IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 9, Issue 6 (Mar. - Apr. 2013), PP 74-79
www.iosrjournals.org

Web Data Extraction and Generating Mashup

Achala Sharma[1], Aishwarya Vaidyanathan[2], Ruma Das[3], Sushma Kumari[4]
[1](Computer Department, JSPM's BSIOTR(W)/ Pune University, India)
[2] (Computer Department, JSPM's BSIOTR(W)/ Pune University, India)
[3](Computer Department, JSPM's BSIOTR(W)/ Pune University, India)
[4](Computer Department, JSPM's BSIOTR(W)/ Pune University, India)

Abstract: *Web contains data that are assorted and are present in abundance. Various kinds of data can be easily extracted from the web, although not all of the data are relevant to the users. Maximum number of the web pages are in unstructured HTML format due to which problems arise in querying data sources making web data extraction process extremely time consuming and expensive. Therefore there arises a necessity to convert the unstructured HTML format into a new structured format such as XML or XHTML.*

To address this issue, we propose an approach for implementing web data extraction by extracting targeted data from various data sources and make Mashup by generating Extractor system. The Extractor system collaborates and integrates various stages of building a Mashup. Algorithms are used so that the Extractor system can specifically analyze the HTML tags and extract the data into a new format; however the core algorithm used, extracts data using recursive technique while rendering the DOM tree model automatically. Furthermore, the Mashup being created will help in the decision making process, which is the primary requirement for success in corporate world.

Keywords - *DOM tree, Extractor, HTML, Mashup, Web Data Extraction, XML.*

I. Introduction

Presently, various information's can be easily extracted through the Web. But the data that is extracted is not completely relevant to the user. Hence, it is necessary to have a web data extraction system which is capable of extracting the relevant data as per the users requirements. Existing web contents are mainly in unstructured HTML formats that are basically presentation-oriented. Unstructured HTML format is not suited for database applications. Also querying data in HTML contents incurs high cost and time [2].

To address these drawbacks we design a toolkit that implements web data extraction and design Mashup applications, called Extractor system. The software first extracts the desired data from targeted web pages with the help of Robots. Wrappers help in identifying the targeted data. The Document Object Model Tree commonly referred to as DOM Tree is constructed using HTML parser. This tree is used to perform comparisons between the newly generated path from the DOM tree and previously saved path (path that is traversed in the previously generated DOM tree). Further, the extracted information needs to be transformed as per user requirement which is done using Extractor system. The ultimate result is then sent to Database Management System or to Data Warehouses for further processing.

II. Literature Review

2.1 Typical Web Data Extraction System

A web data extraction system is a software system that automatically and repeatedly extracts data from web pages with continually changing contents and delivers the extracted data to a database or some other application.

Fig 1. depicts a high-level Architectural view of a typical web data extraction system [1].

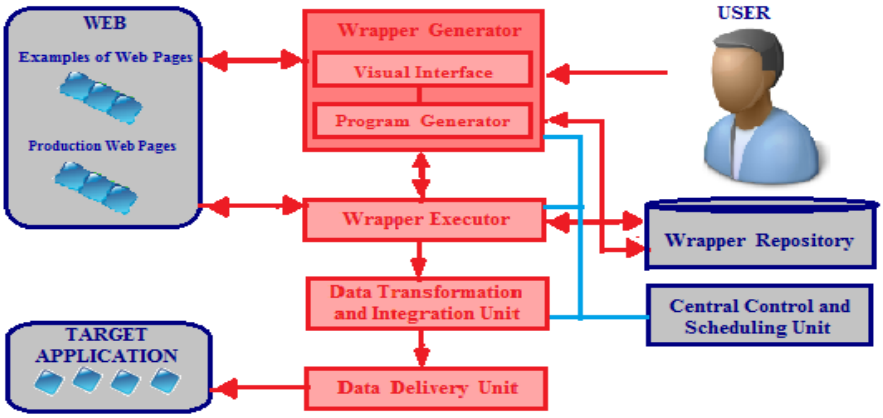

Figure 1. Architectural View of Typical Web Data Extraction System

This system comprises of various tightly connected components and three external entities namely the web, target application and user.

- The Web is the major source that contains web pages with information of interest.
- The Target Application is the one to which the extracted and refined data will be ultimately delivered.
- The User is the one who will interactively design the wrapper [1].

