

Generation of Query URL for Search Sites

Nakatoh, Tetsuya

Computing and Communications Center, Kyushu University

Koga, Yasunori

Graduate School of Information Science and Electrical Engineering, Kyushu University

Sakai, Miyuki

Graduate School of Information Science and Electrical Engineering, Kyushu University

Hirokawa, Sachio

Computing and Communications Center, Kyushu University

<https://hdl.handle.net/2324/1546305>

出版情報 : Proceedings of SSGRR 2001, 2001

バージョン :

権利関係 :

Generation of Query URL for Search Sites

Tetsuya Nakatoh, Miyuki Sakai, Yasunori Koga, Sachio Hirokawa

Abstract— There are increasing number of sites that proved search facility of their own. They are a kind of databases open to public with HTML interface, and are referred as Invisible Web. We are developing a system, which integrates these specialized search sites for user's purpose. A solution is the automatic wrapper generation. In this paper, we show how we can extract the attributes of the query parameters to construct a query URL for each site.

Keywords— Wrapper, Search Engine, Query Form, Query URL

I. INTRODUCTION

The flood of information on the Internet is a serious problem for people and companies. Search engines are keys to get rid of this flood of information. We use search engines, e.g., Yahoo!, Alta-vista, google, which search for the information we need over the WWW. One of the problems of general search engines is the quality of search result. The search results tend to contain many irrelevant pages. On the other hand, many companies are providing their own information with their own search engines[7]. We call such web sites as "Search Sites" compared with general search engines. A search site of a company focuses on their information and the quality is guaranteed.

The number of such search sites is increasing rapidly. The next problem we face is that we have to visit many search sites one by one to collect all information we need. A solution is the integration of search sites for each purpose. Fig. 1 shows a typical example of integration of search sites of electronic companies, Sony, Panasonic, Victor, Hitachi and Pioneer that produce DVD Players. A user can search and compare DVD players with one interface.

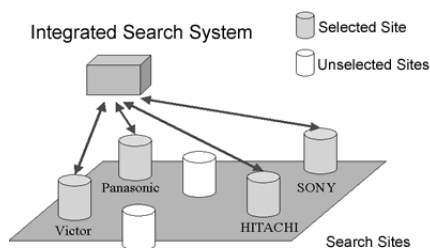


Fig. 1. Integration of Search Sites

CompletePlanet¹ estimates that there are more than 100,000 searchable databases available on the Web. To

T. Nakatoh and S. Hirokawa are with the Computing and Communications Center, Kyushu University, Hakozaki 6-10-1, Fukuoka 812-8581, JAPAN. E-mail:nakatoh@cc.kyushu-u.ac.jp. M. Sakai and Y. Koga are with Graduate School of Information Science and Electrical Engineering, Kyushu University.

¹<http://www.completeplanet.com/>

integrate these search sites we need to conceal the difference of the sites. But the query forms of these search sites vary. The metasearch engines integrate a small number of targets using manually written wrappers. But the fast expansion and change of WWW is beyond our manual effort. Automation of all process is necessary and the followings are main problems to achieve the automatic integration of search sites.

1. Pattern extraction of search result
2. Automatic generation of query form
3. Feature extraction of search sites for site selection
4. Interface for integration of search result

1 and 2 are necessary for automatic wrapper generation. 3 and 4 are necessary to integration many sites. For 1, we developed a wrapper generation method in [8] and [3]. As for 3 and 4, we proposed a framework for feature extraction of search sites in [1] and [5]. In this paper, we propose a method of the automatic generation of the query URL. With these methods, we can integrate search sites if we are given a list of sites for integration.

II. ACTUAL SEARCH SITES

The screens of Search Site vary. But, fundamental screen composition at Search Sites looks like a Fig. 2.



Fig. 2. The example of Search Site

Generally two of the next are indispensable for the search.

- Key word input part (1)
- The submit button part (2)

It is added, and the next thing is spent well, too.

- The check box(3).
- Radio button
- Pulldown menu

These are used for a search range and the kind of the database, the control in indication form, and so on. Generally, a part to accept input with WebPage is called a form.

The HTML source of the Fig.2 is simplified and shown in the TABLE.I.

