ブックマークの自動管理システムの開発

03H074 中前嘉人

目次

1 はじめに 1

2 ユーザ 2

3 ABM 3
- 3.1 現在の ABM の問題点 . . . 3
- 3.2 表示方法 . . . 3
- 3.3 新機能 . . . 5

4 EBM 7
- 4.1 EBM とは . . . 7
- 4.2 EBM の登録方法 . . . 8
- 4.3 EBMng . . . 9

5 まとめ 10

付録 A 各プログラム 12
- A.1 アクセスログからユーザの ABM を作るプログラム . . . 12
- A.2 アクセスログから user 名を抽出 . . . 14
- A.3 EBM 登録 . . . 16

1　はじめに

現在インターネットは世界中の人が利用しており、いろいろな人がいろいろなことを調べている。調べていると、ブックマークに登録していない Web ページでも利用頻度が高い Web ページもある。そのような Web ページの URL を自動で収集し表示されるように出来ると便利になると考えられる。また、利用頻度が高くなくとも調べているときに重要な情報を得ることもある。それら全ての URL の履歴を自動で保存し表示することによってブックマーク登録の手間を省く。そうすることによって URL や検索するキーワードがわからなくなった場合でも容易に過去に見た欲しい Web ページの情報を手に入れることが可能になる。

Web ブラウザには Web のアクセス履歴を残す機能がある。しかし、使用した PC にしか履歴は残っていないので他の PC ではアクセス履歴を見ることができない。EBMng は研究室外でも使用できるので Web ブラウザの機能より使用し易くなる。情報安全工学研究室内では web にアクセスするのに proxy サーバを経由させているので proxy サーバに web ページのアクセス履歴が残る。本研究では proxy サーバに蓄積されているアクセス履歴を使用して研究を進める。情報安全工学研究室内で使用している proxy サーバソフトウェアは squid というソフトウェアを使用している。squid[*1]には ident デーモン[*2]を使用して web ページ毎に閲覧したユーザ名を記録する機能があるので、この機能を使いユーザを識別する。

情報安全工学研究室では AutoBookMark(ABM)[*3]という仕組みがある。ABM を使えば自分が web ページにアクセスしたら自動的にログを蓄積させ自動で web 上にブックマークを作成することができる。しかし、現在の ABM の設定では他のユーザの履歴を閲覧することが出来ないようになっており情報が共有することが出来ないという面がある。

ユーザが一度見た web ページというのは他のユーザにとっては検索エンジンで検索して出てくる結果より利用価値があると考えられる。何故ならば検索したい事柄に関係ないと思われるものは既に省かれているからである。つまり各ユーザが簡単なフィルタの役目を果たしているということになる。余分なものを省いた履歴を見ることが出来るようになれば自分にとって有益な情報を素早く得ることが出来る可能性がある。自分と同じ事について調べている人から情報を得たり、自分が興味を持っていなかった事について他者の履歴をきっかけにして興味を持つ可能性もある。それらの可能性を試すために ABM の仕様変更をする。

情報安全工学研究室内では ABM の他に Everybody'sBookMark(EBM)[*4]というブックマークを管理する仕組みがある。EBM とは ABM を情報共有に特化させたものである。本研究では ABM と EBM という情報安全工学卒業研究室の先輩が作った二つのシステムを利用して新しいブックマーク管理システム EBMng 開発することを目的とする。

[*1] proxy サーバソフトウェア

[*2] 接続したサーバーのユーザー情報について、付加的な情報を返すためのサーバー。

[*3] ABM とはユーザ毎のアクセスログの履歴を蓄積させ、web ページ上に自分専用のブックマークを作成する仕組み。

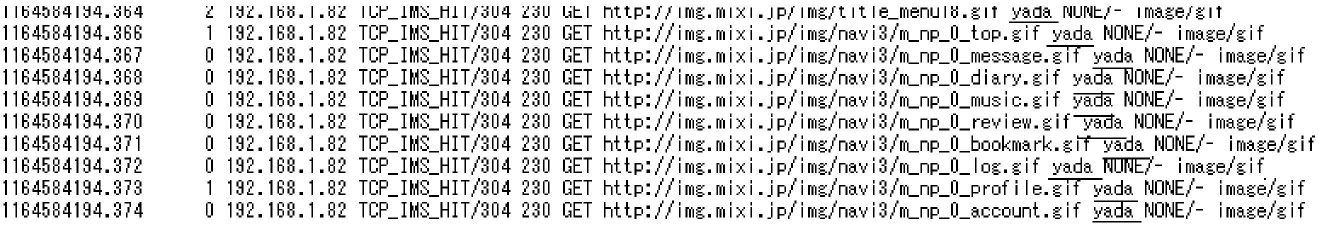
[*4] 複数のユーザが自分のお気に入りの web ページを登録して、Book Mark を共有できるようにしたもの。

2 ユーザ

今までの ABM では REMOTE_INDENT*[5]を他のユーザの ABM を見ることは出来なかった。また、を使用しているために研究室外では ABM を見ることが出来ない。情報共有の為には他のユーザの履歴も見れた方が情報が共有出来ると考えられる。しかし、REMOTE_INDENT を使用しない場合はユーザは手動で登録しないといけなくなる。手動になってしまうと登録されるまで新規ユーザは ABM を使えなくなってしまう。

この問題を解決する方法として研究室内の PC では ident デーモンが起動しているので研究室内の PC を使用して web ページにアクセスすると、squid のアクセスログにユーザ名の記録された履歴が残る。そこにあるユーザ名だけを抽出すれば研究室内のネットワークを使用して web ページにアクセスしたユーザのユーザ名が分かる。新しいユーザを見つけるとユーザを自動で登録することで解決した。

