Functional Composition of Web Databases

Mori, Masao
Office of Information for University Evaluation, Kyushu University

Nakatoh, Tetsuya
Communication and Computing Center, Kyushu University

Hirokawa, Sachio
Communication and Computing Center, Kyushu University

https://hdl.handle.net/2324/3627

出版情報：Proceedings of ICADL2006 (Lecture note in Computer Science 4312). 4312, pp.439-448, 2006-11. Springer Berlin / Heidelberg
バージョン：
権利関係：

Functional Composition of Web Databases

Masao Mori[1], Tetsuya Nakatoh[2], and Sachio Hirokawa[2]

[1] Department of Informatics, Kyushu University.
6-10-1 Hakozaki, Fukuoka, 812–8581, Japan.
masa@i.kyushu-u.ac.jp,
[2] Computing and Communications Center, Kyushu University.
6-10-1 Hakozaki, Fukuoka, 812–8581, Japan.
{nakatoh, hirokawa}@cc.kyushu-u.ac.jp

Abstract. This paper proposes the architecture of the functional composition of Web databases (WebDBs). Unlike a general search engine which receives keywords and returns a list of URLs, a WebDB receives a complex query and returns a list of records. The complex query specifies the condition of each field of the records. The process of composing WebDBs is described as a script, where a user chooses the target WebDBs and describes how to connect the output from one WebDB to the input of another WebDB and how to generate outputs. The novelty of the proposal is that both the WebDBs and output formats are considered as components of the same level and that the reuse of new keywords is represented as a connection (CGI links). Once the process is described as a script, the user can use the script for a new WebDB of his own.

1 Introduction

An increasing number of search engines are available on the Web besides general search engines such as Google. There are also databases with a Web interface. We can obtain high-quality information for a particular purpose from these databases.

However, information on these Web databases (WebDBs) cannot be indexed by general search engines and cannot be referred to directly because they are referred to only by the page that is generated dynamically from the database according to the user's query. Because of that, these databases are called by such names as Invisible Web [13, 12], Deep Web [1] and Hidden Web [4, 5].

We developed a system **DAISEn** [15] which performs a metasearch for such databases on the Web. Conventional metasearch engines integrate a fixed number of particular general search engines. The goal of DAISEn is the dynamic integration of an arbitrary set of databases on the Web.

On the other hand, there is a new trend of databases on the Web for a user to send a complex query. The queries are not just simple keywords; instead, they are the keywords which specify each field of the records that the user wants to retrieve from a database. For example, Amazon.com returns a list of book information which consists of the author, the title, the publisher, the price, ISBN, and so on. kakaku.com returns a list of prices of PCs and other electric products. Travelocity.com returns a list of hotel information for a specified location.

When we survey a specific subject with such WebDBs, we do not stop searching with a single trial. We usually keep searching until we have enough information. In many cases, we obtain new keywords during the search process and use them for the next step of the search. For example, we can get a list of local restaurants by a search and then collect information about the menu and price of each restaurant. People who want to buy a used car can collect and compare detailed information on the cars obtained by a search. Those searches are performed by the same WebDB repeatedly, or are performed by different WebDBs. Some keywords in the output can be used as input for the next step of the search. If an attribute of the output data of a WebDB is "NAME", it can be connected to the input of another WebDB which receives the name as a search keyword.

When an author name and a keyword are sent to a WebDB of scientific journal articles, the result is not a list of Web pages but a list of articles with the author, the title, the magazine name and the publication year. When investigating papers exhaustively, even if the first search result is obtained, the investigation is not finished. The search is further continued based on the obtained information. As an example, we consider the following search.

- Are there any other articles written by the same author?
- Are there any articles written by the coauthors?
- What kind of articles does the article cite?
- How is the paper cited?
- What are the important keywords in related research?
- Where are the authors' home pages?
- Is there any related project?

