AN APPROACH FOR EXTRACTING INFORMATION FROM NARRATIVE WEB INFORMATION SOURCES

A. Vashishta
Computer Science & Software Engineering Department
University of Wisconsin - Platteville
Platteville, WI 53818, USA
Email: vashisha@uwplatt.edu

A. Shah
Computer Science Department
King Saud University
P. O. Box 51178, Riyadh 11543
Saudi Arabia
Email: aashah@ccis.ksu.edu.sa

Abstract

There is valuable information on many different subjects available online on the Web. Many web information sources (WISs) present their information in semi-structured format, which made it uneasy to directly extract and manipulate the desired information. However, there have been many attempts are made to develop some approaches that can automatically generate the procedures required for extracting desired information from particular WISs. Those procedures are referred to as *wrappers*, which are basically a code that is tailored for extracting some relevant information from a particular WIS. However, current approaches extract and tackle the information that are presented in a tabular form on the WISs, even though there exist a big number of WISs that present their information in a narrative form. In this paper, we purpose an approach for wrappers that can automatically extract the desired information from narrative WISs and model them into XML, and this extracted information as XML document, can easily be queried using XQuery.

Introduction

There is a wealth of interesting and valuable information sources on the Web. Currently, there are a variety of information sources available online such as telephone directories, product catalogs, weather forecasts, stock exchange, and many more. It is not much possible to do with these information except look at them or build a specific application by processing the available data. This problem stems from the fact that the contents of these Web information sources (WISs) presented on the Web, are in the semi-structured

format, and the data in the semi-structured format cannot be queried and manipulated in a general and flexible way. A data that has irregular or unknown structure, and this structure changes rapidly (such as weather forecasting and stock exchange data), is referred to as the *semi-structured data* (Dan Suciu (1998).

Many researches proposed different automated approaches to extracted desired information from particular web information sources (WISs). Some of these approaches focus on the extracting the desired information from WISs, on which the information is presented in a tabular form (e.g., see Figure 1), even though there are a large number of WISs that present their information in a narrative (or text) form (see Figure 2). For instance, if we would like to answer some queries such as "What is the location of Michigan?" or "What are the countries that have their total area greater than 500,000 square miles?". We cannot answer to this type of queries using the presently available the information extracting approaches. To get answer of this type of queries from the narrative form WISs, we need an approach (or technique) that can extract desired information from such type of WISs. In this paper, we propose an automated approach for the extracting information from narrative type of WISs, and this approach is referred to as the *Narrative Information Extractor (NIE)*. In order to allow the extracted information to be queried and manipulated in a flexible way, the extracted information is transformed to Extensible Markup Language (XML) format (or XML documents) (Dan Suciu, 1998). Then, XML query language, XQuery or other similar type of languages can be used to query the extracted document, as this query language, XQuery, allows to pose queries, translate, and integrate the extracted XML document.

Most of current information retrieval and integrating systems/approaches work in two steps. In the first step the desired information is extracted from different WISs, and the second step integrates the extracted information in a desired format for the user. Later, the user may need to use some tool in order to query and manipulate the information. Our approach, i.e., NIE, works is simple and it works in one step. It extracts the desired information from the different WISs and transforms the extracted information in an XML documents into a XML document. Since XQuery is already available for querying the data in the XML format, the user can query and manipulate the extracted information in the way he wants by using XQuery.

The remainder of the paper is organized as follows. In Section 2, we give an insight to the problem's background and related work. In Section 3, we proposed the approach Narrative Information Extractor (NIE). We give concluding remarks and future directions of the work in Section 4.