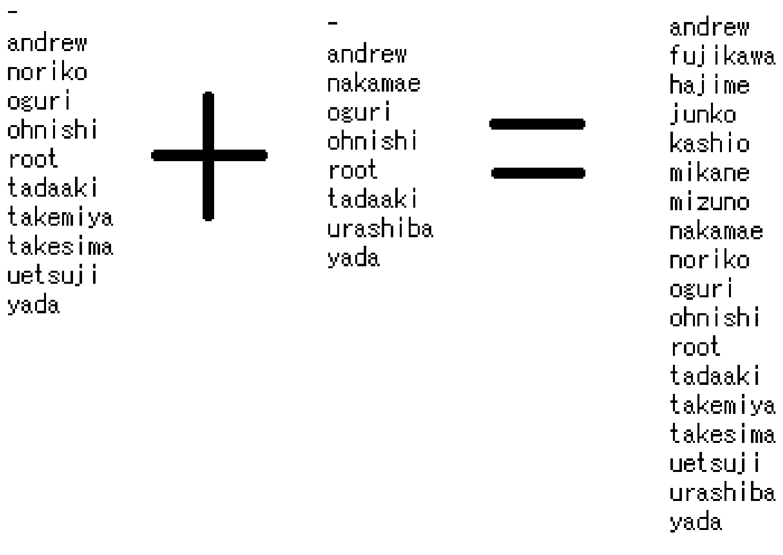
自動で ABM にユーザ登録するようにしたことで簡単に ABM を使用できるようになった。また研究室の新規ユーザが増えた場合はすぐにわかるようになった。



```
1164584194.364      2 192.168.1.82 TCP_IMS_HIT/304 230 GET http://img.mixi.jp/img/title_menu18.gif yada NONE/- image/gif
1164584194.366      1 192.168.1.82 TCP_IMS_HIT/304 230 GET http://img.mixi.jp/img/navi3/m_np_0_top.gif yada NONE/- image/gif
1164584194.367      0 192.168.1.82 TCP_IMS_HIT/304 230 GET http://img.mixi.jp/img/navi3/m_np_0_message.gif yada NONE/- image/gif
1164584194.368      0 192.168.1.82 TCP_IMS_HIT/304 230 GET http://img.mixi.jp/img/navi3/m_np_0_diary.gif yada NONE/- image/gif
1164584194.369      0 192.168.1.82 TCP_IMS_HIT/304 230 GET http://img.mixi.jp/img/navi3/m_np_0_music.gif yada NONE/- image/gif
1164584194.370      0 192.168.1.82 TCP_IMS_HIT/304 230 GET http://img.mixi.jp/img/navi3/m_np_0_review.gif yada NONE/- image/gif
1164584194.371      0 192.168.1.82 TCP_IMS_HIT/304 230 GET http://img.mixi.jp/img/navi3/m_np_0_bookmark.gif yada NONE/- image/gif
1164584194.372      0 192.168.1.82 TCP_IMS_HIT/304 230 GET http://img.mixi.jp/img/navi3/m_np_0_log.gif yada NONE/- image/gif
1164584194.373      1 192.168.1.82 TCP_IMS_HIT/304 230 GET http://img.mixi.jp/img/navi3/m_np_0_profile.gif yada NONE/- image/gif
1164584194.374      0 192.168.1.82 TCP_IMS_HIT/304 230 GET http://img.mixi.jp/img/navi3/m_np_0_account.gif yada NONE/- image/gif
```

図 1　access log

squid のアクセスログの一部分。このアクセスログからコンソールを使用して web ページを見たユーザを抽出していくことによって全てのユーザの ABM を作成することができる。



```
-
andrew            -                  andrew
noriko            andrew             fujikawa
oguri             nakamae            hajime
ohnishi           oguri              junko
root        +     ohnishi      =     kashio
tadaaki           root               mikane
takemiya          tadaaki            mizuno
takesima          urashiba           nakamae
uetsuji           yada               noriko
yada                                 oguri
                                     ohnishi
                                     root
                                     tadaaki
                                     takemiya
                                     takesima
                                     uetsuji
                                     urashiba
                                     yada
```

図 2　ユーザ処理の例

一日目に情報安全工学研究室内のコンソールを使ったユーザと二日目に使ったユーザのユーザ名を抽出し一日目にいなかったユーザ名を結合させる。

---

[5] リクエストを発行したユーザ側の ident デーモンによってユーザ名が通知される。

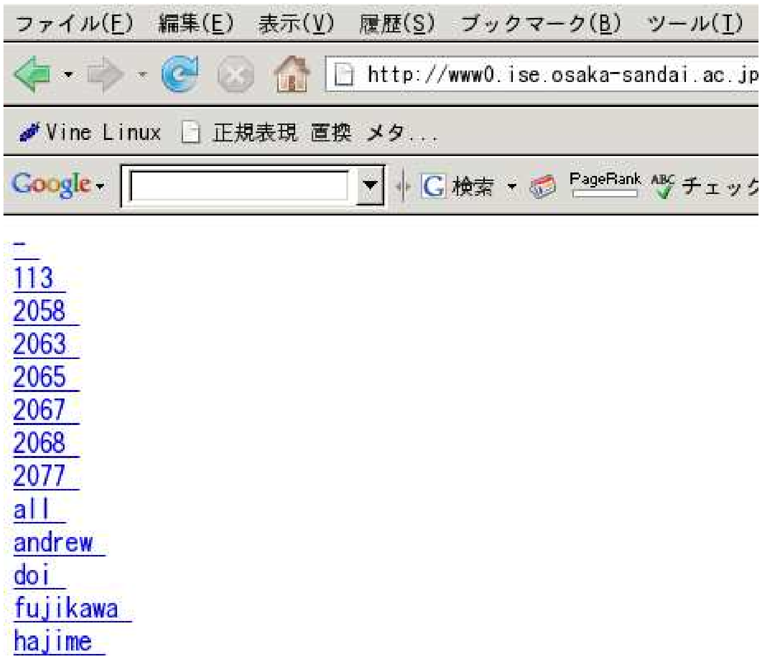
3　ABM

3.1　現在の ABM の問題点

REMOTE_INDENT を使用しているために研究室外では ABM を見ることが出来ない。
大量の履歴が存在しているので、その中から欲しい情報だけを探すには時間がかかってしまう。