We have proposed architecture to realize a search engine that combines several WebDBs. A large listing of such WebDBs is available at Dnavi, a Database Navigation service provided by the National Diet Library, Japan[3]. We confirmed that there are 2,800 WebDBs in Dnavi and proposed a method to estimate the query form automatically [10]. And, we reported the current situation of Web databases with a complex query and the possibility to guess the input metadata from the output metadata [9]. Furthermore, we proposed the algorithm which extracts items of each record from an HTML source of an output result [11].

In this paper, we discuss the co-operation between WebDBs. The script language which we propose describes the data flow between WebDBs as components with input and output metadata. Furthermore, the special component which specifies the output format can be treated similarly. The system constructs a CGI from the script and performs a semantic metasearch using the target WebDBs. To demonstrate the feasibility of this approach, we show a personal WebDB that connects four WebDBs of major Japanese IT related journals.

There are many previous works, such as TSIMMIS [2], that have examined information integration on the Web. However, in those studies, there is a requisite

[3] http://dnavi.ndl.go.jp/

that a developer must offer a conversion program to a common data format or detailed information of the database. The technique of this paper obtains the required information only from the Web interface of each WebDB, and can realize results independently of the system of each site.

Kitamura *et al.* [6] proposed the script language MetaCommander for extracting and unifying information from the Web. The extraction of the information needed is attained by describing the procedure as a script. However, a user needs to describe the script which extracts or converts data from HTML documents. Therefore, a semantic description such as "extracting the element of authors from the list of the outputted book" cannot be performed in MetaCommander.

Information extraction from Deep Web, WISE-Integrator [3] and SE-LEGO [14] are known; in these, the metasearch to WebDB is built automatically. The architecture of the present paper is not a simple metasearch that integrates the outputs of heterogeneous WebDBs but a creation of a new WebDB from several WebDBs as its components.

The works by Knoblock *et al.* [7, 8] consider the construction of a personal information gathering tool with the integration of agents. Their goal is similar to ours. In order to define the connection between WebDBs, our system uses the script and, therefore, is more comprehensive. Moreover, we also propose the mechanism of the gathering information repeatedly by interaction with the user.

2 Composition of WebDBs

WebDBs can be considered as functions for which complex queries are input via Web interfaces and they output search results. Essentially, complex queries are a list of pairs of an attribute and instance. Similarly, search results essentially consist of a list of pairs of an attribute and instance. For example, when we make use of the WebDBs of electric journals, we provide some keywords into the query boxes. The WebDB shows the search results in browsers; these results can be a table of journals with title, author and coauthors, research keywords, etc. Each of the query boxes in the input form and each of the attributes of the table in the output data correspond to the input and output channels, respectively. In our architecture WebDBs are called *components* and their input and output interfaces are called *channels*. Composition is realized by passing data from output channels to input channels between WebDBs. We call the pairs of output and input channels *connections*.

Our architecture is characterized by the generation of CGI programs and executing them for each individual purpose. From the Web interface of our system, users give a statement of the WebDBs (called *components*) and define the connections of the components. Our architecture consists of three parts:

1. Components,
2. Composition of components, and
3. Personal output forms.

In this section we introduce the idea of components and their composition. In Section 3, the output components will be discussed.

2.1 Components

A WebDB is described as a *component* with input channels and output channels. Those channels are named with data *types*. A set of instances of these types is called a *record*. A component inputs and outputs a list of partial records: a tuple of instances in the form of a subset of types (Fig. 1). Note that "to output" is not meant as output for browsers. We consider the input and output user–interface as components.

Start component The *start component* is a component with no input channels. It only outputs queries from users.

WebDB component The components are equipped with wrappers and labeled with the names corresponding to the WebDBs. As WebDBs vary in their formats of input and output, wrappers of the components unify their formats.

Output component This is the interface to the users. Input data to the user component are shown to the users. Users choose something which is output data that become queries for the next search.

