ISSN No. 0976-5697

Volume 2, No. 4, July-August 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

SOA: A Service – Oriented Technique for Deep Web Data Extraction

Dr.Ch. G. V. N. Prasad*, P. Satyavathi, S. Swathi
SICET, Hyderabad
prasadch204@gmail.com
satya.potlapalli@gmail.com
swathibhouni@gmail.com

Abstract- World Wide Web is a largest source of information, due to its inherent dynamic characteristics, the task of finding useful and qualified information can become a very frustrating experience. This paper describes the detailed information on web mining databases and the study of wrapper generation algorithm and deep web contents are accessed by queries submitted to Web databases and the returned data records are enwrapped in dynamically generated Web pages (deep Web pages). A research on the Visualization of web page in Xml format and also the information mining systems in the Web and proposes an implementation of the service oriented architecture components that can be built using the technology of Web services.

Keywords- Web databases, text mining, wrapper generation, deep web pages.

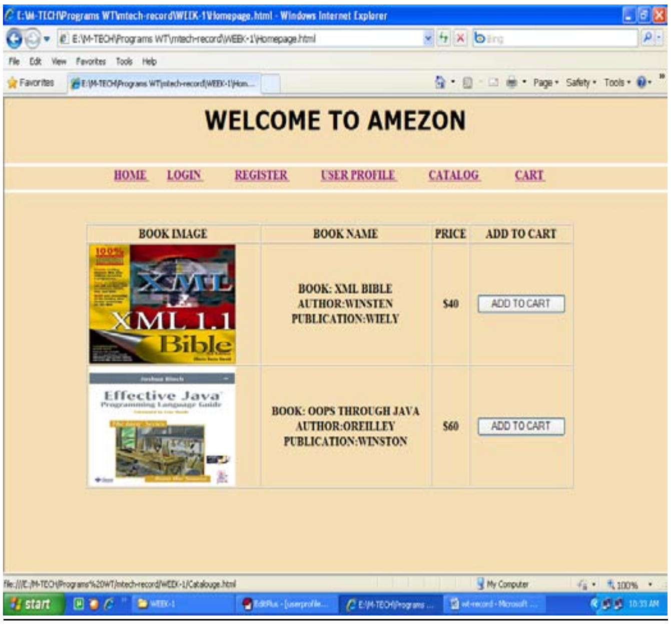
I. INTRODUCTION

With the increasing popularity of the internet through World Wide Web access to more data and information, work, study of the great changes taking place, much higher efficiency, resources of information are the greatest degree of sharing. Web mining technology and data mining is combination of web technology, is an integrated resources extracted from WWW information of the course is the implication of web resources interest, un known the potential value of the mode of extraction it repeated use of a variety of data mining algorithms from the observation data to identify patterns are a reasonable model but also to data mining technology and application of the theory of world wide web resources to carry out excavation of new research field.

In the recent explosive growth of the amount of content on the Internet, it has become increasingly difficult for users to find and utilize information and for content providers to classify and catalog documents. Traditional web search engines often return hundreds or thousands of results for a search, which is time consuming for users to browse. On-line libraries, search engines, and other large document repositories *(e.g.* customer support databases, product specification databases, press release archives, news story archives, *etc.)* are growing so rapidly that it is difficult and costly to categorize every document manually. In order to deal with these problems, look toward automated methods of working with web documents so that they can be more easily browsed, organized, and cataloged with minimal human intervention.

In contrast to the highly structured tabular data upon the machine learning methods are expected to operate, web and text documents are semi-structured. Web documents have well-defined structures such as letters, words, sentences, paragraphs, sections, punctuation marks, HTML tags, and so forth. We know that words make up sentences, sentences make up paragraphs, and so on, but many of the rules governing the order in which the various elements are allowed to appear are vague or ill-defined and can vary dramatically between documents. It is estimated that as much as 85% of all digital business information, most of it web-related, is stored in non-structured formats *(i e .* non-tabular formats, such as those that are used in databases and spreadsheets) [pet]. Developing improved methods of performing machine learning techniques on this vast amount of non-tabular, semi-structured web data is therefore highly desirable.

World Wide web has more online web databases, which can be searched through their web query interfaces. The number of web databases has reached 25 millions.[21]. All the web databases make up the deep web (hidden web or invisible web). Often the retrieved information (query results) is enwrapped in web pages in the form of data records. These special web pages are generated dynamically and are hard to index by traditional crawler-based search engines, such as Google and Yahoo. Fig. 1 shows a typical developed deep web page from Amazon.com. On this page, the books are presented in the form of data records, and each data record contains some data items such as title, author, etc. In order to ease the consumption by human users, most web databases display data records and data items regularly on web browsers. However, to make the data records and data items in them machine process able, which is needed in many applications such as deep web crawling and meta searching, the structured data need to be extracted from the deep web pages. In this paper, we study the problem of automatically extracting the structured data, including data records and data items, from the deep web pages.


E:\M-TECH\Programs WT\mtech-record\WEEK-1\Homepage.html - Windows Internet Explorer
E:\M-TECH\Programs WT\mtech-record\WEEK-1\Homepage.html
File Edit View Favorites Tools Help
Favorites
Page Safety Tools
file:///E:/M-TECH/Programs%20WT/mtech-record/WEEK-1/Catalouge.html
My Computer
100%
start
WEEK-1
EditPlus - [userprofile...
E:\M-TECH\Programs ...
wt-record - Microsoft ...
10:33 AM

WELCOME TO AMEZON

HOME LOGIN REGISTER USER PROFILE CATALOG CART

BOOK IMAGE	BOOK NAME	PRICE	ADD TO CART
	BOOK: XML BIBLE AUTHOR:WINSTEN PUBLICATION:WIELY	$40	ADD TO CART
	BOOK: OOPS THROUGH JAVA AUTHOR:OREILLEY PUBLICATION:WINSTON	$60	ADD TO CART

Figure:1 Deep Web Page from Amezon.com

The problem of web data extraction has received a lot of attention in recent years and most of the proposed solutions are based on analyzing the HTML source code or the tag trees of the web pages. These solutions have the following main limitations: First, they are web page programming language dependent, or more precisely, HTML dependent. As most web pages are written in HTML, it is not surprising that all previous solutions are based on analyzing the HTML source code of web pages. However, HTML itself is still evolving (from version 2.0 to the current version 4.01, and version 5.0 is being drafted [14]) and when new versions or new tags are introduced, the previous works will have to be amended repeatedly to adapt to new versions or new tags. Furthermore, HTML is no longer the exclusive web page programming language, and other languages have been introduced, such as XHTML and XML (combined with XSLT and CSS). The previous solutions now face the following dilemma: should they be significantly revised or even abandoned? Or should other approaches be proposed to accommodate the new languages? Second, they are incapable of handling the ever-increasing complexity of HTML source code of web pages. Most previous works have not considered the scripts, such as JavaScript and CSS, in the HTML files. In order to make web pages vivid and colorful, web page designers are using more and more complex JavaScript and CSS. Based on our observation from a large number of real web pages, especially deep web pages, the underlying structure of current web pages is more complicated than ever and is far different from their layouts on web browsers. The makes it more difficult for existing solutions to infer the regularity of the structure of web pages by only analyzing the tag structures.