TABLE I
HTML SOURCE

```
<FORM name="seek1" method=GET
  action="http://search.panasonic.com/polnav_query.html">

<INPUT type=checkbox name=col value="pol" checked>Entire Site
<INPUT type=checkbox name=col value="store" checked>Select Direct
<INPUT type=checkbox name=col value="bldgcomm">Building Commercial
<INPUT type=checkbox name=col value="brdcast">Broadcast CCTV Vision Systems
<INPUT type=checkbox name=col value="busoffic">Business Office
<INPUT type=checkbox name=col value="computer">Computer Products
<INPUT type=checkbox name=col value="consumer">Consumer Electronics
<INPUT type=checkbox name=col value="industrl">Industrial Components
<INPUT type=checkbox name=col value="notebook">Computer Notebook
<INPUT type=checkbox name=col value="press">Press Releases
<INPUT type=checkbox name=col value="support"> Customer Support (BR)

<INPUT type=hidden name=ht value="0">
<INPUT type=hidden name=qp value="">
<INPUT type=hidden name=qs value="">
<INPUT type=hidden name=ht value=0>
<INPUT type=hidden name=qp value="">
<INPUT type=hidden name=qs value="">

<INPUT type="text" name=txtqt>
<INPUT type="image" img src="seek.gif" alt="Start Search" name="Submit">

</FORM>
```

We describe about these each elements independently in the following.

A. FORM element

A form begins in `<FORM>` tag, and ends in `</FORM>` tag.

A form has some control to accept user's input, and transmits that information to the server.

With this example, this form has three attributes.

1. name attribute
 - name="seek1"
 - A name is given to this form.
2. method attribute
 - method=GET
 - How to send data to WebServer is specified.
3. action attribute
 - action = "http://search.panasonic.com/polnav_query.html"
 - The destination URL of the data is specified.

B. INPUT element

A `<INPUT>` tag create the parts which accept input from the user on the screen.

Some different control corresponding to a method to use for a user's inputting data is used properly by the type attribute.

1. type=text
 - Create a text input field.
 - As for this example, inputted data are handle with the variable name "txtqt".
2. type=checkbox
 - Create a checkbox.
 - The checkbox is on/off switch that may be toggled by the user.
 - 11 checkboxes are created with this example.

- A CHECKBOX with CHECKED is formed under the condition of "on" from the beginning.
 - When a submission button is pushed, the "name=value" pair of the check box which is on is actually submitted. In this sample, "col=pol" and "col=store" are returned.
3. type=hidden
 - It isn't indicated on the WEB screen.
 - The data that it doesn't need to show to the user are specified.
 - It is submitted to the server with other items at the time of submit of FORM.
 4. type=image
 - Creates a graphical submit button.
 - When the entry of the necessary information is finished, a user clicks on this button, and submits data to the server.

We write "DVD" in the text input field, and push the submit button. The next URL is formed and sent to the server by GET.

```
http://search.panasonic.com/polnav_query.html?
col=pol&col=store&txtqt=DVD&qp=&qs=&qc=
&pw=100%25&ws=1&qm=0&st=1&nh=10&lk=1&rf=0
&oq=&rq=0&qt=+dvd&image1.x=22&image1.y=15
```

We can get the page of the Fig.3 as a result.