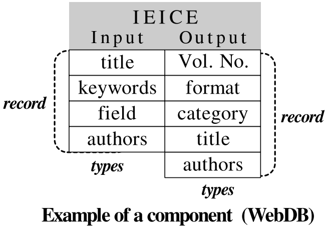


Fig. 1. Structure of a component

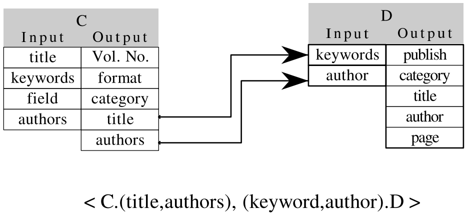


Fig. 2. Connection between components

2.2 Composition of components

Suppose that a component C with output channels $o_1, \cdots, o_p$ and a component D with input channels $i_1, \cdots, i_q$ are given. A pair

$$\langle C.(o_{p_1}, \cdots, o_{p_k}), \quad (i_{q_1}, \cdots, i_{q_k}).D \rangle$$

of component channels denotes a *connection* from C to D through the corresponding channels $o_{p_l} \rightarrow i_{q_l}$. For $l = 1, \cdots, k$, each output channel o_{p_l} pipes the data to input channel i_{q_l} (Fig 2). If the sets of channels are singular, the connection is denoted by an abbreviated mode such as $\langle C.author, keywords.D \rangle$. The components and their connections forms a directed graph called a *connection graph* with components as nodes and connections as directed edges.

Cycles are allowed in a connection graph under the condition that at least one *CGI link* edge is included. CGI link edges are introduced in Section 3.2.

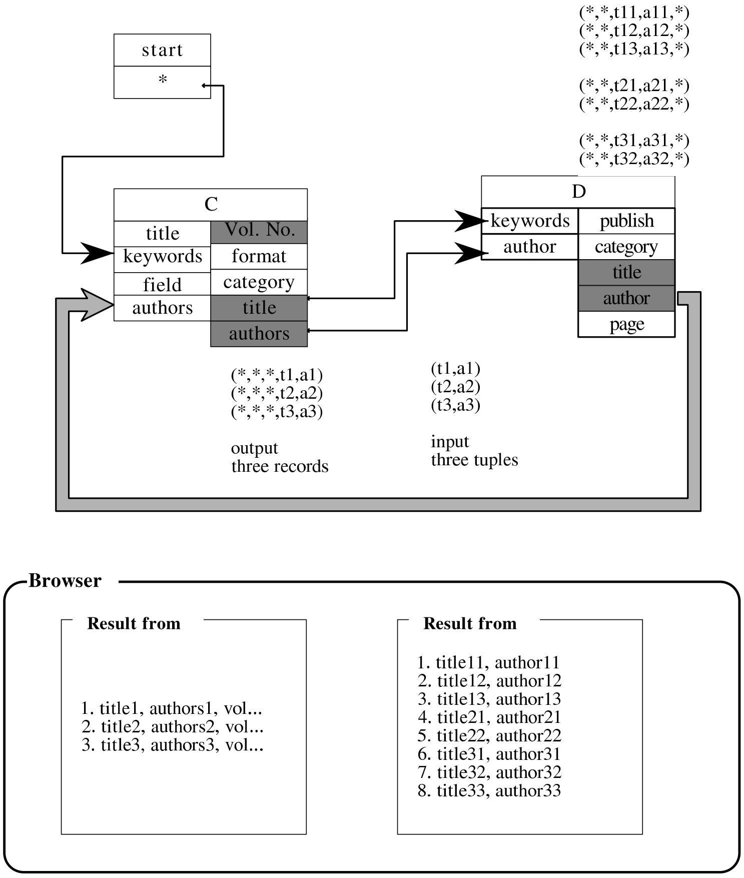


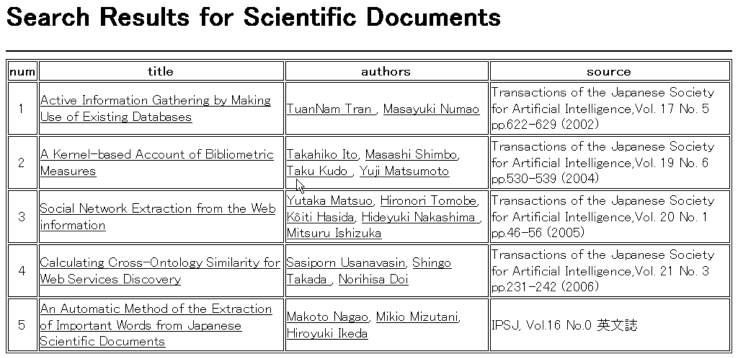
Fig. 3. Display channels

3 Output components

Output components are interfaces to users with functions for arranging the search results. We introduce three functions which users can combine to give output forms.