II. SECTION

A. *Web Mining*

Web is a collection of inter-related files on more web servers; web mining is the application of data mining techniques to extract knowledge from web data. Web mining is classified into the following

B. *Classification of Web Mining*

a. Web Content Mining :

Web Content Mining is the process of extracting knowledge information from the contents of Web documents. Content data corresponds to the collection of facts a Web page was designed to convey to the users. It may consist of text, images, audio, video, or structured records such as lists and tables. Activities used in this Research techniques from other disciplines such as Information Retrieval (IR) and Natural Language Processing (NLP).

Identify the topics represented by a Web Documents Categorize Web Documents Find Web Pages across different servers that are similar Applications related to relevance Queries –Enhance standard Query Relevance with User, Role, and/or Task Based Relevance Recommendations –List of top "n" relevant documents in a collection or portion of a collection.

Filters –Show/Hide documents based on relevance score

b. Web Structure Mining:

The structure of a typical Web graph consists of Web pages as nodes, and hyperlinks as edges connecting between two related pages. In addition, the content within a Web page can also be organized in a tree structured format, based on the various HTML and XML tags within the page. Thus, Web Structure Mining can be regarded as the process of discovering structure information from the Web. This type of mining can be performed either at the (intra-page) document level or at the (inter-page) hyperlink level (Figure 2.1).

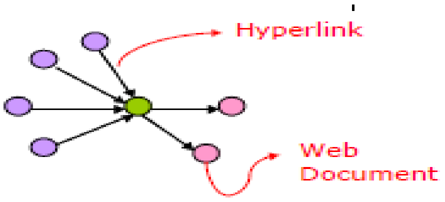


Figure 2.1. Shows the process of web structure mining

Web Structure is a useful source for extracting information such as Quality of Web Page
-The authority of a page on a topic
-Ranking of web pages
Interesting Web Structures
-Graph patterns like Co-citation, Social choice, Complete bipartite graphs, etc.
Web Page Classification
-Classifying web pages according to various topics
Which pages to crawl
-Deciding which web pages to add to the collection of web pages
Finding Related Pages
-Given one relevant page, find all related pages
Detection of duplicated pages
-Detection of neared-mirror sites to eliminate duplication

c. *Web Usage Mining*:

Web Usage Mining is the application of data mining techniques to discover interesting usage patterns from Web data, in order to understand and better serve the needs of Web based applications. Usage data captures the identity or origin of Web users along with their browsing behavior at a Web site. Some of the typical usage data collected at a Web site include IP addresses, page references, and access time of the users.

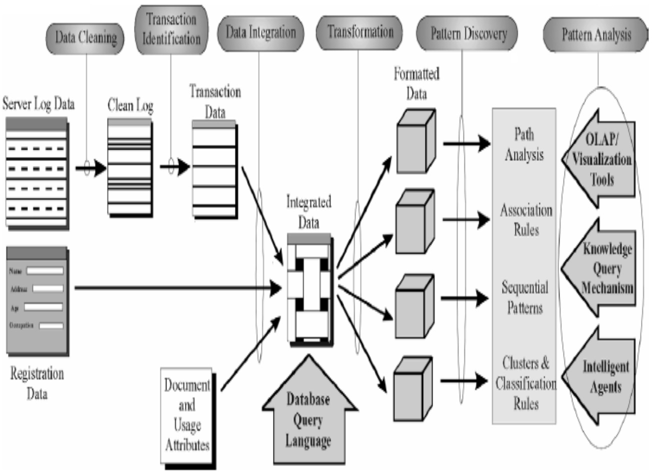


Figure 2.2. Shows the architecture of web usage mining

d. Text Mining :

Due to the continuous growth of the volumes of text data, automatic extraction of implicit previously unknown and potentially useful information becomes more necessary to properly utilize this vast source of knowledge. Text mining, therefore, corresponds to extension of the data mining approach to textual data and its concerned with various tasks, such as extraction of information implicitly contained in collection of documents or similarity- based structuring. Text collection in general, lacks the imposed structure of a traditional database. The text expresses the vast range of information, but encodes the information in a form that is difficult to decipher automatically.

C. Web Data Mining Process using XML Technology

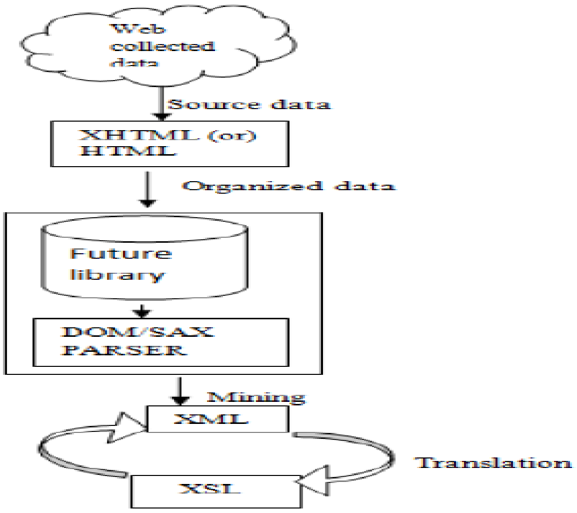


Figure 2.3. Shows the web page is in XML format

Fig shows the Web page is into XML format, and use the tools to deal with the structure of XML data in order to extract the appropriate data. HTML files can be used to correct common error in the layout and format to generate the equivalent of a good document, we can used to generate XHTML (XML subset of) the format of the document. By constructing a XML Helper to complete the Java type data from XML to HTML conversion, as well as with other XML-related tasks.

The main steps are as follows:

a. Recognize the source of data and map it into XHTML (Or) HTML. In most cases, the source of informationis obvious, but in a dynamic environment to be extracted for use, reliable and stable sources ofinformation more difficult. To determine the source of information, through the structure, called the Java class of XML Helper to complete the data from XML to HTML conversion.

b. To find the data points used. Both the Web page and XHTML source in view of the vast majority of information has nothing to do with the information collected, the next in the XML tree to find a specific region, the need to extract the data. We find the data generally contains the same elements <table> this table will contain general information required for key words, the note observed, the analysis of the page generated XHTML, and the table as Reference points, or anchor.

c. Data will be mapped into XML. You can create data taken from the actual codes when you find the anchor, the code will be Extensible Style sheet Language file the form.XSL document is intended to anchor logo, and specify how to get from the anchor is looking for data, and by that we needed to construct an XML format output files.

d. Joint results of the data. If only the implementation of a data extraction, in accordance with the abovementioned steps have been completed. However, Web data mining is a week of back and forth, a few simple data collected has not yet completed the task of data mining. Web data mining for the special, it is necessary to keep the Internet on the collected data and the results into XML data files.