3.2　表示方法

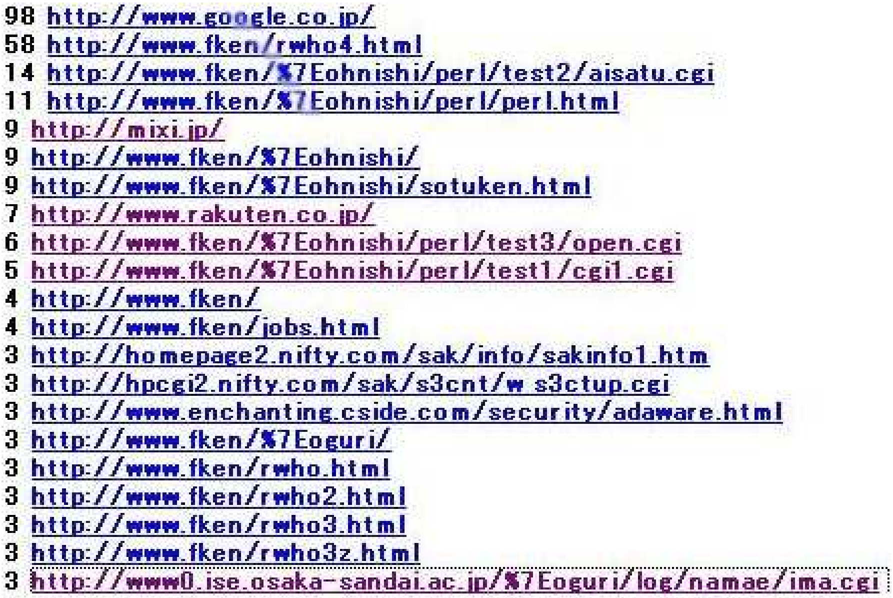
図 3 はユーザの名前を選択する画面である。ユーザの名前を選択すると、そのユーザの ABM が表示される仕組みになっている。


ファイル(F)　編集(E)　表示(V)　履歴(S)　ブックマーク(B)　ツール(T)
http://www0.ise.osaka-sandai.ac.jp
Vine Linux　正規表現　置換　メタ...
Google　検索　PageRank　チェック

_
113
2058
2063
2065
2067
2068
2077
all
andrew
doi
fujikawa
hajime

図 3　user 選択画面

図 4 のように閲覧回数の高いものから順に表示させた。閲覧回数の高いものから表示させた理由としては閲覧回数の高い URL ほど利用頻度が高く重要な情報があると考えたからである。



```
98 http://www.google.co.jp/
58 http://www.fken/rwho4.html
14 http://www.fken/%7Eohnishi/perl/test2/aisatu.cgi
11 http://www.fken/%7Eohnishi/perl/perl.html
9 http://mixi.jp/
9 http://www.fken/%7Eohnishi/
9 http://www.fken/%7Eohnishi/sotuken.html
7 http://www.rakuten.co.jp/
6 http://www.fken/%7Eohnishi/perl/test3/open.cgi
5 http://www.fken/%7Eohnishi/perl/test1/cgi1.cgi
4 http://www.fken/
4 http://www.fken/jobs.html
3 http://homepage2.nifty.com/sak/info/sakinfo1.htm
3 http://hpcgi2.nifty.com/sak/s3cnt/w s3ctup.cgi
3 http://www.enchanting.cside.com/security/adaware.html
3 http://www.fken/%7Eoguri/
3 http://www.fken/rwho.html
3 http://www.fken/rwho2.html
3 http://www.fken/rwho3.html
3 http://www.fken/rwho3z.html
3 http://www0.ise.osaka-sandai.ac.jp/%7Eoguri/log/namae/ima.cgi
```

図 4　ABM の表示例

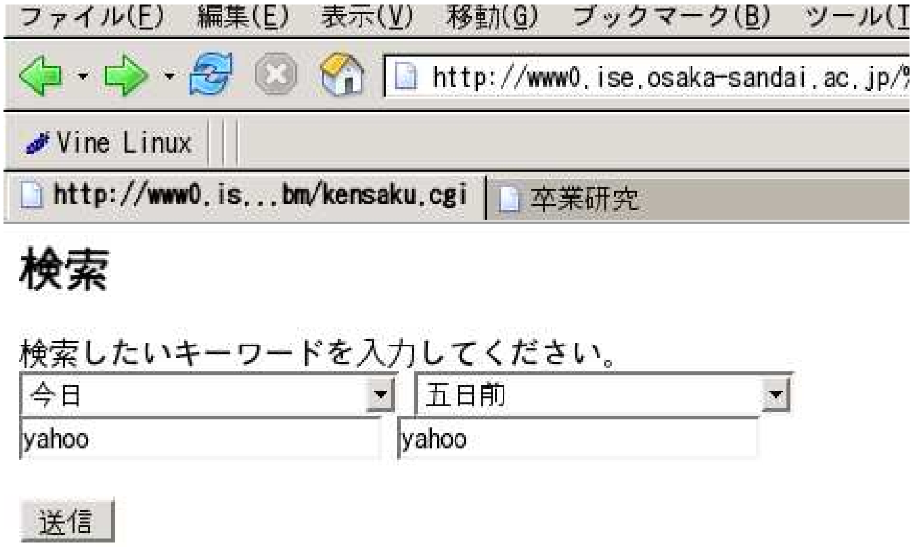
3.3　新機能

3.3.1　改良

他のユーザの ABM を自由に閲覧できるようにした。他のユーザが閲覧した web ページの中から知りたい情報に関連する情報を得ることで時間の節約に繋がるようになる。