3.1 Displaying channels

Given a connection graph, the set of output channels, called *displaying channels*, is selected to display several outputs on the browser screen. In Fig 3, it contains the two output forms $C.\{volno, title, authors\}$ and $\{title, author\}.D$, which generate two listings separately on the browser screen.



Search Results for Scientific Documents

num	title	authors	source
1	Active Information Gathering by Making Use of Existing Databases	TuanNam Tran , Masayuki Numao	Transactions of the Japanese Society for Artificial Intelligence,Vol. 17 No. 5 pp.622-629 (2002)
2	A Kernel-based Account of Bibliometric Measures	Takahiko Ito, Masashi Shimbo, Taku Kudo , Yuji Matsumoto	Transactions of the Japanese Society for Artificial Intelligence,Vol. 19 No. 6 pp.530-539 (2004)
3	Social Network Extraction from the Web information	Yutaka Matsuo, Hironori Tomobe, Kôiti Hasida, Hideyuki Nakashima , Mitsuru Ishizuka	Transactions of the Japanese Society for Artificial Intelligence,Vol. 20 No. 1 pp.46-56 (2005)
4	Calculating Cross-Ontology Similarity for Web Services Discovery	Sasiporn Usanavasin, Shingo Takada , Norihisa Doi	Transactions of the Japanese Society for Artificial Intelligence,Vol. 21 No. 3 pp.231-242 (2006)
5	An Automatic Method of the Extraction of Important Words from Japanese Scientific Documents	Makoto Nagao, Mikio Mizutani, Hiroyuki Ikeda	IPSJ, Vol.16 No.0 英文誌

Fig. 4. Plain listing

3.2 Embedded CGI links

In Fig 3, there is an edge $\langle D.author, authors.C \rangle$ which yields a cycle. Such edges are called *CGI links* and indicate parameters passing toward the next search step. The third record "3. title13, author13" shown in the browser as the result from C, is represented as

3. title13, `<a href="http://..../**.cgi">` author13 `</a>`.

When the user clicks this link, the next step of the search hit component C is activated.

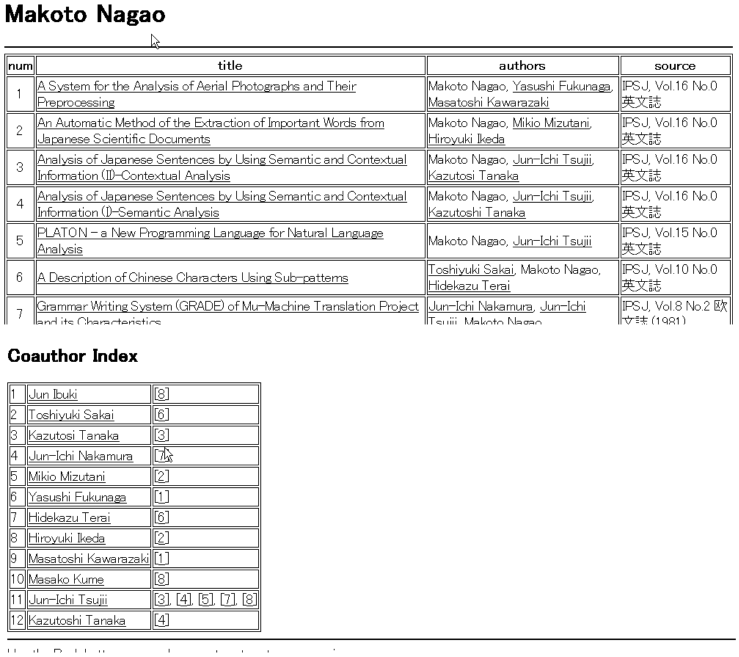
3.3 Basic filters for listing

The format of search results are unified by each wrapper of WebDB components. For example, in Fig 4 the search results for the query "Scientific Documents" are shown by simply listing the items from the WebDB components. Each author name has a CGI link which is a filter to obtain the coauthor list (Fig 5).