III. SECTIOON

A. Web Mining Techniques

Traditional data mining techniques can also be used for web mining, such as classification, clustering, association rule mining, and visualization. In web mining, classification algorithms can be used to classify users into different classes according to their browsing behavior, for example according to their browsing time. After classification, a useful classification rule like "30% of users browse product/food during the hours 8:00-10:00 PM" can be discovered. The difference between classification and clustering is that the classes in classification are predefined (supervised), but in clustering are not predefined (unsupervised). The criterion by which items are assigned to different clusters is the degree of similarity among them. The main purpose of Clustering is to maximize both the similarity of the items in a cluster and the difference between clusters [2].The association rule technique can be used to indicate pages that are most often referenced together and to discover the direct or indirect relationships between web pages in users' browsing behavior [1]. For example, an association rule in the web usage mining area could take the form "the people who view web page index.htm and also view product.htm the support=50% and the confidence=60%". Visualization is a special analytical technique in web mining that allows data and information to be understood or recognized by human eyes by using graphical and visualized means to represent data, information and analysis results [3].In web structure mining, it usually plays an important role in illustrating the structure of hypertexts and links in a website or the linking relationship between websites. For the other two types of web mining technique, visualization is also an ideal tool to model the data or information. For example, a graph (or map) can be used for web usage mining to present the traversal paths of users or a graph may show information about web usage. This approach enables the analyst to understand and efficiently interpret the results of web usage mining.

B. Processing steps for Web Mining

The techniques being applied to Web content mining draw heavily from the work on information retrieval, databases, intelligent agents, etc. Most of these techniques are well known and reported elsewhere, hence in this survey we have not focused on Web content mining. Hence, in this survey we have focused on Web Usage Mining, which is just starting as an area of research, and hence has a number of open issues. In the following we provide some directions for future research:

***a. Mining Process* :**

The key component of Web mining is the mining process itself. As discussed in this paper, Web mining has adapted techniques from the field of data mining, databases, and information retrieval, as well as developing some techniques of its own, e.g. path analysis. A lot of work still remains to be done in adapting known mining techniques as well as developing new ones. Specifically, the following issues must be addressed:

***b. Knowledge to be Mined*:**

Web usage mining studies reported to date have mined for association rules, temporal sequences, clusters, and path expressions. As the manner in which the Web is used continues to expand, there is a continual need to figure out new kinds of knowledge about user behavior that needs to be mined for.

***c. Mining Algorithms*:**

The quality of a mining algorithm can be measured both in terms of how effective it is in mining for knowledge and how efficient it is in computational terms. There will always be a need to improve the performance of mining algorithms along with these dimensions.

d. Data Usage on Web Mining:

Usage data collection on the Web is incremental in nature. Hence, there is a need to develop mining algorithms that take as input the existing data mined from various logs can be integrated together into a more comprehensive model.

C. Data Extraction

HTML Target pages are subjected to a sequence of data extraction. Much of the HTML content on the web is ill-defined because it does not conform to HTML specifications. Therefore, the first step in data extraction is to translate the content to a well-formed XML syntax because this helps in subsequent data extraction steps. The specific approach taken in the ANDES framework is to pass the original HTML page through a filter that "repairs" the broken syntax and produces well-formed HTML, or what is today known as Extensible HTML (XHTML) [5]. Toolkits for this step exist already, including the Tidy package [4]. Since XHTML is based on XML, any XML tool can be used to further process target HTML pages. Given that the goal of ANDES(web data extraction framework) is to produce XML as output, we view the task of converting XHTML to XML as an XML transformation problem. The data transformation mechanism chosen for ANDES is Extensible Style sheet Language Transformations (XSLT) [8], a language that provides powerful XML path expressions (XPath) [7] combined with regular expressions through the XSLT extension mechanism.

As shown in Figure 2, the URL of an XHTML document is used to determine which set of XSLT files to apply to it. The XHMTL document is passed through the first XSLT file and the output is pipelined through other XSLT files defined for that URL. The final output is an XML file whose structure and content is determined by the last XSLT file. This is typically an XML application2, for instance iCalendar XML or NewsML. The pipeline approach fits well with the goals of domain-specific ANDES applications; the first XSLT file merely extracts data from an XHTML page, while subsequent XSLT files in the pipeline can refine the data and fill in missing data from domain knowledge. The main criticism directed towards HTML data extraction projects is that the approach essentially amounts to "screen scraping" and fails miserably when the design (structure and content) of a Web site changes. While total isolation from these changes is difficult to achieve, we believe the ANDES approach is solid and produces very robust wrappers. This is achieved by relying less on HTML structure and more on content. Some wrapper languages (e.g. HTML Extraction Language in W4F) require the use of absolute HTML paths that point to the data item to be extracted. An absolute path describes the navigation down an HTML tree, starting from the top of the tree (<HTML> tag) and proceeding towards child nodes that contain the data to be extracted. The path is made absolute by the fact that

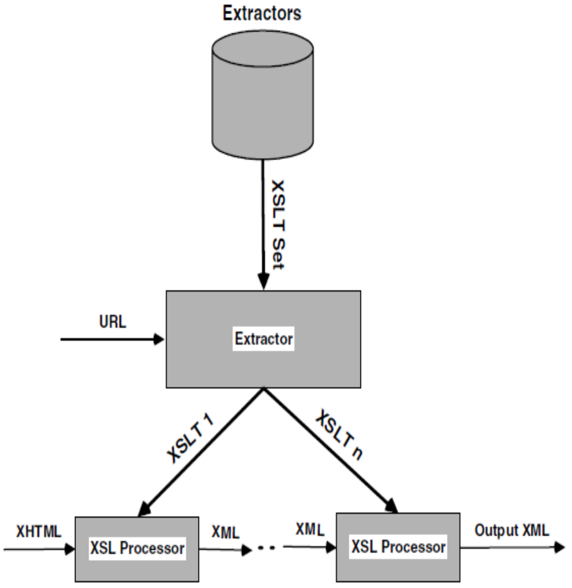


Figure 3.1. Extractor identifies XSLT files to be used. A pipeline of XSLT processors extract and refine data, yielding an XML application file as the ultimate output.

It lists tag names expected to be seen in the tree and their absolute positions. For instance, an absolute path to the third table, first row, and second column in an HTML document could be expressed in XPath as /HTML/BODY/TABLE [3]/TR TD[9]. The absolute path approach is likely to fail when the target HTML page changes. The most common change in HTML design is changing the positioning of items on the page. Layout is typically performed by using tags like <TABLE>, <TR>, and <TD>, as seen in the example above. When new content (e.g. advertising) is added to a page or when existing content is moved around, the absolute location of tags changes. For this reason, it is important to establish the location of data items independently of their absolute paths.

Our approach involves finding anchors within the page that serve as starting points for data extraction. Ideally, anchors are established based on the content of a data item, not on its HTML path. For instance, a page that contains the price of a book probably has the word “Price” somewhere near the price value. By looking for the word “Price,” we can establish an anchor for the price value and be independent of its absolute location. An example of XSLT code that extracts the last stock quote from a Yahoo! Finance page is shown in Figure 3. Note that we look for a table cell containing the words “Last Trade” and extract the value contained in the B (bold) tag. The XSL processor starts from the root of the XHTML tree and recursively looks for a matching table cell. Once the table cell is found, the instructions contained in the template are executed, in this case the production of a PRICE element in the output XML document.