1.2 DOM Tree

The Document Object Model most often referred to as DOM is a cross-platform and language-independent convention for representing and interacting with objects in HTML and XML documents. It is an API for valid HTML and well formed XML documents. The DOM tree defines the logical structure of documents and the way a document is accessed and manipulated. It is constructed based on the organization of HTML structures (tags, elements, attributes). The HTML or XML DOM views a HTML or XML document as a tree-structure (node-tree). Every node can be accessed through the tree. Their contents can be modified or deleted. New elements can also be created [2].

In this paper, the basic approach of web data extraction process is implemented through the Document Object Model (DOM) tree. Using a DOM tree is an effective way to identify a list or extract data from the web page. Anything found in an HTML or XML document can be accessed, changed, deleted or added using the DOM tree. Fig 2. shows an Overview of the DOM Tree depicting the set of nodes that are connected to one another. The tree starts at the root node and branches out to the text nodes at the lowest level of the tree.

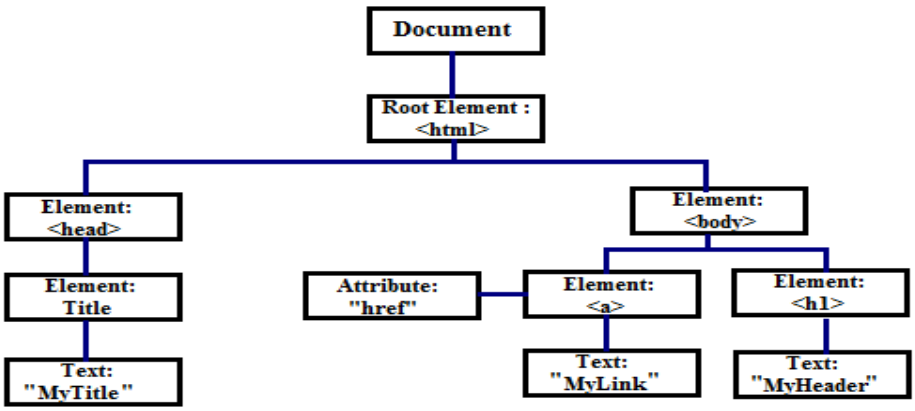

Figure 2. Overview of DOM tree

1.3 Mashup

Mashup in web development is a web page or web application that combines data, presentation or functionality from two or more sources to create new services and applications. The major characteristics of Mashup are combination, visualization and aggregation. It makes the existing data more useful for professional as well as personal uses. But the major problem is that creating Mashup requires a great deal of programming expertise in areas such as web crawling, text parsing, databases and HTML. Designing a Mashup needs to deal

with four basic issues of Data Retrieval, Data Source Modelling, Data Cleaning/ Filtering, Data Integration, Data Transformation.

III. Design and Architecture

Extractor is a working prototype that is designed to implement web data extraction and developing a Mashup. This system will ease the task of extracting the relevant data from various web pages for the user, without having programming skill. The Extractor system architecture is depicted in Fig 3. below.

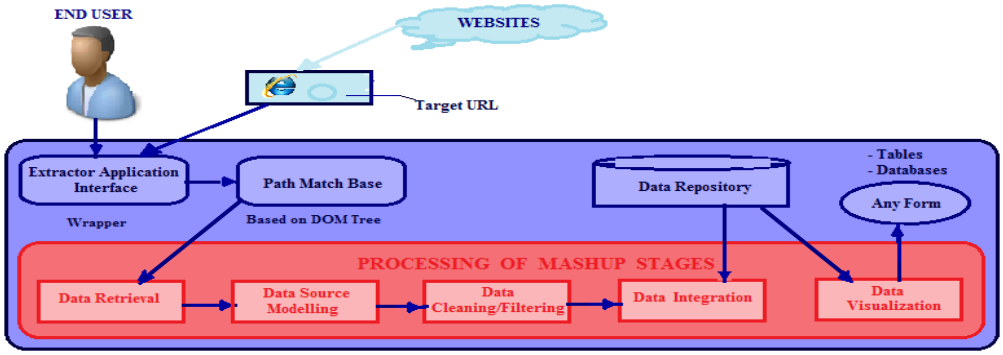

Figure 3. Extractor system architecture

3.1 Major Modules

The major modules includes the following :

1) Creation of Extractor : The Extractor creation involves the following steps,

- Specification of URL and Parameters :
  Specifying the URL and marking the data. It mainly navigates to the pre- targeted web pages containing the information of interest.