By clicking the author name "Makoto Nagao" in the fifth line, for instance, the coauthors table in Fig 5 appears as a histogram listing the articles for each coauthor. This filter is useful not only for the coauthor list, but also for previous results, e.g., the table of writers and the number of their articles associated with the keyword.

4 Example

In this section, we explain the prototype of the system which gathers information by composing a WebDB. It collects information about the papers from the WebDB of each of the following academic societies in Japan.



Makoto Nagao

num	title	authors	source
1	A System for the Analysis of Aerial Photographs and Their Preprocessing	Makoto Nagao, Yasushi Fukunaga, Masatoshi Kawarazaki	IPSJ, Vol.16 No.0 英文誌
2	An Automatic Method of the Extraction of Important Words from Japanese Scientific Documents	Makoto Nagao, Mikio Mizutani, Hiroyuki Ikeda	IPSJ, Vol.16 No.0 英文誌
3	Analysis of Japanese Sentences by Using Semantic and Contextual Information (II)-Contextual Analysis	Makoto Nagao, Jun-Ichi Tsujii, Kazutosi Tanaka	IPSJ, Vol.16 No.0 英文誌
4	Analysis of Japanese Sentences by Using Semantic and Contextual Information (I)-Semantic Analysis	Makoto Nagao, Jun-Ichi Tsujii, Kazutoshi Tanaka	IPSJ, Vol.16 No.0 英文誌
5	PLATON - a New Programming Language for Natural Language Analysis	Makoto Nagao, Jun-Ichi Tsujii	IPSJ, Vol.15 No.0 英文誌
6	A Description of Chinese Characters Using Sub-patterns	Toshiyuki Sakai, Makoto Nagao, Hidekazu Terai	IPSJ, Vol.10 No.0 英文誌
7	Grammar Writing System (GRADE) of Mu-Machine Translation Project and its Characteristics	Jun-Ichi Nakamura, Jun-Ichi Tsujii, Makoto Nagao	IPSJ, Vol.8 No.2 欧文誌 (1981)

Coauthor Index

1	Jun Ibuki	[8]
2	Toshiyuki Sakai	[6]
3	Kazutosi Tanaka	[3]
4	Jun-Ichi Nakamura	[7
5	Mikio Mizutani	[2]
6	Yasushi Fukunaga	[1]
7	Hidekazu Terai	[6]
8	Hiroyuki Ikeda	[2]
9	Masatoshi Kawarazaki	[1]
10	Masako Kume	[8]
11	Jun-Ichi Tsujii	[3], [4], [5], [7], [8]
12	Kazutoshi Tanaka	[4]

Fig. 5. Complex listing; coauthors list(the rest of article list from no.8 are omitted)

- IPSJ (Information Processing Society in Japan) [4]
- IEICE (The Institute of Electronics, Information and Communication Engineers)[5]
- JSAI (The Japanese Society for Artificial Intelligence) [6]
- JSSST (Japan Society for Software Science and Technology) [7]

This system mainly consists of the following three functions.

1. The function to search simultaneously to two or more WebDBs, and to show the user the integrated result. This is what is called a metasearch.
2. The function which extracts and lists authors in the result.
3. The function to offer the next search using the listed author as a keyword.

The mimetic diagram of the data connection of this system is shown in Fig. 6. We publish this system in `http://matu.cc.kyushu-u.ac.jp/whirler/`.