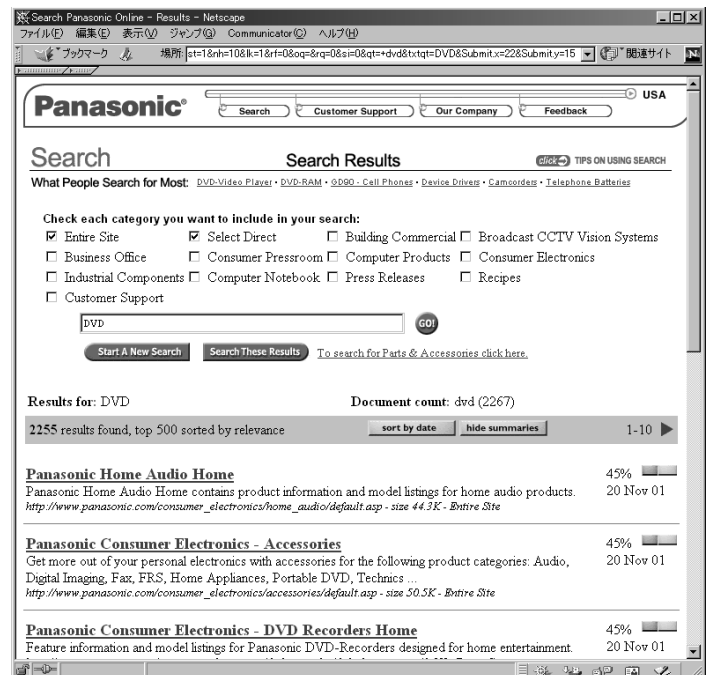


Fig. 3. The example of Search Result

III. THE INVESTIGATION OF SOME ACTUAL FORM

We investigated the matter whether the actual Search Site described a form how. We are collecting Search Site URL's of 852 by now. A change is drastic in Search Site in that nature. 553 site was effective in 852 site as Search Site by last investigation.

A. FORM element

717 forms were found in the 553 site. Because there is SearchSite which has more than one form, there are more forms than the number of sites. ALL FORMs have some attribute.

A.1 METHOD attribute

METHOD attribute specifies how to send form data to WebServer. Two kinds of values of “POST” and “GET” can be specified. When this attribute is omitted, it is considered being “POST”.

<i>METHOD</i>	<i>The number of appearances</i>
POST	183
GET	498
Omission (GET)	36
Total	717

A.2 ACTION attribute

ACTION attribute specifies a destination to send form data to WebServer. Generally the URL of the cgi script is specified. In the URL, there are a Relative-URL to specify only the pass of the resource inside the host, and an absolute-URL that contain a host name to begin “http://”. There are a Relative-Pass and an Absolute-Pass in the pass of the resources specified in the Relative-URL as well.

<i>Full-URL</i>	<i>Relative-URL</i>		invalid URL
	Absolute-Pass	Relative-Pass	
247	352	115	3

A.3 ENCTYPE attribute

ENCTYPE attribute specifies the method which encodes form data. As for Search Site, “application/x-www-form-urlencoded” is the suitable way of encoding it. “application/x-www-form-encoded” is thought to be the actually same way of encoding it, too.

<i>ENCTYPE</i>	<i>The number of appearances</i>
application/x-www-form-urlencoded	10
application/x-www-form-encoded	5
Omission (application/x-www-form-urlencoded)	702
total	717

A.4 ACCEPT-CHARSET attribute

This attribute specifies the list of character encodings for input data that is accepted by the server processing this form. It was only 7 sites that this attribute was specified.

<i>accept-charset</i>	<i>The number of appearances</i>
iso-8859-1	1
iso-2022-jp, EUC-JP	3
shift_jis	3

A.5 Others attribute

The following attribute appeared except for the thing put together in the above.

“name,target,class,id, onsubmit” These are used by the style seat and the script.

“style, align” These do designation about the indication.

There were some things that were not in the WWWC advice which it thought of with the mistake or typo.

B. INPUT element

3672 INPUT element’s existed in 717 forms. The appearance rate of type attribute is as mentioned in the next table. The default value of type attribute is text. Therefore, when a type attribute is omitted, it is considered being text.

<i>attribute</i>	<i>The number of appearances</i>
text	611
omission(text)	67
checkbox	552
radio	133
submit	613
hidden	1483
reset	85
button	35
password	5
image	88
total	3672

It is seen independently in the following.

B.1 type=text

The form of Search Site has one and more text input field.

There was the following thing in specified attributes.

<i>attribute</i>	<i>The number of appearances</i>
name	678
size	661
value	226
maxlength	89
style	13
id	8
class	17
accesskey	3
tabindex	1

Attribute which is indispensable to the automatic search is only “name”. About others, “maxlength” restricts the length of the string. And, it has the possibility to use class and id for the distinction.

B.2 type=checkbox

86 forms had checkbox, and there were 552 checkboxes in that. The form with most checkboxes had 84, and 1 form has 6.4 checkbox on the average. As for checkbox, it is important the choice of a database to use is often used.

<i>attribute</i>	<i>The number of appearances</i>
name	552
value	543
checked	193
id	11
accesskey	9
tabindex	9
onclick	2

Attributes which is indispensable to the automatic search are “name” and “value”. It is added, and checked is used for the confirmation of the initial value.