D. Process of Deep Web Data Integration & Extraction

In deep web data integration process two continuously executed phases can be distinguished System preparation is creation and continuous update of resources required for usage of system.

System usage is actually lazy or eager data integration from dispersed sources.

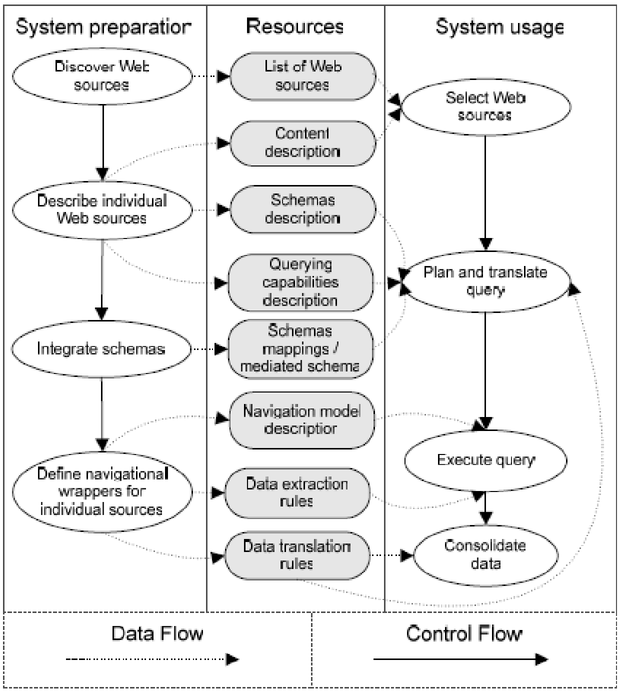


Figure 3.2. Deep Web data extraction and integration process

During system preparation phase, first the sources need to be discovered. This step is currently assumed to be done by human beings in the list of sources is usually provided to the integration system. Subsequently the sources need to be described in their schema, query capabilities, quality, coverage needs to be provided for further steps in the integration process. Most importantly schema descriptions are used in the next step to map them either one to the other or to some global schema. The last part of system preparation is a definition of navigation and extraction rules that allow the system to get to the data through a source’s Web interface. These resources are used once query needs to be answered: sources to ask are selected based on their subject and coverage, the plan is generated using schema mappings and source query capabilities and the query is executed through a Web interface using navigation model for particular queries and data extraction rules defined for them. Once data are extracted from HTML pages, they need to be translated and cleaned according to data translation rules. The overall process includes all the activities necessary to integrate data from Web databases identified so far in the literature, and adds few steps (discovery of web sources, declarative description of navigation model) that were neglected so far. As such it can serve as a basis for generic data integration system.

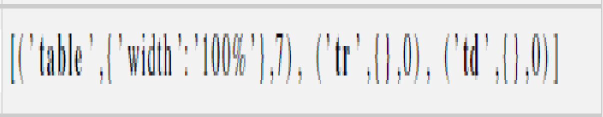
E. Wrapper Generation

The automatic generation of new wrappers is necessary so rules can be easily maintained while reducing to a minimum the amount of manual work to be done by the user. The generation process requires examples, which in this context will consist of tuples with the value that has to be extracted and the URL of the web page containing it. For highly-structured pages the minimum number of examples needed to build good-enough rules is as low as one or two. On the other hand, if the desired information is mixed with other text, more noise has to be omitted and having a greater number of them is extremely useful. To ease the task of collecting examples, references that have been previously extracted are stored along with the source URL so they can also be used when generating new wrappers. As before, highly simplified pseudo code for wrapper generation is shown in listing 2. Basically, rules are built by following these steps:

a. Create rule for one example.

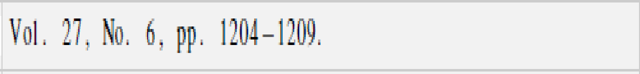
b. Merge it with the previous rules. If not possible, create new rule.

c. Repeat while there are more examples available. Specific details vary for each of the implemented type of rules, which we describe next in more detail. Path Rules define the location of an element by specifying its path in the HTML tree. They are represented by a list of triplets with the element name, attributes with their corresponding values and, finally, the sibling number. These rules are created bottom-up and stopping once The first element of the path is unambiguous. Example:



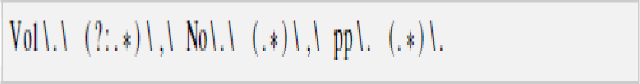
```
[('table',{'width':'100%'},7), ('tr',{},0), ('td',{},0)]
```

Two paths are only merged if they have the same length, element names and attributes, i.e. the only thing that varies is the sibling number. If two paths cannot be merged, different wrappers are then created. When applying this kind of rules, if more than one element is matched, all values are evaluated before choosing. As one may expect, we have found that CSS class names are especially useful for distinguishing among different fields. Regex Rules are a bit more interesting. They define the location of a piece of information with in some text, which is necessary when the same HTML element not only contains the desired value but some other information as well. For example, getting the volume number from a text that contains other fields



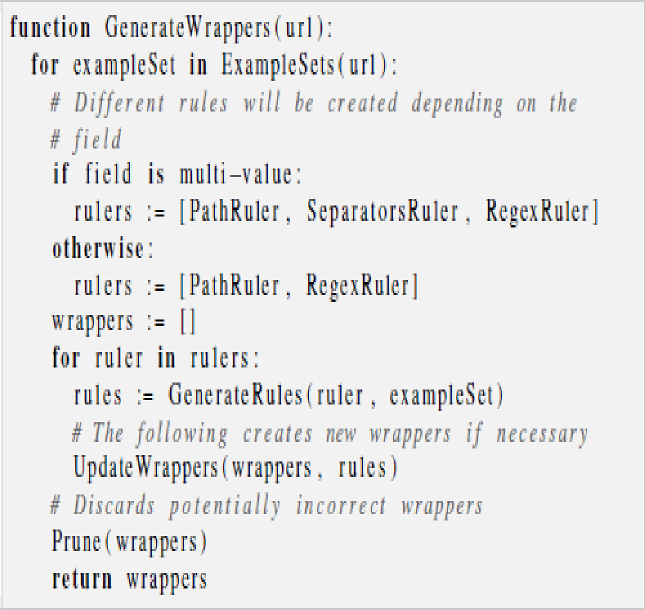
```
Vol. 27, No. 6, pp. 1204-1209.
```

would require an expression like 5



```
Vol\.\ (?:.*)\,\ No\.\ (.*)\,\ pp\. (.*)\.
```

This is obviously not the most preferable regular expression that would allow us to get the volume number, but it is easy to generate it from a set of sufficiently different examples. The more different the examples, the more general the final rule will be. These kinds of expressions are created by stripping the value to be extracted from the text and comparing it to other examples. The comparison is done with Python Difflib module, which internally uses, Ratcliff and Metzener (1988). If texts have a similarity greater than a given threshold, then they are merged by generalizing the substrings on which they disagree. The first approach for creating these rules was similar to the one used in WHISK, Soderland (1999), that is, removing the value and checking the text on the sides. This was later abandoned because these characters differed considerably among pages from the same library. Looking at it in perspective, it could be interesting to combine this strategy with the current one.



```
function GenerateWrappers(url):
  for exampleSet in ExampleSets(url):
    # Different rules will be created depending on the
    # field
    if field is multi-value:
      rulers := [PathRuler, SeparatorsRuler, RegexRuler]
    otherwise:
      rulers := [PathRuler, RegexRuler]
    wrappers := []
    for ruler in rulers:
      rules := GenerateRules(ruler, exampleSet)
      # The following creates new wrappers if necessary
      UpdateWrappers(wrappers, rules)
    # Discards potentially incorrect wrappers
    Prune(wrappers)
    return wrappers
```