[4] `http://www.bookpark.ne.jp/ipsj/`

[5] `http://search.ieice.org/bin/search.php`

[6] `http://tjsai.jstage.jst.go.jp/`

[7] `http://www.jstage.jst.go.jp/browse/jssst/`

5 Conclusion

We proposed an architecture of the *functional composition of WebDBs* that aggregates WebDBs for an individual user's purpose. A user specifies the target WebDBs and how he uses them. The user's purpose is described as a script that consists of the list of WebDBs, the connection information which shows how the input and output are connected between WebDBs, and the list of display formats. Some fields of the output are shown as *CGI links* that are used as parameters for the next search step. As an output format, the user may write a sorted list of data, a histogram of some selected fields, and other filters as well as a plain listing of the records. Once the process is described as a script, the user can use the script as a new WebDB of his own.

In this paper, the authors considered the realistic situation that a user uses 5 or 6 WebDBs in his daily work and that he knows and can describe how he is using them. He only has to describe his search process as a script. Automatic selection of WebDBs will be necessary for a large pool of WebDBs. We assumed a wrapper for each WebDB. The automatic generation of such wrappers is indispensable and is an important problem for our architecture.

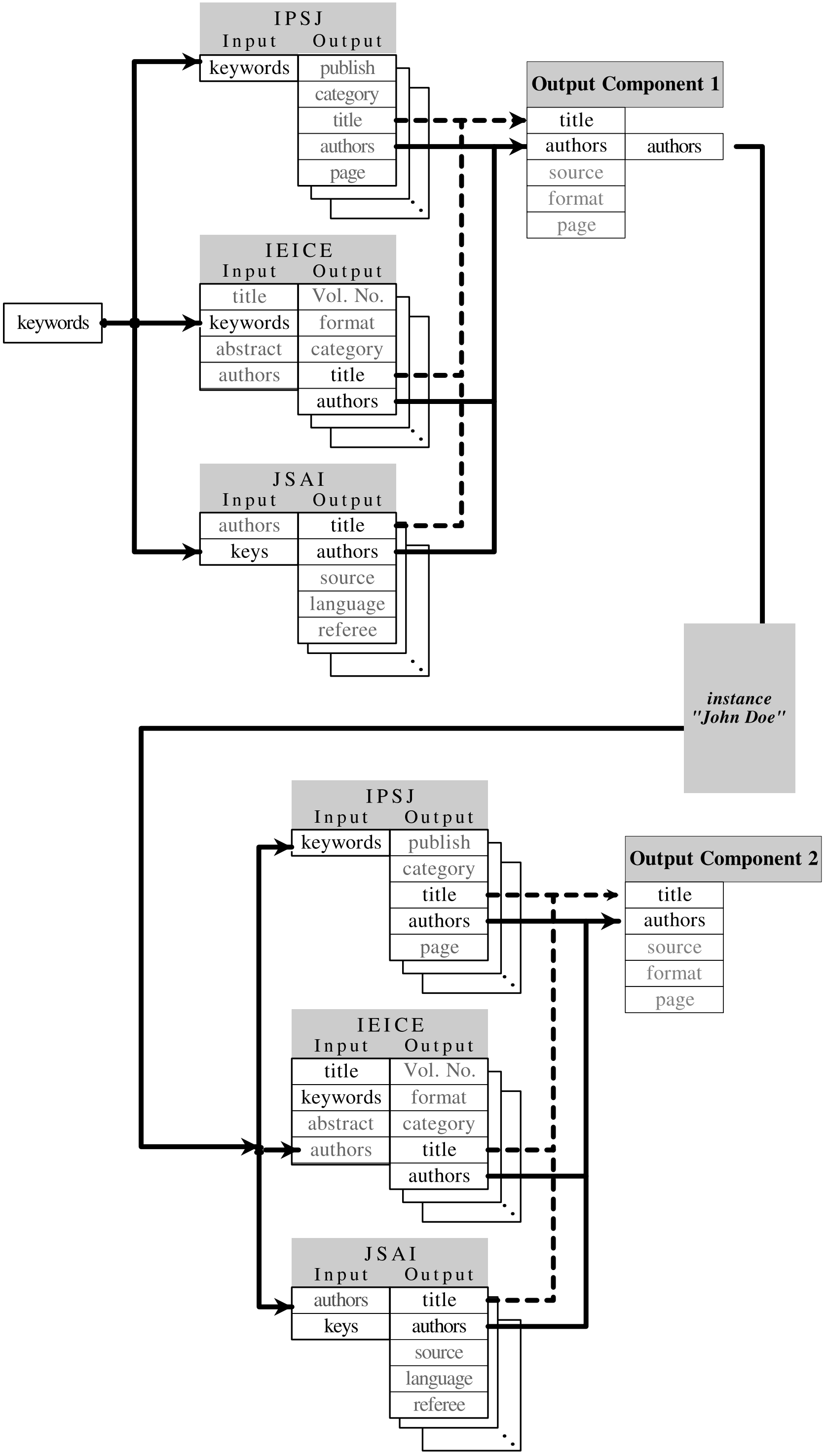


Fig. 6. Channels and connection graph

References

1. BrightPlanet, *The Deep Web: Surfacing Hidden Value*, BrightPlanet White Paper, 2000.
2. S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman and J. Widom, *The TSIMMIS Project: Integration of Heterogeneous Information Sources*, In Proceedings of IPSJ Conference, pp. 7-18, Tokyo, Japan, October 1994.