B.3 type=hidden

An INPUT element with “type=hidden” isn’t indicated. But, it is important to intend often to send information to cgi script and so on. Though 1483 hidden were found, all has only name attribute and value attribute.

B.4 radio button

133 “radio” appear in 48 forms, and the group of 51 is constructed. The group of 22 is using it for switching of and/or of more than one key word. The group of 12 is using it for the designation of the search range. A judgment was difficult for other groups.

The group of 47 has the CHECKED INPUT element. Therefore, the automatic search uses that.

The attributes that it appears are name, value and checked.

B.5 submit and image

A presentation button is constructed. More than one surely exists in one form. Image creates a graphical submit button.

It accesses a server by the automatic search without using the function of submit. Because of this, submit-button doesn’t influence it very much.

<i>attribute</i>	<i>The number of appearances</i>
submit	613
image	88

B.6 reset

The contents written in the form are erased, and it is returned to the initial condition. It isn’t concerned by the automatic reference.

<i>attribute</i>	<i>The number of appearances</i>
reset	85

B.7 button

35 buttons is appeared. It is an event trigger to script to work with a browser.

B.8 password

5 password’s appeared.

A password input form is formed. It can’t be only seen in screen of the browser, and it doesn’t always encipher. It exists at the reference service site of the member limitation.

C. SELECT element and OPTION element

The SELECT element creates a menu. Each choice offered by the menu is represented by an OPTION element. A SELECT element must contain at least one OPTION element.

C.1 SELECT element

781 SELECT elements appeared in total. There were the 744 ones with name attribute, and the 79 ones with onchange attribute.

In others as well, there were the ones with the following attributes. ID, SIZE, class, style, tabindex, width.

C.2 OPTION element

4881 OPTION elements is appeared in total. All OPTION elements is appeared with name-attribute and value-attribute. 710 selected-attributes is appeared in 781 menus. And, the items of the contents of 781 menus were as the next.

The number of matters to indicate at a time is specified: 274. The item of the indication and length are specified: 211. The order of the indication is specified: 137. The source (database) of the search object is specified: 101. A logic-type (AND/OR) between more than one key word is specified: 12. A language is specified: 11. What can’t be judged: 35.

The formation of the form data and the presentation method of the form.

The formation of the form data.

form data are formed in accordance with the next process.

1. Control names and values are escaped.
2. Space characters are replaced by ‘+’.
3. Non-alphanumeric characters are replaced by ‘%HH’, a percent sign and two hexadecimal digits representing the ASCII code of the character.
4. Line breaks are represented as “CR LF” pairs (i.e., ‘%0D%0A’).
5. The control names/values are listed in the order they appear in the document.
6. The name is separated from the value by ‘=’ and name/value pairs are separated from each other by ‘&’.

IV. HOW TO GENERATE QUERY URL

We argue how to form query URL with using the result of the above investigation. Many elements appeared, and each element had the attribute of very many kinds. But, the item which should be necessary for the automatic search to Search Site is limited. For example, we don’t need to think about an element about the indication.

A. FORM element

Three kinds of attributes, METHOD, ACTION and ENCTYPE, appear in the form element. They are each independent.

A.1 NAME attribute

NAME attribute gives a form name. As for the automatic search, we don't need to refer to the name of the form. Therefore, we may ignore this item.

A.2 METHOD attribute

METHOD attribute specifies how to send form data to WebServer. Two kinds of values of "POST" and "GET" can be specified. When this attribute is omitted, it is considered being "POST". We will be able to pick out METHOD attribute easily from the form.

A.3 ACTION attribute

ACTION attribute specifies a destination to send form data to WebServer.

An Absolute-URL is necessary for the automatic search. Because of that, we complement the value of ACTION attribute from the URL of Search Site.

A.4 ENCTYPE attribute

ENCTYPE attribute specifies the method which encodes form data. The default value in the omission is "application/x-www-form-urlencoded". Even when there is designation, this value is almost "application/x-www-form-urlencoded". Therefore, we can ignore this item.

A.5 ACCEPT-CHARSET attribute

We weren't handle about this attribute by this paper.

B. INPUT element

An INPUT element indicates a text field, a checkbox and a radio button, and so on on the WEB screen. A user uses them, and sends information for the reference to WebServer.