City	Province	Population	Low	High
Riyadh	Central	4,000,000	1	45
Dammam	Eastern	2,000,000	3	40
Jeddah	Western	3,000,000	5	45

Figure 1: A web information source that presents information in a table

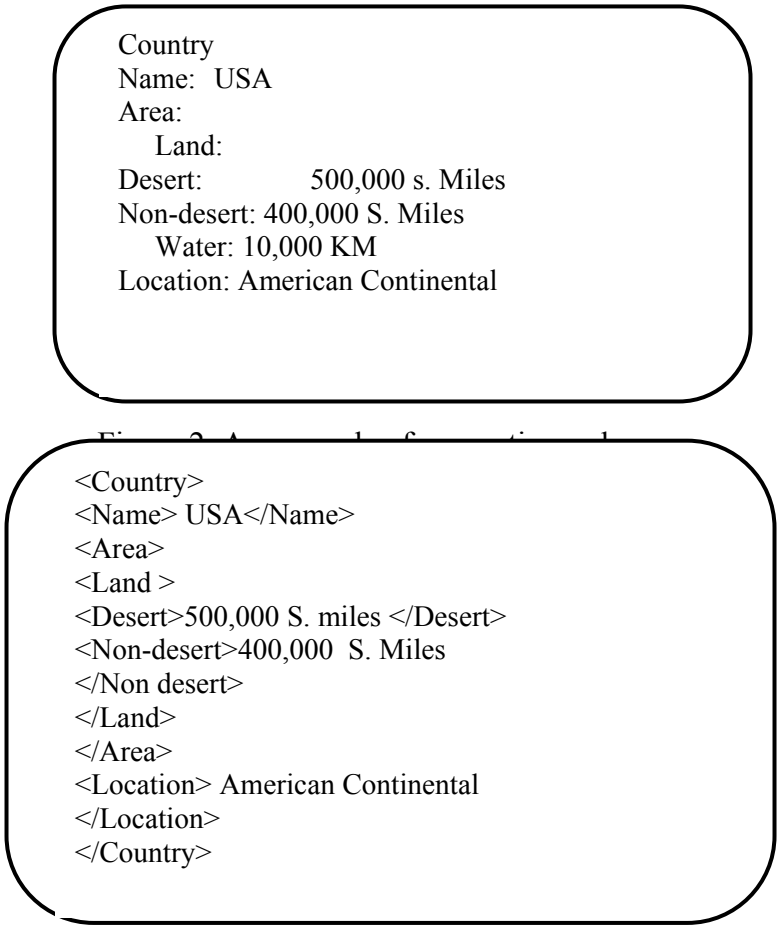

Figure 3: An XML document representing some information about USA

2. Background and Related Work

The introduction of the Web technology has tremendously increased the number and size of the WISs. The huge number and size of these WISs have necessitated the need for some approaches/techniques that can extract desired information directly and automatically from the WISs. These approaches are known as Information Extraction approaches. We define *Information Extraction* (IE), as the task of locating and extracting desired information from some specified WISs. Generally, the objective of the information extraction task is to extract and transfer the extracted unstructured information into a structured format (Line, 1999). Information that is extracted from heterogeneous WISs but with similar schemas can also be integrated and presented in a uniform way.

Nowadays the information extraction (IE) has become a topic of a general interest. In the following, we give the reasons of interest of IE and their associated issues.

(i) **Querying and manipulation of information**: Due to the semistructured format of the information on the Web, WISs cannot easily be queried and manipulated like the relational databases. For example, a WIS provides weather forecasts for the cities of the Michigan state, and if someone decides to have a standalone application that can automatically use the weather WIS for the answering his queries about the forecasts of some specific cities, then this would require some work to be done by an IE tool.

(ii) **Comparison of facts**: Many heterogeneous WISs contain similar information that pertains to the same topic, area and context. Due to huge number and size of these current WISs, it is almost impossible to examine all of these WISs at the same time using an integration mechanism that can gather relevant information from all available WISs and integrate them into a uniform structure. For example, one may want to compare the prices of one specific item offered by different merchants' sites on the Web.