3. H. He, W. Meng, C. Yu, Z. Wu, *WISE-Integrator: A System for Extracting and Integrating Complex Web Search Interfaces of the Deep Web*, Proceedings of the 31st International Conference on Very Large Data Bases (VLDB2005), Trondheim, Norway, August 30 - September 2, 2005. pp.1314- 1317.
4. P. Ipeirotis, L. Gravano and M. Sahami, *PERSIVAL Demo: Categorizing Hidden-Web Resources*, JCDL2001, 2001.
5. P. Ipeirotis, L. Gravano and M. Sahami, *Probe, Count, and Classify: Categorizing Hidden-Web Databases*, ACM SIGMOD 2001, 2001.
6. Y. Kitamura, T. Noda and S. Tatsumi, *Single-agent and Multi-agent Approaches to WWW Information Integration*, Multiagent Platforms, Lecture Notes in Artificial Intelligence, Vol. 1599, Berlin et al.: Springer-Verlag, 133-147, 1999.
7. C. A. Knoblock, S. Minton, J. L. Ambite, N. Ashish, I. Muslea, A. G. Philpot, and S. Tejada, *The Ariadne Approach to Web-Based Information Integration*, International Journal of Cooperative Information Systems, vol.10, no.1-2, pp.145-169, 2001.
8. C. A. Knoblock, *Deploying Information Agents on the Web*, IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003. pp. 1580-1586.
9. T. Nakatoh, K. Ohmori and S. Hirokawa, *A Report on Metadata for Web Databases*, IPSJ SIG Technical Reports, 2004-ICS-138(17), pp. 95-98, 2004.
10. T. Nakatoh, K. Ohmori, Y. Yamada and S. Hirokawa, *COMPLEX QUERY AND METADATA*, Proc. ISEE2003, pp. 291-294, 2003.
11. T. Nakatoh, Y. Yamada and S. Hirokawa, *Automatic Generation of Deep Web Wrappers based on Discovery of Repetition*, Proc. of the First Asia Information Retrieval Symposium (AIRS 2004), pp.269-272, 2004.
12. P. Pedley, *The invisible web*, ASLIB, 2001.
13. C. Sherman and G. Pric, *The Invisible Web*, Information Today, Inc., Medfore, New Jersey, 2001.
14. Z. Wu, V. Raghavan, C. Du, K. Sai C, W. Meng, H. He and C. Yu, *SE-LEGO: creating metasearch engines on demand*, Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '03), 2003.
15. *Project DAISEn: Directory Architecture for Integrated Search Engines*, `http://daisen.cc.kyushu-u.ac.jp/`