B.1 attribute type="text"

An INPUT element with attribute type=text formed a text input field. This text input field is used for the input of the keyword in Search Site. If this doesn't appear, we can't input a keyword for the search. In that case, we can judge that the form is not for the search.

The name of this element is given by name attribute. We get the pair of the name and the inputted string as a result. For example, txtqt= "DVD".

B.2 type="checkbox"

An INPUT element with attribute type=checkbox formed a checkbox.

A checkbox has two condition of "on" (checked) and "off" (unchecked) by the user's input. The initial condition of checkbox is "off". But, the initial condition of checkbox becomes "on" by writing "checked" in INPUT element.

The name is given by name attribute as well as other elements. And, value is specified with value attribute, too. If checkbox is on, we can get the pair (for example, here, col= "pol") of the value which name attribute and value

attribute have. If value attribute was omitted, the value of that value is supposed to be "on".

Which checkbox should we check for the automatic search? This has very difficult argument. But, let's take the best plans temporarily. If there is checkbox which CHECKED from the beginning, we will adopt that. A site author is supposed to intend that. If there is no checkbox which CHECKED from the beginning, we will make all checkboxes on. In many cases, the database for search is because it is specified by checkbox.

There is a room of more argument in handling of checkbox. In the future, the structure of the distinction of the author's intention is necessary.

B.3 type= "radio"

An INPUT element with attribute type=radio formed a radio button. Some radio buttons which have the same name become a group. We can turn "on" only one radio button in the group.

Which radio button should we turn "on" for the automatic search? It is a difficult problem as well as the case of the checkbox. Let's take the best plans temporarily, too. We do a radio button according to the first condition.

In case of most, only one radio button which a author intends is "on". Even when every radio button is not "on", we actually could do a search.

C. SELECT element and OPTION element

Menus offer users options from which to choose. The SELECT element creates a menu, in combination with the OPTION elements. A database is often specified by this menu. Or, an indication form, the number of matters, and so on are sometimes specified. The same argument as checkbox and radio-button is necessary.

We will choose that option if there is option chosen from the beginning. When every option isn't chosen, we will choose the first option. This follows [RFC1866].

V. THE PRESENTATION METHOD OF THE FORM

A. In case of METHOD=GET

With the HTTP "get" method, the form data set is appended to the URI specified by the action attribute (with a question-mark ("?") as separator) and this new URI is sent to the processing agent.

B. In case of METHOD=POST

With the HTTP "post" method, the form data set is included in the body of the form and sent to the processing agent.

VI. EXPERIMENT

We are collecting the URL's of 852 Search Site by now. Query form formation was tested to them. A change is drastic in Search Site in that nature. First, those URL's confirmed whether it was Search Site in the present as well. The existence of form was used as an index of what keeps being Search Site. We could confirm that 553 Search Site

TABLE II
QUERY URL

ID	method	Query-URL	query name
2	get	http://akari.nfri.affrc.go.jp/namazu/namazu.cgi?whence=0&max=20&result=normal&sort=score&	query
3	GET	http://aoki2.si.gunma-u.ac.jp/NMZ/BotanicalGarden/namazu.cgi?whence=0&max=20&result=normal&sort=score&	key
4	GET	http://aoki2.si.gunma-u.ac.jp/NMZ/Statistics/namazu.cgi?whence=0&max=20&result=normal&sort=score&	key
5	GET	http://apacheml.ecc.u-tokyo.ac.jp/cgi-bin/namazu.cgi?whence=0&max=20&format=long&sort=score&	key
13	GET	http://biore.co.jp/namazu.cgi?whence=0&max=20&format=long&sort=score&	key
16	get	http://bpc49.narc.affrc.go.jp/faps/ML/namazu.cgi?whence=0&max=20&format=long&sort=score&	key
18	GET	http://camellia.fukuyama.hiroshima-u.ac.jp/cgi-bin/namazu.cgi?whence=0&max=20&format=long&	key
22	GET	http://cgi3.osk.3web.ne.jp/%7Easataku/namazu/namazu.cgi?whence=0&max=20&format=long&sort=score&	key
23	GET	http://cgisv.children.net/~myamya/pgp-verification/namazu.cgi?max=20&format=long&whence=0&	key
24	GET	http://chiringi.or.jp/k_library/namazu.cgi?whence=0&max=20&format=long&sort=score&	key
25	GET	http://clug.linux.or.jp/ml-archive/namazu.cgi?whence=0&max=20&format=long&sort=score&	key
32	GET	http://dennou.gaia.h.kyoto-u.ac.jp/cgi-bin/namazu.cgi?whence=0&max=20&format=long&dbname=bunken&	key
34	GET	http://dennou-t.ms.u-tokyo.ac.jp/cgi-bin/namazu.cgi?whence=0&max=20&format=long&dbname=dcusers&	key
35	get	http://doc.medic.mie-u.ac.jp/cgi-bin/namazu.cgi?whence=0&max=20&format=long&sort=score&dbname=graduate&	key
39	GET	http://alpha.fine.chiba-u.ac.jp:8080/~nagasaki/human-search/namazu.cgi?max=20&format=long&whence=0&	key
40	GET	http://feya.city.tokushima.jp/cgi-bin/fsearch/fsearch.cgi?from=0&n=20&index=default&n=20&	key
43	GET	http://grape.c.u-tokyo.ac.jp/~nakano/namazu.cgi?whence=0&max=20&format=long&sort=score&	key