Algorithm of wrapper Generation

IV. SECTION

A. Extract Data from Service-Oriented Architecture

We can extract multi-valued fields that are located in different HTML elements but that have a similar path (e.g. siblings or cousins). However, there is another situation that should be considered. Separator Rules can be used when the multiple values of a field are located within the same HTML element. As their name suggests, these rules consist of a list of separator strings usually ", " and "and ". Given an example, they are created by removing the values to be extracted and keeping the substrings in-between. Merging is done by joining lists. For the specific problem of reference extraction the only field that allows multiple values is the author name. For them, an additional fixed rule to distinguish between name and last name is applied.

The definition used by ANSI/IEEE affirms that software architecture considers basically the intrinsic and extrinsic relationship among the fundamental components of a system [17]. The concept of services is the fundamental component of a SOA. SOA can conceive a relatively cheap solution with a better cost-benefit than when referring to systems that need to talk amongst themselves and processes that demand a larger flexibility and agility to assist the market revolutions. SOA introduces a new logic (services) layer within the computations distributed platform [14].

The architecture definition of SOA is a development guided to services. This means that the applications will be allocated in an interdependent way the response of an infrastructure of preset and pondered technology to create services with enough flexibility for being reused among the systems. Nowadays, the SOA is recognized as an important alternative for development, especially for business systems applications, allows flexibility, as the services can be supplied both locally or outsourced.

Two important points in the SOA: the consumer and the provider. The former consumes and requests the results to the provider, whereas the latter executes the service and answers the needs, as depicted in Figure 4.1 [15].

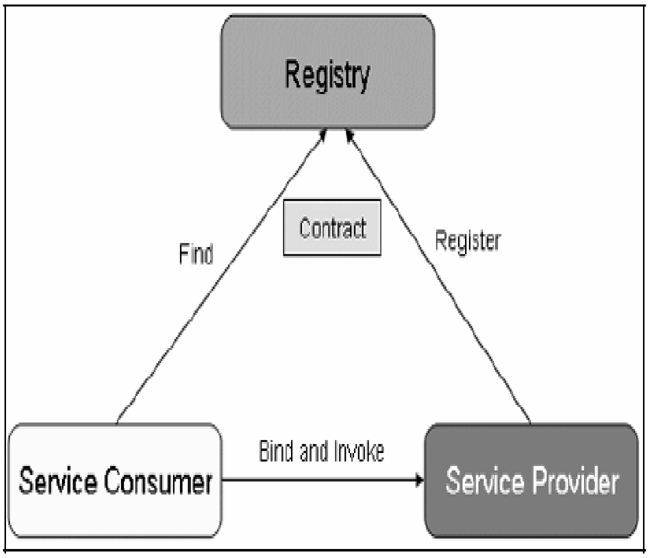


Figure 4.1. Basic paradigm for SOA [11]

Every service should possess a public interface, exposed at a place where the providers can access it. To request a service, it should only be necessary to obtain the interface. Moreover, the interface should only possess the relevant functions, in a high abstraction level, taking into account the gross granularity principle. Hence, while any functionality can be transformed into a service, the challenge is to define a service interface that is at the right level of abstraction [15].

a. Web Services:

Principle of SOA is its interoperability. It is fundamental that all the components communicate independently of the language in which they were built, of the operational system in which they are being executed and of the hardware architecture. Web Services conform well with the requirements since they use protocols and standard formats and accepted documents, today it is common to mix up Web Services with SOA. However, Web Services is a sort of incarnation of SOA, but not its definition. By its nature, Web Service technology favors the creation of components weakly coupled with gross granulation, but SOA components can be created using any other technology.

A Web Service exposes its interface for the users using a XML framework, known as Web Services Description Language, or WSDL. Using WSDL one can discover which are the data types, messages format and services made available by a Web Service.

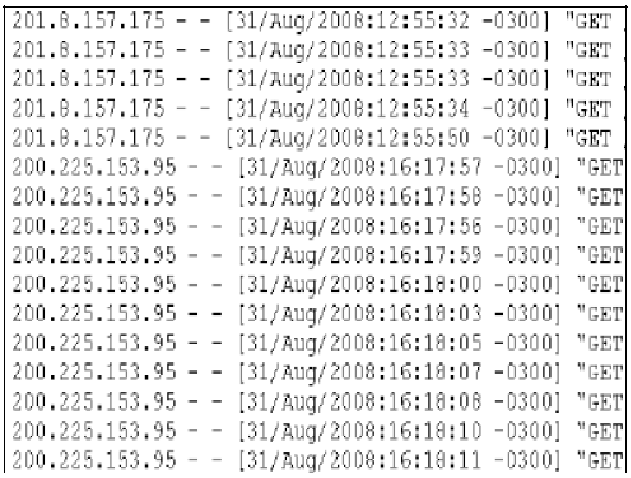
Request can be a Web Services using two ways: either they have direct access to WSDL or they use a registration service through an UDDI (Universal Discovery Description and Integration) interface. The WSDL pattern defines a Web Service as a collection of net endpoints, better known as ports. A port allows for some operations and each operation implies in the exchange of some messages that are formed by defined data types in an XML outline.

B. Proposed Architecture :

The implementation of a Web Service in the context of SOA provides flexibility for the distribution of services in the system, making it easier for commitment, in contrast with other more traditional technologies. Several works show the potential of Web Service and SOA combination for data mining systems on the web [16].