3.3.2　検索機能

各ユーザの履歴は用意したので個人々々の ABM はあるがユーザ以外で調べたい場合も出てくる。例えば yahoo というキーワードについて調べたいと思ったときに研究室内で他の人が yahoo について調べている可能性がある。そういう場合の為に図 6 の例のように入力した文字列に対しての結果を返し閲覧回数も表示するようにした。


ファイル(F)
編集(E)
表示(V)
移動(G)
ブックマーク(B)
ツール(T
http://www0.ise.osaka-sandai.ac.jp/%
Vine Linux
http://www0.is...bm/kensaku.cgi
卒業研究
送信

検索

検索したいキーワードを入力してください。

今日　五日前

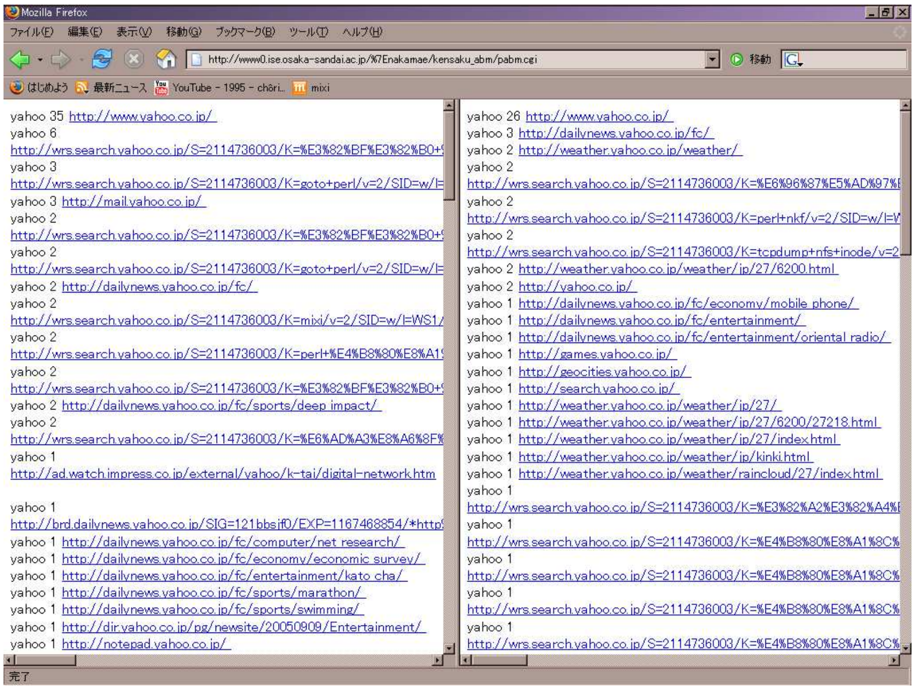
yahoo　yahoo

図 5　検索画面

3.3.3　検索結果の比較

HTML の Frame 機能を使って一つの画面に 2 つの検索結果を表示できるようにする。2 つ表示することによって 1 日前と 4 日前との履歴の比較やユーザ A とユーザ B との比較などが簡単に出来るようになる。