(iii) **Automatic analysis**: When information is extracted and integrated in a uniform structure, then data mining techniques can be used for the discovery of the desired information from the uniform structure. An IE system can, for instance, extract and integrate the stock market rates provided by the relevant WISs. Data mining algorithms can then be applied, in order to help the broker/investor to decide whether it is good to buy particular stocks or not, based on the current information and circumstances.

(iv) **80% of the Web is hidden** (Line, 1999): Search engines use specific information retrieval techniques and information retrieval (IR) models, such as Boolean, Vector and Probailistic classical IR models and their extensions (Baeza-Yates, R. & Ribeiro-Neto,1999), for indexing the information available in WISs. Current statistics show that, these search engines are able to include and index only 20% of the available information in their WISs (Line, 1999). This means that in the future there will be a need of special tools that can extract information from the remaining 80% of remaining information from the WISs to full utilize of the information on the Web.

2.1 Extensible Markup Language (XML)

XML is a standard markup language that is proposed by World Wide Web Consortium (W3C). It is a nonprofit organization that studies and recommends the technical standards for the Web (Dan Suciu (1998).. The first draft of XML was approved by W3C in 1998. XML is used for defining, displaying and exchanging data and information on the Web, and it has become an international standard for these purposes.

The XML language was developed to overcome a major weakness of Hypertext Markup Language (HTML), which is its inability to be used for defining and describing the structure of a text document. XML is considered to be more powerful than HTML due to the following reasons:

(i) Users can define their own tags.
(ii) Document structures can be nested to any level.
(iii) Any XML document can contain an optional description of its grammar for use by applications that need to perform structural validation of the document. This grammar portion of an XML documents is called *Document Type Descriptor* (DTD), and it is considered as schema of the XML document.

As it is shown in Figure 3, the information in XML is grouped into elements delimited by tags, and elements can be nested. The example in the figure contains *country* enclosed between start tag <*country*> and end tag <*/country*>. The *country* tag has three nested tags: *name, area*, and *location*. The start and end tags with the text in-between is called an element, e.g. <*name*> USA<*/name*>.