a. Building the SOA Web Services

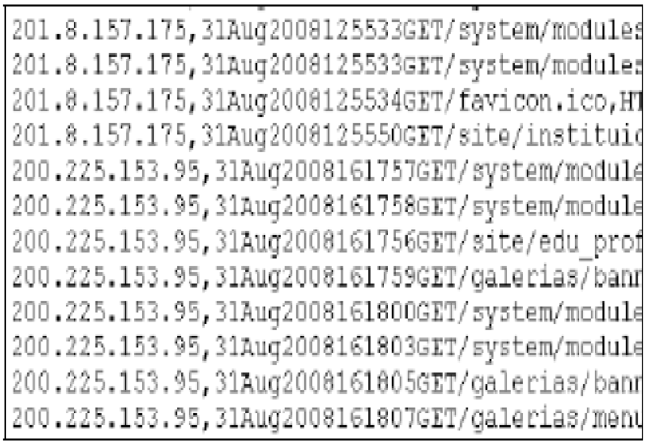
A SOA framework over traditional approach is Selection and retrieval, pre-processing, transformation, data mining and visualization modules were implemented as services and individually tested. Each service had its WSDL document created with the support of the development platform. After its creation, the tool also allowed the creation of each customer for the service in question, in an automated manner. Only after the creation of each customer, the overall framework was tested to verify the results and implementation consistency. After testing each individual service in place, services were grouped to verify the implementation of the whole process of Web Mining and observation of the automatic collaboration of each service with the next service. The first service to be implemented was the selection and retrieval module. This service, which is responsible for receiving an input file from the client, has methods to process the input file and specific methods to evaluate its structure and content. If data are in the correct format, the received file is sent in an automated way to the next service that executes preprocessing. With the help of a customer that was automatically generated by the development environment using the WSDL document, the log file is first sent to this service and the system should be checked to its structure. A fragment of this file can be seen in Figure.



```
201.8.157.175 - - [31/Aug/2008:12:55:32 -0300] "GET
201.8.157.175 - - [31/Aug/2008:12:55:33 -0300] "GET
201.8.157.175 - - [31/Aug/2008:12:55:33 -0300] "GET
201.8.157.175 - - [31/Aug/2008:12:55:34 -0300] "GET
201.8.157.175 - - [31/Aug/2008:12:55:50 -0300] "GET
200.225.153.95 - - [31/Aug/2008:16:17:57 -0300] "GET
200.225.153.95 - - [31/Aug/2008:16:17:58 -0300] "GET
200.225.153.95 - - [31/Aug/2008:16:17:56 -0300] "GET
200.225.153.95 - - [31/Aug/2008:16:17:59 -0300] "GET
200.225.153.95 - - [31/Aug/2008:16:18:00 -0300] "GET
200.225.153.95 - - [31/Aug/2008:16:18:03 -0300] "GET
200.225.153.95 - - [31/Aug/2008:16:18:05 -0300] "GET
200.225.153.95 - - [31/Aug/2008:16:18:07 -0300] "GET
200.225.153.95 - - [31/Aug/2008:16:18:08 -0300] "GET
200.225.153.95 - - [31/Aug/2008:16:18:10 -0300] "GET
200.225.153.95 - - [31/Aug/2008:16:18:11 -0300] "GET
```

Figure 4.2 Fragment of the log file

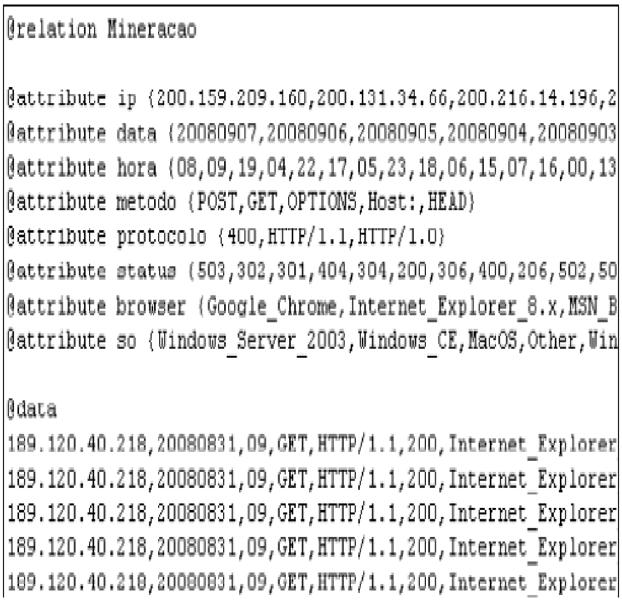
Pre-processing methods are responsible for eliminating incomplete records of the file and duplicate records, besides removal of fields and records that will not be used at the data mining stage. It also has a method that after the changes made in the file, sends the resulting file to the next service in the process of Web Mining. The file size is an important point to be considered, since the whole process will be conducted on the web. Thus, it is necessary to eliminate as much data as possible that are not interesting and necessary for the service of mining, ie, this service also deleted data that are not useful for the mining process. One example was the exclusion of characters "[]" and ":", as can be seen in Figure 5. What is wanted is the removal of all data that are needed or will not interfere with the results after service mining, aiming to reducing network traffic whenever is possible. The archive for this study, in its original state, had 24.2 MB. After the preprocessing and processing steps, the file sent to "Mining" had 6.05 MB, ie, the file to be mined now is almost 25% of its original size, which facilitated the traffic of this information over the internet. Thus, only data of interest will be sent to the processing service, which aims to place these data input to the mining algorithms.



```
201.8.157.175,31Aug2008125533GET/system/modules
201.8.157.175,31Aug2008125533GET/system/modules
201.8.157.175,31Aug2008125534GET/favicon.ico,HT
201.8.157.175,31Aug2008125550GET/site/instituic
200.225.153.95,31Aug2008161757GET/system/module
200.225.153.95,31Aug2008161758GET/system/module
200.225.153.95,31Aug2008161756GET/site/edu_prof
200.225.153.95,31Aug2008161759GET/galerias/bann
200.225.153.95,31Aug2008161800GET/system/module
200.225.153.95,31Aug2008161803GET/system/module
200.225.153.95,31Aug2008161805GET/galerias/bann
200.225.153.95,31Aug2008161807GET/galerias/menu
```

Figure 4.3. File after the step of preprocessing

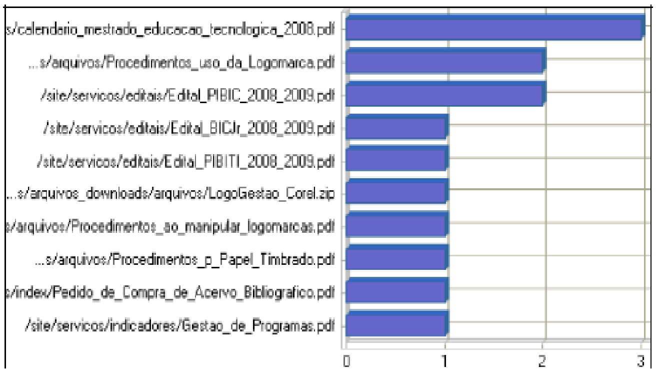
The service of Transformation, which is responsible for incorporating the processing stage of the process of Web mining, consists of methods that transform the file into a format acceptable by the mining algorithm. In the specific case of this implementation, data mining module service uses the Apriori algorithm, imported from the Weka tool. The file is transformed into a specific format for the Apriori algorithm and, thus, association rules can be extracted and data analysis have been implemented in the service of mining. The result of the file after the transformation service can be seen in Figure 4.4.