- Parsing and tree generation :
  Once we specify the required data (parameters) on the targeted web page, the robot will parse the page into tokens. The parser attempts to balance opening tags with ending tags to present the structure of the page. The output of the HTML parser is a Document Object Model tree.

- Locating data :
  The DOM tree generated is traversed and the search is performed using Depth First Search for locating the required data.

- Save the path :
  Once the data is located, the robot will save the path pattern that it took to reach the data in the tree. There after all the robot needs to do is to save the URL and the path it took initially to reach the data.

- Recursive search in targeted Web Pages :
  Since web data extraction is an automated process, when the data needs to be extracted from an updated web page, the robot will parse it again and traverse the same path that it had stored to reach the appropriate data. In order to do this it performs comparison between the two paths.

- Data filtering :
  The DOM tree can be filtered, unwanted tags and data that is not required to the user can be removed from the DOM tree and added as the filters.

The Fig 4. depicts the Creation of Extractor for the purpose web data extraction.

2) Manage Extractor : The previously generated Extractor can be loaded, saved or even modified to enhance reusability. Once the Extractors are loaded they can be executed to perform their intended task.

3) Extractor Execution : Comparison is performed between the newly generated path from the DOM tree and the previously stored path for extraction of the updated data from web page available.
4) Data Transformation : It includes transforming, refining, and integrating the extracted data from multiple sources and reforming the result into desired output format usually XML, relational tabular format or XLS. The Figure 5 depicts the transformation of extracted data into XML format.
5) Mashups : They are deliverable APIs or automated tools that can be used to extract required information using the Wrappers generated when Extractors are created. Wrapper is a program that identifies the desired data on target pages, extracts the data and transforms it into a structured format. Also referred as Wrapper for Extractor.
6) Deliverable Unit : The ultimate result in the form of structured data can be given to various external applications such as database management systems, data warehouses, business software systems, content management systems, decision support systems, RSS publishers, email servers, or SMS servers where they can be further used.

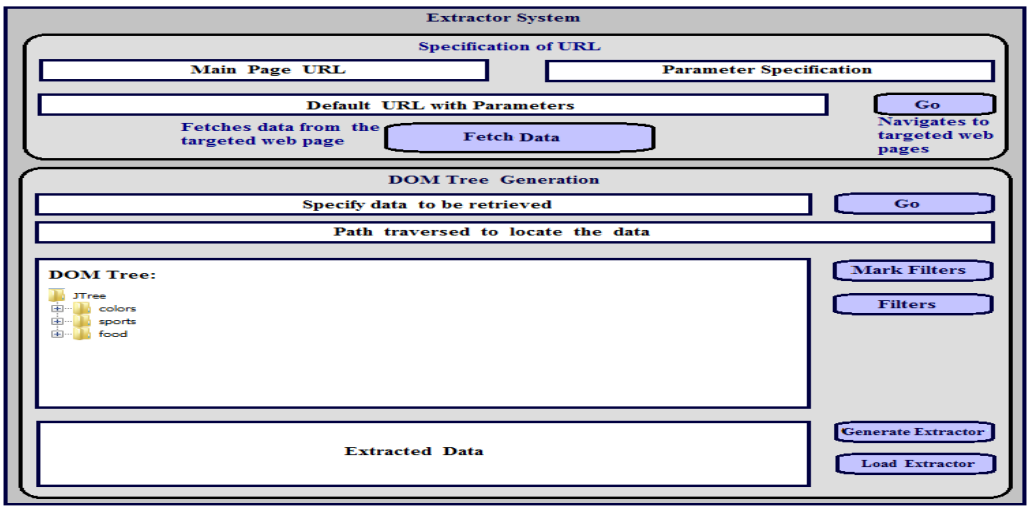

Figure 4. Design of Extractor creation.