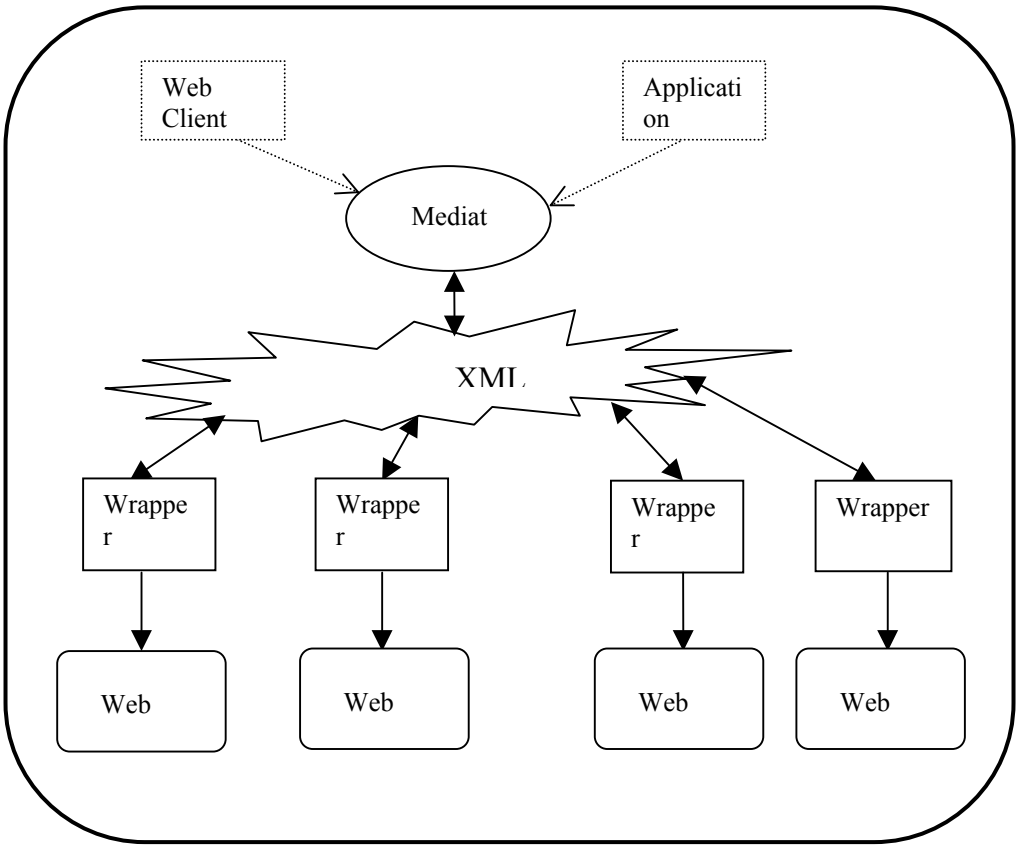

Figure 4: Architecture of the web information integration

2.2 Web Information Integration

The process that is performed on the information after extracting it from heterogeneous WISs, is the integration of the information and it is referred to as the *Web Information*

Integration (or simply *Information Integration*). This process mainly integrates the extracted information into a uniform and regular structure so that it can be used as a single source of information. The general architecture of the information integration systems is shown in Figure 4, where dashed rectangle denotes entities that can access the system. The direction of an arrow denotes the direction of the invocation and exchange of information.

Users can issue queries to the integrated information using a standalone tools (called *mediators*) to answer the queries. The mediators are responsible for receiving a query from a user, making an execution plan for the query, and submitting it to the relevant wrappers (see Section 2.3). It is also responsible for receiving the result back from wrappers, refining and integrating the results, and returning it to the user. A wrapper for a WIS accepts queries about the information in the pages of that WIS, fetches relevant pages from the WIS, extracts the requested information and returns the result to the user. XML is used for the purpose of information exchange between mediator and different wrappers.

Several systems are proposed for the extracting information from different heterogeneous WISs and then integrating the extracted information and querying the integrated information. Infomaster is a complete system for information extraction and integration from different heterogeneous WISs (Line, 1999). A complete architecture of the system is also given (for details see (Line, 1999)). TheaterLoc is another system that allows users to retrieve information about theaters and restaurants in the Los Angeles area (Greg Barish et al, 1999). Its main focus is on the mediator technologies that are, data modeling, and query planning. Some attempts have also been made to use XML for the extraction and integration process. In (Thomas Lee at el 1999), a prototype is described of an information integration system. This prototype uses XML for managing its distributed and heterogeneous type of information. In (Ling Liu et al1999), a system is proposed in which the wrappers are encoded in XML to extract the information from different WISs. This system, then maps the extracted information into the XML format. XQuery (a query language for querying XML collection of documents) can be used to query the extracted information. In (Chaitanya Baru et al 1999), the information integration is done by extracting information from the web pages in XML format. It uses Document Type Description/Definition (DTD) schema of XML document. DTD allows the user to define complex queries that may involve joins.

2.3 Wrappers

A *wrapper* is as a procedure that is designed for extracting the desired contents from a particular information source and delivering those contents in a self-describing representation (Line, 1999). In the Web environment, wrapper converts information that is implicitly stored as an HTML documents, into information with regular structure, in order to be queried and manipulated using the existing relational DBMS techniques. We define *Wrapper Generation* as a tool that develops a wrapper according to given

specifications. Three approaches are used for the development of wrapper generation. These three approaches are given as follows:

(i) **Manual**: This approach involves writing ad-hoc code for extracting relevant information from the related WISs. Infomaster and Tissimis that are described earlier, use the manual approach (Line, 1999;
Hector Garcia-Molina et al,1995; J. Hammer et al 1997; , Joachim Hammer et al1997).

