17] Hanyang L., Jinling G., "Web Data Extraction Based on XBRL-GL Taxonomy," In Proc. of 2009 Asia-Pacific Conference on Information Processing, 2009, vol. 1, pp.358-361,

[18] http://www.w3schools.com/XPath/xpath_intro.asp.

[19] http://www.devx.com/xml/Article/22219/1954.

[20] http://vtd-xml.sourceforge.net/