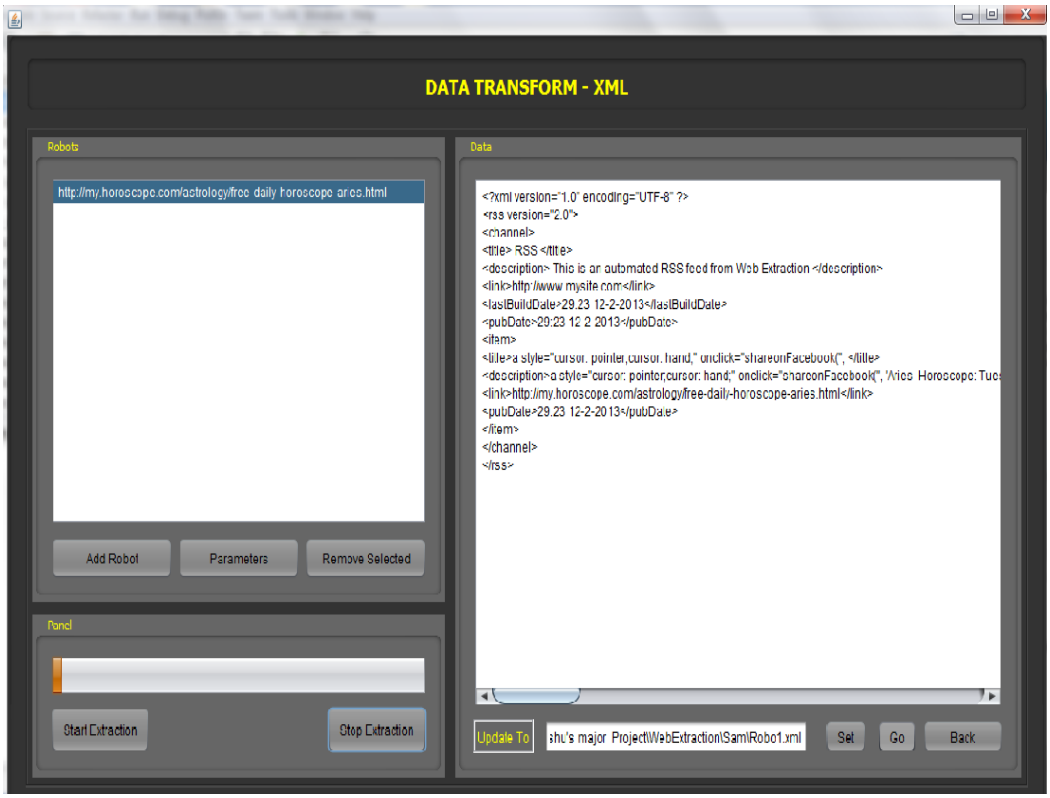

Figure 5. Transformation of extracted data into XML format.

IV. Performance Evaluation

The evaluation is performed by comparing our web data extraction system and making Mashup (Extractor system) with RoboMaker and Karma on the following basis:

- Number of steps required for building a Mashup
- Precision of extracted data

The reasons for choosing RoboMaker and Karma are because:

In RoboMaker, the user can create and debug robots of any kind, including data collection robots that extract objects from a web site as well as clipping robots that clip a part of an HTML page to be shown in another context, e.g. a portal. RoboMaker provides users with powerful programming features. RoboMaker's main focus is on data extraction from web sources and it outputs the result as an RSS feed. For operating on the RoboMaker, users must read tutorials and understand programming concepts, since it has a high learning curve [2].

In Karma, the evaluation is performed on the basis of an approach to data integration. Karma also makes an approach to build Mashup by combining the four information integration techniques, into a unified framework, where users can build a Mashup, without writing any code or understanding programming concepts [2].

On the contrary, Extractor system has capabilities for web data extraction and making a Mashup, by retrieving, modelling, cleaning and integrating extracted data. It performs transformation of the finally outputted data into a structured format as desired by the user. Furthermore, our system is designed to implement robot (wrapper) and completing Mashup stages much faster. No computer programming skills are needed at all in using this system.

Evaluation results for implementing web data extraction and making Mashup between RoboMaker, Karma and Extractor system can be depicted in Figure 6 where it shows the number of steps for each stages (retrieve, model, clean, integrate and visualize). The X axis maps Mashup stages and Y axis maps the number of steps.