(ii) **Semi-Automatic**: This approach uses some support tools to help the user in designing the wrapper. These tools are visuals tools and allow the wrapper developer to highlight the desired information on a web page, in order to automatically generate the wrapper code required for retrieving that information from the page. Shopbot is a semi-automatic wrapper generator (R. Doorenbos et al 1997). In (Jean-Robert Gruser et al , 1998), another semi-automatic wrapper generator is reported that provides a programmable API that can access a WIS to extract selected information from the WIS.

(iii) **Automatic**: This approach uses machine-learning techniques to generate the wrappers automatically. This involves training an automatic wrapper generator by providing it with a set of web pages along with their target information to be extracted. In the area of automatic wrapper generation, not much work has been done. In (Nicholas Kushmerick. 1998; N. Kushmerick et al 1997), some investigators tried to build an automatic wrapper generator by training it. The input to the automatic wrapper generators is usually a set of pages along with labels, and the output is a wrapper program that extracts the values of those labels. The set of wrappers can then be used to extract information from new WISs if they belong to one of the derived classes. Also, in (Ion Muslea et al 1999), the machine-learning techniques are used to train finite automata for extracting the desired information. The pros and cons of each approach are given in Table 1.

Table 1: A comparison of three types of wrappers

Approach	**Advantages**	**Disadvantages**
Manual	1. Simple and straightforward approach.	i) The creator has to spend some time in coding understanding the structure of the document. ii) Coding can be tedious and error prone. iii) Sites are rapidly changing, so the maintenance of the associated wrappers is a difficult task.
Semi-automatic	i) The knowledge about the wrapper coding is not	It must be demonstrated for each new site and for each changed site how the data should be extracted as these systems cannot induce themselves the

	required. ii) It is less error prone and less time-consuming than manual approach.	structure of a site.
Automatic	i) This type of wrappers are induced automatically. ii) Error possibility is minimal. iii) A much faster approach.	i) Training is a very time-consuming and difficult task. ii) Training requires some intervention of human experts. iii) Being automatic, it does not guarantee that the correct wrappers is generated for all WISs.

3. The Proposed Approach: Narrative Information Extractor (NIE)

As we have mentioned earlier, the main objective of our proposed NIE approach is to extract desired information/data from a WIS that keeps its information in HTML and narrative form, and then transform the extracted information into a XML document. This XML document is used to be queried by XQuery language.

Our proposed approach (NIE) works in three (3) steps, namely, i) identification of headings, ii) refining and associating text, iii) mapping to XML form. These three steps work sequentially after fetching a desired web page as a HTML code from a WIS. In the following three sections, we give detailed workings of these steps. To illustrate the working of the NIE approach, we take a running example given in Figure 2 (as a HTML code) that is finally transformed into the XML document shown in Figure 3.

3.1 Step 1: Identification of Headings

In this step the *structure* and the *nesting hierarchy* of a narrative WIS are identified. This process of identification further involves identifying the *headings* that constitute the structure of the narrative WIS. The words or tokens indicate the beginning of a section, or a subsection, and we refer them as the *headings*. For example, in Figure 2, the headings are Country, Name, Area, Land, Desert, Non-desert, and Location. The identification of the headings is performed employing heuristics that analyze HTML code and formatting information. We suggest the following heuristics for the identification of the headings.

- A word is presented in bold.
- A word is presented in upper case.
- A word is followed by a colon.
- A word's font size is slightly greater than its succeeding text.
- A word is underlined.

Table 3: The heading list of the running example: Output of Step 1 of NIE

Heading	Font Size	Indentation
Country	20	0
Name	18	0
Area	18	0
Land	16	4
Desert	14	8
Non-desert	14	8
Location	18	0

Each identified heading is stored in a list called "headings list" along with its font size and indentation information. Font size and indentation of each heading are later used in the Step 3 to identify the nesting hierarchy of the narrative WIS in order to map the extracted information structure to XML format. In our running example, the input to this step is the HTML page shown in Figure 2 and its output is the 'Heading List" shown in Table 3. In the example, the font size and indentation numbers are relatively assumed

3.2 Step 2: Refining and Associating Text

This step performs the following functions:

i) Extraction pf the text that lies between two consecutive headings. For example, after the heading "Name" in Figure 2, the associated text to be extracted is "USA".

ii) Refines of each extracted text segment, by eliminating the HTML tags that may occur in the extracted text such as <*I*>. This tag denotes the italic text.

iii) Associates the refined text to its corresponding heading using the heading list.