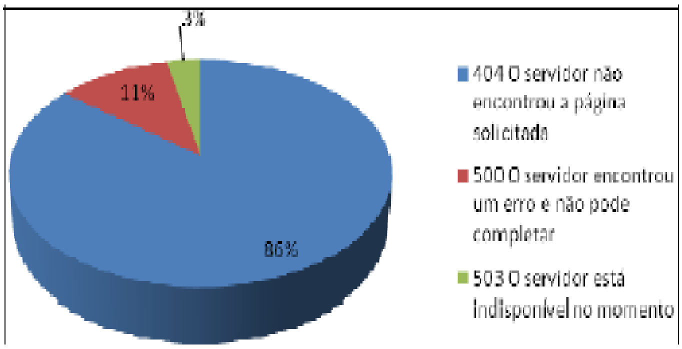


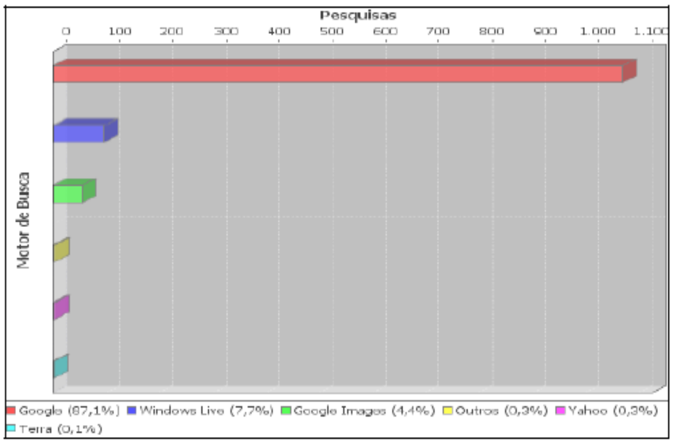
```
@relation Mineracao

@attribute ip {200.159.209.160,200.131.34.66,200.216.14.196,2
@attribute data {20080907,20080906,20080905,20080904,20080903
@attribute hora {08,09,19,04,22,17,05,23,18,06,15,07,16,00,13
@attribute metodo {POST,GET,OPTIONS,Host:,HEAD}
@attribute protocolo {400,HTTP/1.1,HTTP/1.0}
@attribute status {503,302,301,404,304,200,306,400,206,502,50
@attribute browser {Google_Chrome,Internet_Explorer_8.x,MSN_B
@attribute so {Windows_Server_2003,Windows_CE,MacOS,Other,Win

@data
189.120.40.218,20080831,09,GET,HTTP/1.1,200,Internet_Explorer
189.120.40.218,20080831,09,GET,HTTP/1.1,200,Internet_Explorer
189.120.40.218,20080831,09,GET,HTTP/1.1,200,Internet_Explorer
189.120.40.218,20080831,09,GET,HTTP/1.1,200,Internet_Explorer
189.120.40.218,20080831,09,GET,HTTP/1.1,200,Internet_Explorer
```

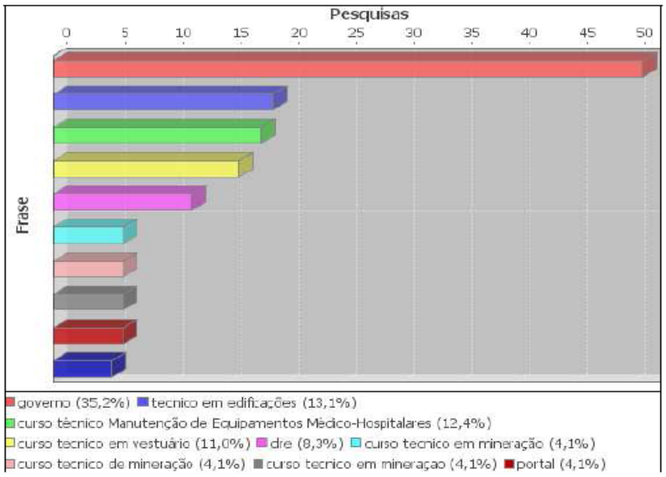
Figure 4.4. Archive generated by the Service Transformation

Finally, a method sends the file already converted directly to the next service. The data mining service has specific methods for data analysis and statistical methods to work with association. To perform statistical analysis there are methods that extract information from the file, such as number of accesses, consumed server bandwidth and access times. Parameters such as confidence and support are informed, and the package extracts association rules, returning a set of drawn rules. After these operations, a method is responsible for receiving statistical analysis and association rules for sending results to the last service in the entire process of Web Mining. Service Mining is considered the most important service the whole process. It is exactly this service that gets information from data that, previously, were not so useful. However, all earlier services that comprise the process are considered essential to the success of mining. After performing this service, the information is sent to the Web Service View, which incorporates the latest stage of the architecture, in order to this information be placed in an easyto- understand and structured graphics and tables. The last service was implemented in the service view. This service is designed to receive data from the data mining service and build an HTML file and to send this file to the client that originally sent the web mining requisition. This file is stored on the server where the service is running. To carry out the construction of the graphics has been implemented in the JFreeChart service library. This library can be used to generate pie charts, bar charts, line charts (with or without 3-D effect), Graphics, among many others. JFreeChart is written entirely in Java and can be used in any implementation of Java 2 (JDK 1.2.2 or higher). Examples of graphs generated can be seen in figures 4.5, 4.6, 4.7 and 4.8.









b. *Performance of the Implemented Services:*

The Apriori algorithm was chosen to extract association rules. The algorithm was imported from the Weka tool. Weka (Waikato Environment for Knowledge Analysis) is a system created to support the learning of algorithms. This whole system is implemented in Java and brings various algorithms used to extract knowledge in DB. It was developed at the University of Waikato, New Zealand and includes several methods for classification, association rules, clustering and prediction. It has the ability to integrate with other tools such as Web Services.

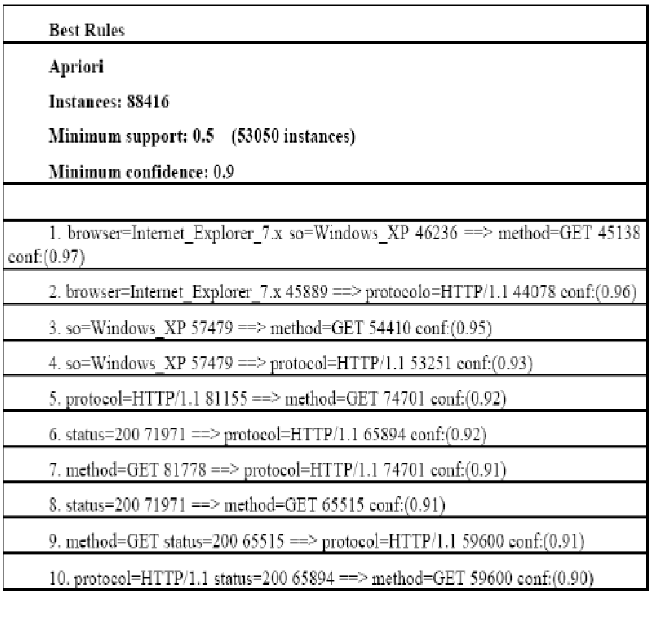
All the association rules drawn by the Apriori algorithm were analysed and considered as valid relation. They represent occurrence of values in the input file. The rules are set by combining all the possibilities of association between attributes. Finding the best rules means to select the rules with the greatest possible confidence. For this, the rules may be useful to find patterns of behaviour of users to access a Web site. The implementation will begin from a client that was built based on the WSDL document of the first Web service architecture that embodies the stage of selection and data recovery. This customer has the option to select a log file stored at some place and hold your shipment. The server generates a file with the records to access the pages, and by default saved in a single file to access the data in the period of one week. Each file with a log is automatically saved to a directory on the server. The file chosen for this covers the period from 2/05/2011 to 09:14 am on the date to 1/06/2011 at 06:23 am, corresponding to a week. The file in question has 24.2 MB in size and consist of accesses to CEFET-MG Web Site. After sending the file to the Web Service Selection and Recovery of Data, the mining process begins.