この機能によって研究室内の興味の違いなどを見易く表示できるようになった。



http://www0.ise.osaka-sandai.ac.jp/%7Enakamae/kensaku_abm/pabm.cgi

```
yahoo 35 http://www.yahoo.co.jp/
yahoo 6
http://wrs.search.yahoo.co.jp/S=2114736003/K=%E3%82%BF%E3%82%B0+
yahoo 3
http://wrs.search.yahoo.co.jp/S=2114736003/K=goto+perl/v=2/SID=w/l=
yahoo 3 http://mail.yahoo.co.jp/
yahoo 2
http://wrs.search.yahoo.co.jp/S=2114736003/K=%E3%82%BF%E3%82%B0+
yahoo 2
http://wrs.search.yahoo.co.jp/S=2114736003/K=goto+perl/v=2/SID=w/l=
yahoo 2 http://dailynews.yahoo.co.jp/fc/
yahoo 2
http://wrs.search.yahoo.co.jp/S=2114736003/K=mixi/v=2/SID=w/l=WS1/
yahoo 2
http://wrs.search.yahoo.co.jp/S=2114736003/K=perl+%E4%B8%80%E8%A1
yahoo 2
http://wrs.search.yahoo.co.jp/S=2114736003/K=%E3%82%BF%E3%82%B0+
yahoo 2 http://dailynews.yahoo.co.jp/fc/sports/deep impact/
yahoo 2
http://wrs.search.yahoo.co.jp/S=2114736003/K=%E6%AD%A3%E8%A6%8F%
yahoo 1
http://ad.watch.impress.co.jp/external/yahoo/k-tai/digital-network.htm

yahoo 1
http://brd.dailynews.yahoo.co.jp/SIG=121bbsjf0/EXP=1167468854/*http
yahoo 1 http://dailynews.yahoo.co.jp/fc/computer/net research/
yahoo 1 http://dailynews.yahoo.co.jp/fc/economy/economic survey/
yahoo 1 http://dailynews.yahoo.co.jp/fc/entertainment/kato cha/
yahoo 1 http://dailynews.yahoo.co.jp/fc/sports/marathon/
yahoo 1 http://dailynews.yahoo.co.jp/fc/sports/swimming/
yahoo 1 http://dir.yahoo.co.jp/pg/newsite/20050909/Entertainment/
yahoo 1 http://notepad.yahoo.co.jp/
```

```
yahoo 26 http://www.yahoo.co.jp/
yahoo 3 http://dailynews.yahoo.co.jp/fc/
yahoo 2 http://weather.yahoo.co.jp/weather/
yahoo 2
http://wrs.search.yahoo.co.jp/S=2114736003/K=%E6%96%87%E5%AD%97%
yahoo 2
http://wrs.search.yahoo.co.jp/S=2114736003/K=perl+nkf/v=2/SID=w/l=
yahoo 2
http://wrs.search.yahoo.co.jp/S=2114736003/K=tcpdump+nfs+inode/v=2
yahoo 2 http://weather.yahoo.co.jp/weather/jp/27/6200.html
yahoo 2 http://yahoo.co.jp/
yahoo 1 http://dailynews.yahoo.co.jp/fc/economy/mobile phone/
yahoo 1 http://dailynews.yahoo.co.jp/fc/entertainment/
yahoo 1 http://dailynews.yahoo.co.jp/fc/entertainment/oriental radio/
yahoo 1 http://games.yahoo.co.jp/
yahoo 1 http://geocities.yahoo.co.jp/
yahoo 1 http://search.yahoo.co.jp/
yahoo 1 http://weather.yahoo.co.jp/weather/jp/27/
yahoo 1 http://weather.yahoo.co.jp/weather/jp/27/6200/27218.html
yahoo 1 http://weather.yahoo.co.jp/weather/jp/27/index.html
yahoo 1 http://weather.yahoo.co.jp/weather/jp/kinki.html
yahoo 1 http://weather.yahoo.co.jp/weather/raincloud/27/index.html
yahoo 1
http://wrs.search.yahoo.co.jp/S=2114736003/K=%E3%82%A2%E3%82%A4%
yahoo 1
http://wrs.search.yahoo.co.jp/S=2114736003/K=%E4%B8%80%E8%A1%8C%
yahoo 1
http://wrs.search.yahoo.co.jp/S=2114736003/K=%E4%B8%80%E8%A1%8C%
yahoo 1
http://wrs.search.yahoo.co.jp/S=2114736003/K=%E4%B8%80%E8%A1%8C%
yahoo 1
http://wrs.search.yahoo.co.jp/S=2114736003/K=%E4%B8%80%E8%A1%8C%
```

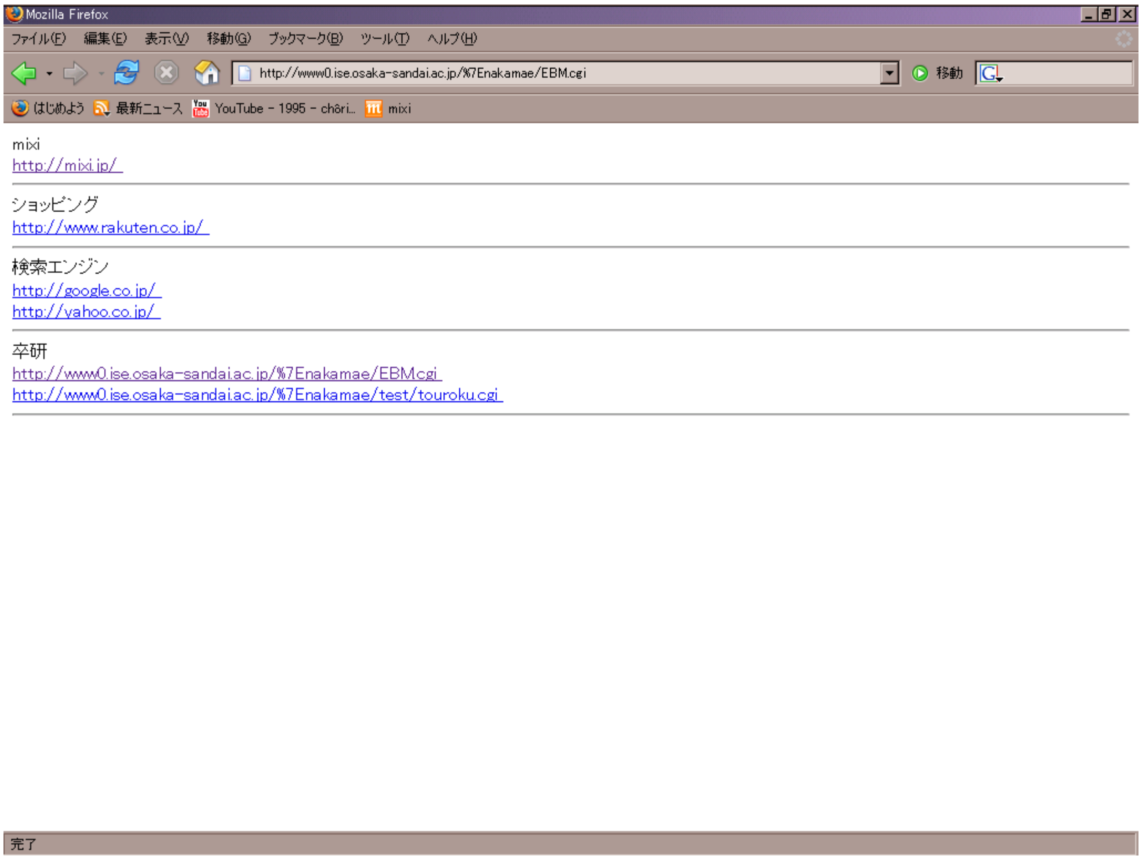
図 6　検索結果

4　EBM

4.1　EBM とは

ユーザが役に立つと思った web ページを適したカテゴリに登録しておくことが出来るようになってある。この機能でユーザが重要だと思った web ページだけを登録することが出来る。