existed in 852 site at present as a result. The URL that is not Search Site is classified in two kinds of the next. That URL stopped existing, or form disappeared from page of that URL.

Query form was formed about 553 WebPage's, and an automatic search was done by using that. We could get a search result from 397 Search Site as that result. It is judged that proper query form was formed.

In other words, we couldn't get some proper search result form 156 SearchSites. The following reason was found.

- 100 : METHOD is POST.
- 14 : The problem of the character code in Japanese.
- 1 : Very long query URL.
- 41 : The problem on the WebServer side.

In last experiment, we couldn't get a good result in the METHOD which POST was used for. But, we know how to solve it.

The example of query form that it is formed is shown in the TABLE.II.

VII. RELATED WORK

Integration of multiple search engines is known as a metasearch engine [6]. Most of their targets are general search engines that may overlap each other. On the other hand, the targets of our project are independent search sites that do not overlap. They may be homogeneous, like competing electronic companies, or may be heterogeneous, like airlines, hotels and restaurants. The contents are qualified by each site, so that we do not need filtering and ranking to the search results. The wrappers used in metasearch engines are usually written manually. We are proposing an automatic generation of wrappers for search sites.

In [2], Ipeirotis et al. used the similar query probing method for feature extraction of text databases. But they used a single keyword and used the number of search result. We proposed a method for complex query and used the pattern extraction, which Ipeirotis et al. admits to be desirable.

Kushmerick et. al. [4] introduced a learning algorithm to generate a wrapper from several examples. Our wrapper generation [8] is based on the observation that the search

result contains repetition of the same tag sequence. So we do not need examples.

REFERENCES

- [1] S. Hirokawa, S. Watanabe, Y. Koga and T. Taguchi, *Automatic Feature Extraction of Search Sites*, Proc. SSGRR2001(CD-ROM).
- [2] P. Ipeirotis, L. Gravano and M. Sahami, *Automatic Classification of Text Databases through Query Probing*, Proc. of the ACM SIGMOD Workshop on the Web and Databases (WebDB'00), 2000.
- [3] Y. Koga, T. Taguchi and S. Hirokawa, *Wrapper Generation for Search Sites Integration(in Japanese)*, Proc. DEWS'01, 2001.
- [4] N. Kushmerick, D. Weld and B. Doorenbos, *Wrapper induction for information Extraction*, IJCAI'97, pp. 729-737, 1997.
- [5] T. Nakatoh, Y. Koga and S. Hirokawa, *Automatic Classification of Search Sites(in Japanese)*, Proc. DBWeb2001, pp. 225-228, 2001.
- [6] E. Selberg and O. Etzioni, *The MetaCrawler architecture for resource aggregation on the Web*, IEEE Expert, Vol.12, No.1, pp. 11-14, 1997.
- [7] C. Sherman and G. Price, *The Invisible Web*, Information Today, Inc, Medfore, New Jersey, 2001.
- [8] T. Taguchi, Y. Koga and S. Hirokawa, *Integration of Search Sites of the World Wide Web*, Proc. of International Forum cum Conference on Information Technology and Communication, Vol. 2, pp. 25-32, 2000.