As a final result, association rules are extracted by the implemented algorithms. After execution, the information is sent to the visualization service, which incorporates the final stage of the architecture, putting this information in formats easy to understand, structured as charts and tables.

c. Produced Rules:

Implementation result of the Apriori algorithm, some association rules were obtained, as shown in Table 1. Some rules cannot provide information that may significantly impact on business or in the structure of the site. Most of the rules listed above are called "simple", that means, rules known by the business analysts. However, the association rules provide information that is important and useful for the Usage Mining user.

Table 4.1.Best Rules from the Algorithm Apriori



Best Rules
Apriori **Instances: 88416** **Minimum support: 0.5 (53050 instances)** **Minimum confidence: 0.9**
1. browser=Internet_Explorer_7.x so=Windows_XP 46236 ==> method=GET 45138 conf:(0.97)
2. browser=Internet_Explorer_7.x 45889 ==> protocolo=HTTP/1.1 44078 conf:(0.96)
3. so=Windows_XP 57479 ==> method=GET 54410 conf:(0.95)
4. so=Windows_XP 57479 ==> protocol=HTTP/1.1 53251 conf:(0.93)
5. protocol=HTTP/1.1 81155 ==> method=GET 74701 conf:(0.92)
6. status=200 71971 ==> protocol=HTTP/1.1 65894 conf:(0.92)
7. method=GET 81778 ==> protocol=HTTP/1.1 74701 conf:(0.91)
8. status=200 71971 ==> method=GET 65515 conf:(0.91)
9. method=GET status=200 65515 ==> protocol=HTTP/1.1 59600 conf:(0.91)
10. protocol=HTTP/1.1 status=200 65894 ==> method=GET 59600 conf:(0.90)

V. CONCLUSION

In general, the desired information is embedded in the deep Web pages in the form of data records returned by Web databases when they respond to users' queries. Therefore, it is an important task to extract the structured data from the deep Web pages for later processing. In this paper, we focused on the structured Web data extraction and integration, including data record extraction and data item extraction. First, we surveyed on the Web data mining and research process in web databases meanwhile, we found that the visual information of Web pages can implement in xml format. In this work, it was demonstrated the great potential of using SOA and Web.

VI. REFERENCES

[1] J.Srivastava, R.Cooley, M.Deshpande, P.tan,web Usage Mining: Discovery and applications of usage Patterns from web data,SIGKDD Explorations,Vol.1,No.2,Jan.2000

[2] J.han and M.kamber, Data Mining Concepts and Techniques, Morgan Kaufmann Publisher, 2001.

[3] R. Agrawal, T. Imielinski, A. Swami. Mining Association Rules between Sets of Items in Large Databases. Proceedings of the 1993 ACM SIGMOD Conference,1993.

[4] HTML Tidy. http://www.w3.org/People/Raggett/tidy/.

[5] XHTML: The Extensible HyperText Markup Language,W3C Recommendation, January2000. ttp://www.w3.org/TR/xhtml1.

[6] XML Path Language (XPath), W3C Recommendation, November 1999. http://www.w3.org/TR/xpath.html.

[7] XSL Transformations (XSLT), W3C Recommendation, November 1999. http://www.w3.org/TR/xslt.html Web gen

[8] Naveen Ashish and Craig Knoblock. Wrapper Generation for Semi-structured Internet Sources. In Proc. ACM SIGMOD Workshop on Management of Semistructured Data, Tucson, Arizona, May 1997

[9] KACZMAREK T., Deep Web data integration for company environment analysis (in Polish). PhD thesis, Poznan University of Economics, 2006.Deep web

[10] KELLER A., GENESERETH M., DUSCHKA O., Infomaster: an information integration system. 1997 ACM SIGMOD International Conference on Management of Data, 1997, 539–542.

[11] MELNIK S., PETROPOULOS M., BERNSTEIN P., QUIX C., Industrial-strength schema matching. SIGMOD Record, 33(4), 2004, 38–43.

[12] ORDILLE J., RAJARAMAN A., HALEVY A., Data integration: The teenage years. 32nd International Conference on Very Large Data Bases, 2006.

[13] Erl T., "Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services", Pearson Education Inc., 2004.

[14] Mahmoud Q.H., "Service-Oriented Architecture (SOA) and Web Services: The Road to Enterprise Application Integration (EAI)", Sun Microsystems Inc, 2005. Retrieved 04 April 2008 from java.sun.com/developer/technicalArticles/WebServices/soa.

[15] Guedes D.O., W. Meira Jr., R.A.C. Ferreira, "Anteater: A Service- Oriented Architecture for High-Performance Data Mining", IEEE Internet Computing, 10, 36-43, 2006.

[16] ANSI/IEEE, "Recommended Practice for Architectural Description of Software-Intensive Systems", ANSI/IEEE Std 147, 2000.

[17] [PNM1995]M. Pazzani, L. Bguyen, S. Mantik"Learning from Hotlistsand Coldlists–Towards a WWW Information Filtering and Seeking Agent", in Proceedings of the IEEE International Conference on Tools with AI, 1995.

[18] [PMB1996]M. Pazzani, J. Muramatsu, D. Billsus, "Syskilland Webert: Identifying Interesting Web Sites", in Proceedings of AAAI/IAAI Symposium, 1996.

[19] [PBMW1998] L. Page, S. Brin, R. Motwaniand T. Winograd"The PageRankCitation Ranking: Bringing Order to the Web" Stanford Digital Library Technologes, 1999-0120, January 1998.

[20] [PE1999]M. Perkowitz, O. Etzioni, "Adaptive Web Sites: Conceptual Cluster Mining", in Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, Sweden, 1999.

[21] [PPT+2002]B.Prasetyo, et. al., "Naviz: User Behavior Visualization of Dynamic Page", Pacific-Asia Conference on Knowledge Discovery and Data Mining 2002, Taipei, Taiwan

[22] [PSS2001] A. Pandey, J. Srivastava, S. Shekhar, "A Web Intelligent Prefetcherfor Dynamic Pages Using Association Rules –A Summary of Results, SIAM Workshop on Web Mining, 2001.

Dr. Ch.G.V.N. Prasad, has done his M.Tech in Computer Science From JNTU, Hyderabad. Ph.D (CS) in Data Mining from Allahabad University. He has 20 years of teaching and Industry experience, Also has worked as Scientist in National Informatics Centre, Hyderabad . Currently working as Professor and Head of the Department CSE at Sri Indu College of Engineering &Technology, Hyderabad, He has guided many PG level and engineering students. Life member of CSI

P. Satyavathi pursuing M.Tech CSE at Sri Indu College of Engineering & Technology JNTUH. Her areas of interest include Data Mining, Web Application, Networks.

S. Swathi pursuing M.Tech CSE at Sri Indu College of Engineering & Technology JNTUH. Her areas of interest include Data Mining, Web Application, Wireless Networks.