EBM を使えば複数のユーザで一つのブックマークを作ることが出来る。



Mozilla Firefox
ファイル(F)　編集(E)　表示(V)　移動(G)　ブックマーク(B)　ツール(T)　ヘルプ(H)
http://www0.ise.osaka-sandai.ac.jp/%7Enakamae/EBM.cgi
移動
はじめよう　最新ニュース　YouTube - 1995 - chôri...　mixi
完了

mixi
http://mixi.jp/

ショッピング
http://www.rakuten.co.jp/

検索エンジン
http://google.co.jp/
http://yahoo.co.jp/

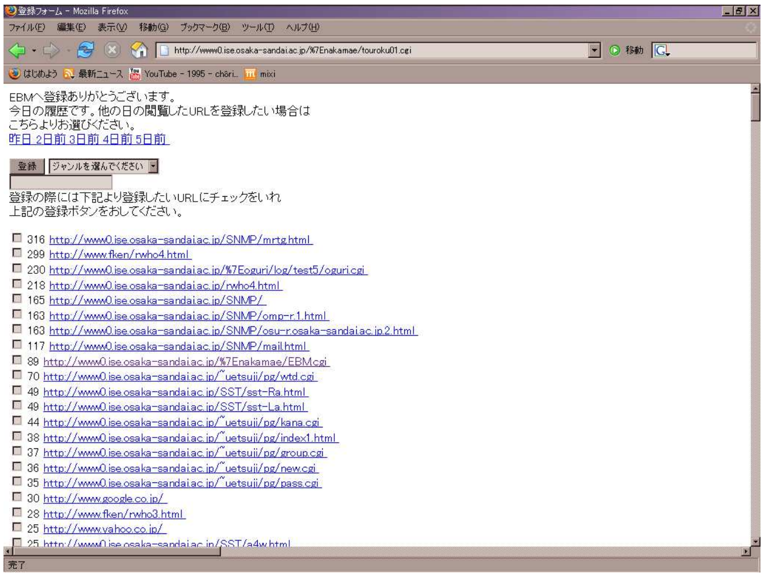
卒研
http://www0.ise.osaka-sandai.ac.jp/%7Enakamae/EBM.cgi
http://www0.ise.osaka-sandai.ac.jp/%7Enakamae/test/touroku.cgi

図 7　EBM の表示画面

4.2　EBM の登録方法

登録方法は URL を直接入力して EBM に登録する方法と、アクセスログを利用して EBM に登録する方法がある。アクセスログを利用する方法は、現在のではプルダウンメニューでカテゴリを選択し、ABM の URL の側にラジオボタンを設置し、選択したものを登録できるようになっている。しかし、これでは一つずつしか登録できない。チェックボックスに変更して複数同時に登録できるように変更した。

カテゴリの中に無いものはユーザが自由に名前をつけて新しいカテゴリを作ることができる。より細かいカテゴリに分けることができるので興味のある情報をより簡単に得ることが出来るようになっている。


登録フォーム - Mozilla Firefox
ファイル(F)　編集(E)　表示(V)　移動(G)　ブックマーク(B)　ツール(T)　ヘルプ(H)
移動
はじめよう　最新ニュース　YouTube - 1995 - chōri...　mixi
登録
ジャンルを選んでください
完了

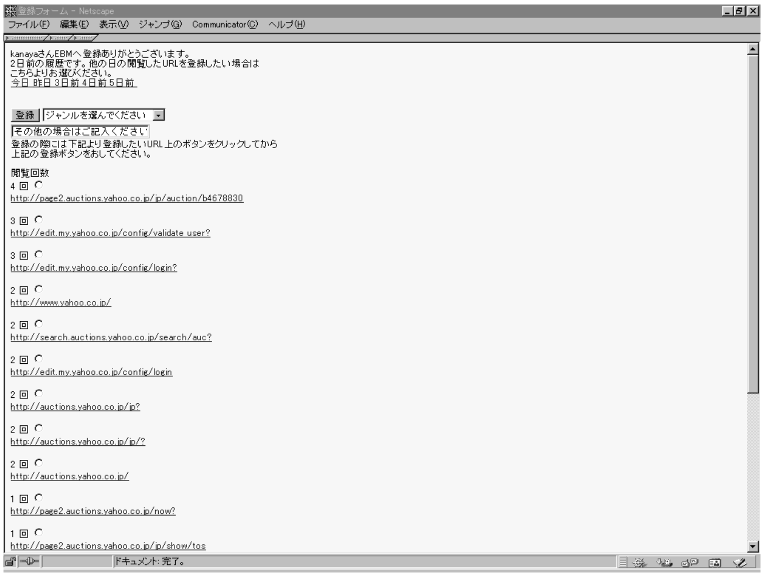
http://www0.ise.osaka-sandai.ac.jp/%7Enakamae/touroku01.cgi