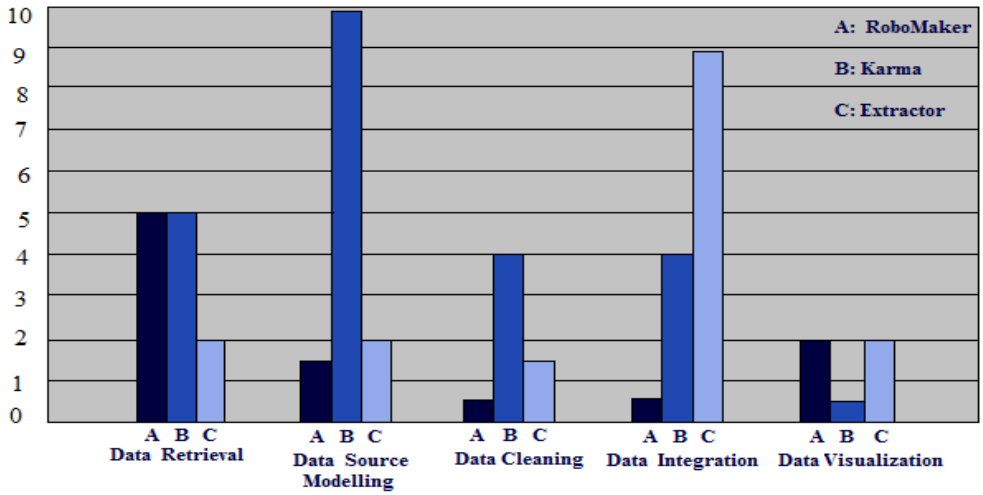

Figure 6. Evaluation of Result

V. Applications

The resulting data generated can be given as input to data mining tools, data warehouses, Email server or SMS server.

For example:

- The annual rainfall can be measured and compared with the existing database information for analysis of the water levels in dams, predict chances of flood and accordingly alarm the risks through SMS or even through e-mail to the respective authorities.
- Web market monitoring can be performed by legally retrieving information about competitors from public information sources on the web such as annual reports, press releases, public databases automatically by providing scalable environment and efficiently scheduling the processing of very large scale data sets.

- We can also generate new web applications and services from the existing and continually changing web pages.
- Stock market monitoring can be performed by retrieving information about the continuous changes in stocks automatically for the purpose of stock analysis and according send SMS or email to the stock holders which would help them in decision making.
- Mashup can be developed to implement web process integration where business critical data needs to be retrieved from various divisions such as quality management, marketing and sales, engineering, procurement and supply chain management from web portals thereby improving the workflow capabilities for the whole process of data extraction, transformation and delivery.
- Mashup can be useful in measuring the sea levels and predicting the areas where tsunamis and earthquake can occur to accordingly alarm about the dangers based on the historical data and generating patterns.
- Mashup can be useful in weather forecast or to alarm about a storm or heavy rains based on the historical data and patterns available.

VI. Future Work

In near future, we can plan to extend the capability of Extractor by combining web data extraction technique with Semantic Web approach. JQuery facility and the NLP (Natural Language Processing) can be incorporated with the features of Extractor GUI to further collaborate semantic meaning of the attributes which are extracted. Even in the area of communication, the possibility of aggregating and querying information automatically extracted from different Web news sites, especially combining with the features offered by XML-based query engines together with the Extractor system will certainly help paving the road to the semantic web. Moreover our system can be enhanced to extract images desired by the users in addition to the textual data.

VII. Conclusion

In this paper we propose a non-visual Extractor system which is able to extract data from various web sources continually by automating the entire web data extraction process. This system develops Mashup by generating automated robot (wrapper) and integrates the various phases in developing a Mashup into a single framework. Our approach includes the DOM tree generation, each time the web page is parsed and stores the path traversed to the targeted data. Since the path to the targeted data is being stored and not the entire generated DOM tree, it considerably reduces the required storage space. Extractor system allow the users to efficiently and effectively perform the task of web data extraction through an user interactive GUI, that doesn't require the user to have the knowledge of any programming techniques as well as without having to write any script.

References

[1] Robert Baumgartner , Wolfgang Gatterbauer, "Web Data Extraction", 2010.
[2] Rudy AG. Gultom, Riri Fitri Sari, "Implementing Web Data Extraction and Making Mashup with Xtractorz", 978-1-4244-4791-6/10/$25.00_c 2010 IEEE.
[3] Jer Lang Hong, Fariza Fauzi, "Tree Wrap-data Extraction Using Tree Matching Algorithm", February 2010.
[4] Majlesi Journal of Electrical Engineering Vol. 4, No. 2, June 2010- 43,"Tree Wrap-data Extraction Using Tree Matching Algorithm.