In the running example, the input to this step is the heading list given in Table 3 and the output is the information given in Table 4.

Table 4: The output of Step 2 of NIE

Heading	Associated Text
Country	
Name	USA
Area	
Land	
Desert	500,000 S. Miles
Non-desert	400,000 S. Miles
Location	American Continental

3.3 Step 3: Mapping to XML Form

This step uses the information font size and indentation values associated with each heading which are built by Step 1 and Step 2 (see Table 3 and Table 4) to construct the structure of a XML document. To accomplish this mapping, we propose an algorithm. In

the running example, the inputs to this step are Tables 3-4, and the desired output of this step is the XML document given in Figure 3.

The proposed algorithm is given in Figure 6. It is composed of two basic functions: i) *construct_XML*, ii) *write_heading* (see Figure 6). The function *construct_XML* takes the headings' list as input, initializes the variable *mapped* associated with each heading to "False" to indicates whether the associated heading has already been mapped to the XML document or not, and then calls the recursive function *write_heading* on two parameters. The first parameter is heading to be printed, and the second parameter is the heading's list starting from the second entry. The function *write_heading* is a recursive function that goes through the headings' list and creates corresponding XML nesting hierarchy using the headings' font size and indentation information. This function takes the two parameters: the heading to be printed, and the headings' list to be processed. Table 5 gives a detailed trace of the algorithm for our running example. Note that iteration denotes the number of times the While-loop has been executed with respect to the current "heading". Also "Stops" denotes whether the function execution has reached to its end or not, with respect to the current "heading". The output of the proposed algorithm is the XML document shown in Figure 3.

```
construct_XML (list)
{
   for all headings in list do
         mapped = False;

   first_heading = list;
   write_heading(first_heading, list->next);
}
write_heading(heading, list)
{
1    stop = false;
2    heading->mapped = true;
3    append_start_tag(heading->name);

4    if ( heading has associated text)
5       append_text(heading->text);

6    while (not end of list) and (not stop)
     {
7       curr_head = list;
8       list = list->next;

9       if ( not curr_head->mapped )
        {

10         if ( heading->font_size > curr_head-
>font_size ) or
              ( heading->indentation <
curr_head-> indentation )
11              write_heading(curr_head, list);

12         else
13             stop = true;

        } // End if

    } //End while

14  append_end_tag(heading->name);
}
```

Figure 6: The proposed algorithm – Step 3

Table 5: Trace of the algorithm for the running example

No	Heading	Iteration	curr_head	Appended Text	Stops
1	country	1	name	<country>	No
2	name	1	area	<name>USA</name>	Yes
3	country	2	area		No
4	area	1	land	<area>	No
5	land	1	desert	<land>	No
6	desert	1	non-desert	<desert>500,000 S. Miles </desert>	Yes
7	land	2	non-desert		No
9	area	2	desert		No
10	area	3	non-desert		No
11	area	5	location	</area>	Yes
12	country	3	land		No
13	country	4	desert		No
14	country	5	non-desert		No
15	country	7	location		No
16	location	1		<location> American Continental </location>	Yes
17	country	8		</country>	Yes

4. Conclusions and Future Work

Currently available information extraction approaches deal with those WISs that present their information in a tabular form. There is a huge number of WISs that present their information in a narrative form, and there is a need of an approached (or technique) that can extract information from these WISs. We have presented such an approach for the automatic extraction of information from the narrative WISs. This approach consists of

three main steps. This approach mainly transforms a desired HTML code into an XML documents. This XML documents, then, can be used querying and manipulation purposes by using XQuery and any similar type of language.
We are planning to add some enhancements to the NIE approach to include the following features:

i) Automation of the first two steps
ii) Multimedia data extraction
iii) Extracting information from WISs of tabular form

References

Baeza-Yates, R. & Ribeiro-Neto< B. (1999) .Modern Information Retrieval. Addison-Wesley Publishing Company.