EBMへ登録ありがとうございます。
今日の履歴です。他の日の閲覧したURLを登録したい場合は
こちらよりお選びください。
昨日 2日前 3日前 4日前 5日前

登録の際には下記より登録したいURLにチェックをいれ
上記の登録ボタンをおしてください。

☐ 316 http://www0.ise.osaka-sandai.ac.jp/SNMP/mrtg.html
☐ 299 http://www.fken/rwho4.html
☐ 230 http://www0.ise.osaka-sandai.ac.jp/%7Eoguri/log/test5/oguri.cgi
☐ 218 http://www0.ise.osaka-sandai.ac.jp/rwho4.html
☐ 165 http://www0.ise.osaka-sandai.ac.jp/SNMP/
☐ 163 http://www0.ise.osaka-sandai.ac.jp/SNMP/omp-r.1.html
☐ 163 http://www0.ise.osaka-sandai.ac.jp/SNMP/osu-r.osaka-sandai.ac.jp.2.html
☐ 117 http://www0.ise.osaka-sandai.ac.jp/SNMP/mail.html
☐ 89 http://www0.ise.osaka-sandai.ac.jp/%7Enakamae/EBM.cgi
☐ 70 http://www0.ise.osaka-sandai.ac.jp/~uetsuji/pg/wtd.cgi
☐ 49 http://www0.ise.osaka-sandai.ac.jp/SST/sst-Ra.html
☐ 49 http://www0.ise.osaka-sandai.ac.jp/SST/sst-La.html
☐ 44 http://www0.ise.osaka-sandai.ac.jp/~uetsuji/pg/kana.cgi
☐ 38 http://www0.ise.osaka-sandai.ac.jp/~uetsuji/pg/index1.html
☐ 37 http://www0.ise.osaka-sandai.ac.jp/~uetsuji/pg/group.cgi
☐ 36 http://www0.ise.osaka-sandai.ac.jp/~uetsuji/pg/new.cgi
☐ 35 http://www0.ise.osaka-sandai.ac.jp/~uetsuji/pg/pass.cgi
☐ 30 http://www.google.co.jp/
☐ 28 http://www.fken/rwho3.html
☐ 25 http://www.yahoo.co.jp/
☐ 25 http://www0.ise.osaka-sandai.ac.jp/SST/a4w.html

図 8　EBM の登録画面 (新 EBM)



kanayaさんEBMへ登録ありがとうございます。
2日前の履歴です。他の日の閲覧したURLを登録したい場合は
こちらよりお選びください。
今日 昨日 3日前 4日前 5日前

その他の場合はご記入ください

登録の際には下記より登録したいURL上のボタンをクリックしてから
上記の登録ボタンをおしてください。

閲覧回数
4 回 ○
http://page2.auctions.yahoo.co.jp/jp/auction/b4678830

3 回 ○
http://edit.my.yahoo.co.jp/config/validate user?

3 回 ○
http://edit.my.yahoo.co.jp/config/login?

2 回 ○
http://www.yahoo.co.jp/

2 回 ○
http://search.auctions.yahoo.co.jp/search/auc?

2 回 ○
http://edit.my.yahoo.co.jp/config/login

2 回 ○
http://auctions.yahoo.co.jp/jp?

2 回 ○
http://auctions.yahoo.co.jp/jp/?

2 回 ○
http://auctions.yahoo.co.jp/

1 回 ○
http://page2.auctions.yahoo.co.jp/now?

1 回 ○
http://page2.auctions.yahoo.co.jp/jp/show/tos

図 9　EBM の登録画面 (旧 EBM)

4.3　EBMng

欲しい情報があれば欲しい情報のキーワードを登録する。登録することによって毎日のアクセスログからキーワードを含むURLをEBMに表示するようにする。登録した情報は蓄積させ続ける。この機能を付加したものをEBMngとする。

この機能を付加することによって研究室内のユーザが興味を持ったキーワードを知ることと、調べたいキーワードを登録するだけで自動で情報を収集することが可能になる。

5 まとめ

ABM に URL の検索機能を足したことによって欲しいキーワードを含む URL を抽出することが可能になった。この機能で複数ある URL の中から自分にとって必要な情報を素早く取り出せる可能性が出来た。しかし、URL に欲しいキーワードが含まれていない可能性もあるので欲しい情報をもっと確実に得られるような機能を付け加えたい。現状では指定した複数の日にちを一つの履歴として表示出来ないので表示できるようにしたい。表示出来るようになれば見たい履歴の日にちがはっきりとわからなくても調べ易くなると考えられる。

EBMng を開発したことによって調べ物をするときに重要なキーワードを登録しておくだけで情報を蓄積させることができるようになった。

謝辞

本研究を進めていく上で様々な御協力及び、御指導頂いた藤井 信夫教授、大垣斉講師、情報安全工学研究室のメンバーに深く感謝の意を表します。

参考文献

[1] Perl/CGI 辞典 [昭和システム]

[2] http://www.kent-web.com/perl/index.html[Perl 基礎入門]

付録 A　各プログラム

A.1　アクセスログからユーザのABMを作るプログラム

```
#!/usr/bin/perl
#入力した文字列を抽出してcountするプログラム

open (FH,"zcat /misc/log/squid/access.log.0.gz|");
@line = <FH>;
close (FH);

$user_name = " test "; #testにユーザ名を代入
$i = 0;
$j = 0;
$k = 0;
$l = 0;
$n = 0;
$z = 0;
chomp($user_name);

foreach $line (@line){
        if ($line =~ /$user_name/){ #user_nameに代入した文字列を抽出
    $line =~ s/.+http:\/\//http:\/\//g;#http以前の文字列を削除
    $line =~ s/\s.+//g; #http://----の後にくる1つ目の空白以降を削除

    if($line =~/\/$|html$|htm$|cgi$|aps$|txt$/){#httpの末尾に特定の文字列があるものを抽出
$str[$i] = $line; #上の4つの作業をしたものを$strに代入
$i++;
    }
}
}

@tyusyutu = sort { $b cmp $a } @str; #@ strの内容をソートして@tyusyutuに代入

while($tyusyutu[$k]){
  TO:
    if ($tyusyutu[$j] eq $tyusyutu[$k]){ #配列tyusyutuの中で同じ文字列のものを探し重複しない
ようにする。重複があれば重複の数を数える。
        $k++;
        $suuti++;
        goto TO;
    }
    $count[$l] = $suuti; #重複した文字列の個数を代入していく。
    $last[$l] = "$count[$l] $tyusyutu[$j]"; #個数と文字列を表示。
    $j=$k;
    $l++;
    $suuti=0;
}

@kekka = sort { $b <=> $a } @last; #@count の数値が大きいものから順にソート

foreach $kekka (@kekka){
    @kekka2[$z] = $kekka;
    $kekka =~ s/.*http:\/\//http:\/\//g; #http 以降を抽出。
    @kekka2[$z] =~ s/http.*//g; #http 以降を削除。
    @kekka3[$z] = $kekka;
    $z++;
}

#http 以降の URL 部分だけを保存
open(FH2,">/home/2006/nakamae/public_html/sort/ptoday/test_URL");
print FH2 @kekka3;
close (FH2);

#http 以前の数値部分を保存
open(FH3,">/home/2006/nakamae/public_html/sort/ptoday/test_count");
print FH3 @kekka2;
close (FH3);

exit;
```

A.2 アクセスログから user 名を抽出

```
#!/usr/bin/perl
#user 名を抽出し出力
#すでにある name と現在出力した name とを結合して同じ user 名を削除
#rename_ebm というファイルの一部分を置換

open (FH,"/home/2006/nakamae/EBM_touroku.cgi");
@query = <FH>;
close (FH);

$i=0;
$j=0;

while ($query[$i]){
    $query[$i] =~ s/<.*>|\s|\n//g;
    if ($query[$i] =~ /.+/){
$query2[$j] = "$query[$i] ";
$j++;
    }
    $i++;
}

@query2 = sort @query2;
open (FH3,"/home/2006/nakamae/bin/rename_ebm");
@rename = <FH3>;
@temp = @rename;
close (FH3);

$i=0;
open (FH2,">/home/2006/nakamae/bin/Perl/key");
print FH2 @query2;
close (FH2);

open (FH4,"/home/2006/nakamae/bin/Perl/key");
$name = <FH4>;
close (FH4);

print $query3;
$temp[1] =~ s/name=\(//g; #name=(を削除
    $temp[1] =~ s/\).*\n//g; #) 以降を削除
    while ($rename[$i]){
$rename[$i] =~ s/$temp[1]/$name/g; #@rename の中の$temp[2] の文字列を$name の文字列に置換
    $str[$i] = $rename[$i];
print $str[$i];
$i++;
    }

open (FH6,">/home/2006/nakamae/bin/rename_ebm");
print FH6 @str;
close (FH6);
exit;
```

A.3 EBM登録

```
#!/usr/bin/perl
#EBMで手動でURLとカテゴリを追加するプログラム。

require "./cgi-lib.pl";

if($ENV{’REQUEST_METHOD’} eq ’POST’){
                read(STDIN,$query, $ENV{’CONTENT_LENGTH’});
        }
        else{
                    $query = $ENV{’QUERY_STRING’};
            }

$query =˜ s/=//g;
$query =˜ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
$query =˜ s/\+//g;
$query =˜ s/\&/\n/g;
$query =˜ s/ジャンルa//g;

open (FH,">/tmp/nakamae/war");
print FH $query;
close (FH);

open (FH2,"/tmp/nakamae/war");
@jouken = <FH2>;
close (FH2);

open (FH3,"/tmp/nakamae/EBM_kategori");
@kategori = <FH3>;
close (FH3);

$a = 0;
$b = 0;
$c = 0;
$k = 0;
$i = 0;
$j = 0;

print "Content-type: text/html; charset=euc-jp\n\n";
print <<EOF;
<html>
```

```
<head>
</head>
<br>
EOF
    #ジャンルの後に1文字以上の文字列があるものを抽出
    if ($jouken[0] =~ /ジャンル[0-9]*/){
    print "キーワード<br>";
    $jouken[0] =~ s/ジャンル|\s//g;
    print "$kategori[$jouken[0]]<br>";
    $sinki = $kategori[$jouken[0]];
    $b = 1;
    $c = 1;
}
#上のif文が真であれば$bに0を代入し、その他の場合の処理を行う。
if ($b == 0){
    print "<hr>";

if ($jouken[1] =~ /その他.*/){
    $jouken[1] =~ s/その他|\s//g;
    $touroku = 0;
    while ($kategori[$k]){
if ($jouken[1] =~ /^$/){#空行であった場合処理を開始。
    $touroku = 1;
    print "文字を入力してください。";
    print "<hr>";
    last;
}
    if ($kategori[$k] eq "$jouken[1]\n"){#すでに登録されていた場合。
print "このカテゴリはすでに登録済です。<br>";
$touroku = 1;
$sinki = $jouken[1];
$c = 1;
last;
    }
$k++;
    }
    if ($touroku == 0){#今までに登録されていなかったカテゴリであれば登録する。
open (FH2,">>/tmp/nakamae/EBM_kategori");
print FH2 "$jouken[1]\n";
close (FH2);
$sinki = $jouken[1];
$c = 1;
print "新規カテゴリ『$jouken[1]』登録されました。<br>";
```

```
    }
}
}


if ($jouken[2] =~ /ユー.*http/){
    if ($c == 1){#カテゴリが入力されていれば実行。
$jouken[2] =~ s/ユー/$sinki/g;
$jouken[2] =~ s/\n/ /g;
open (FH2,"/tmp/nakamae/EBM");
@EBM_url = <FH2>;
close (FH2);
while ($EBM_url[$j]){
    $url = 0;
    if ($EBM_url[$j] eq "$jouken[2]\n"){
print "このURLはすでに登録済です。<br>";
$url = 1;
last;
    }
    $j++;
}
if ($url == 0){
    open (FH2,">>/tmp/nakamae/EBM");
    print FH2 "$jouken[2]\n";
    close (FH2);
    print "新規URL 『$jouken[2]』登録されました。<br>";
}
}else{
    print "カテゴリを登録してください。";
}
}
print "<br>";
print "</html>";
```