Chaitanya Baru, Amarnath Gupta, Bertram Ludascher, Richard Marciano, Yannis Papakonstantinou, & Pavel Velikhov (1999) .XML-Based Information Mediation with MIX. *the Proceedings of ACM SIGMOD Conference on Management of Data*, (597-599).

Dan Suciu (1998) .Semistructured Data and XML. *the Proceedings. of the 5th Int. Conference of Foundations of Data Organization (FODO'98)*.

Greg Barish, Craig A. Knoblock, Yi-Shin Chen, Steven Minton, Andrew Philpot, & Cyrus Shahabi, 1999) .TheaterLoc: A Case Study In Building An Information Integration Application. *In IJCAI Workshop on Intelligent Information Integration*, Stockholm, Sweden.

Hector Garcia-Molina, Joachim Hammer, Kelly Ireland, Yannis Papakonstantinou, Jeffrey Ullman & Jennifer Widom (1995) .Integrating and Accessing Heterogeneous Information Sources in TSIMMIS. *the Proceedings of the AAAI Symposium on Information Gathering*, (61-64).

Ion Muslea, Steve Minton & Craig Knoblock (1999) .A Hierarchical Approach to Wrapper Induction. *the Proceedings of the Third Annual Conference on Autonomous Agents*, USA, (190-197).

Jean-Robert Gruser, Louiqa Raschid, Maria Esther Vidal & Laura Bright (1998) .Wrapper Generation for Web Accessible Data Sources. *the Proceedings of the 3rd IFCIS International Conference on Cooperative Information Systems*, New York City, New York, (14-23).

J. Hammer, H. Garcia-Molina, R. Cho, R. Aranha, & A. Crespo (1997) .Extracting semistructured information from the web. *the Proceedings of the Workshop on Management of Semistructured Data*, Tucson, Arizona, (16-25).

Joachim Hammer, Hector Garcia-Molina, Svetlozar Nestorov, Ramana Yerneni, Marcus Breunig, & Vasilis Vassalos (1997) .Template-Based Wrappers in the tsimmis System. *the Proceedings of the ACM SIGMOD Conference on Management of Data*, Arizona, (532-535).

Line Eikvil, (1999) .Information Extraction from World Wide Web - A Survey. A Technical Report (Report No. 945), Norwegian Computing Center, Oslo, Norway.

Ling Liu, Wei Han, David Buttler, Calton Pu & Wei Tang (1999) .An XML-based

Wrapper Generator for Web Information Extraction. *the Proceedings of ACM SIGMOD International Conference on Management of Data*, pp 540-543.

Nicholas Kushmerick. (1998) .Wrapper Induction: Efficiency And Expressiveness (Extended Abstract). *the AAAI-98 Workshop on AI and Information Integration*.

N. Kushmerick, D. Weld & R. Doorenbos (1997) .Wrapper Induction for Information Extraction, *the Proc. 15th Inernational. Joint Conference AI*, (729-735).

R. Doorenbos, O. Etzioni & D. Weld (1997) .A Scalable Comparison-Shopping Agent For The World Wide Web. *the Proceedings. Autonomous Agents*, (39-48).

Thomas Lee, Melanie Chams, Robert Nado, Stuart Madnick, & Michael Siegel (1999) .Information Integration With Attribution Support For Corporate Profiles. *the Eighth International Conference on Information and Knowledge Management (CIKM)*, Kansas City, Missouri,.