大量のブックマークに対する整理方法について

平成 18 年 2月 21日

情報電子工学科
宮沢 雄

目 次

1 はじめに **1**

2 ブックマーク機能について **1**

2.1 Web ブラウザについて 1

2.2 Netscape3.04 でのブックマークの管理方法 2

2.3 問題点 3

3 ブックマークの仕組み **4**

3.1 ブックマークの保存状態 4

3.2 bookmarks.html の基本的な構造 4

3.3 Netscape3.04 との違い 7

3.4 Netscape7.0 でのブックマーク 7

4 整理方法の提案と考察 **8**

4.1 一般的な整理方法 8

4.2 同じサイトでまとめる方法 8

4.3 検索結果でまとめる方法 9

4.4 使用頻度別に整理する方法 11

5 実行結果 **13**

5.1 検索結果でまとめた場合 14

5.2 使用頻度でまとめた場合 16

5.3 考察 18

6 まとめ **18**

参考文献 **20**

概要

WWW の利用に関しては、 Web ブラウザのブックマーク機能を使うことにより簡単に目的のサイトに行くことができる。ブックマークは使用頻度の高い Web サイト の URL を記録しておくことで、そのサイトへのアクセスに関して、 URL を入力する必要がなくなり便利である。しかし、長く使っていくとブックマークが大量に登録されてしまい、逆に使いづらくなってしまう。そこで、本研究ではこの大量のブックマークに対していくつかの整理方法を提案、考察した。また、実際にそれらの整理方法を実現するためのプログラムを作成して実験した結果、新たにでてきた問題を考察した。

1　はじめに

ブックマークとは、書籍の「しおり」に似た Web ブラウザの機能で、Web サイトのアドレス (URL) を記録しておきそのサイトへのアクセスを簡潔にできるというものである。ブックマークは Web ブラウザ上でも多少の整理はできるが、基本的には自分でフォルダを作りその中へリンクを移動させていくという作業の繰り返しとなり、大量のブックマークを手作業で整理することは効率が悪いため、少しでもその作業を減らすことを考えるとソフトウェアに頼ることになる。

しかし、現在ブックマークを管理するソフトウェアは MS-Windows 上ではいくつもあるが UNIX では普及しておらず、手作業で整理せざるを得ない。そこで本研究では、この研究室の中心的なブラウザ環境である Netscape Navigator を元にブックマークの管理方法を考え、既存のブックマーク構造を読み取り、手作業を減らせるような形に変換するプログラムを作ることを目指す。

2　ブックマーク機能について

現在使われている Web ブラウザのブックマーク機能について、管理方法や仕組みを調べ問題点を挙げた。

2.1　Web ブラウザについて

Web ブラウザとは、Web 上で Web コンテンツを回覧するためのソフトウェアのことで、代表的なものとしてネットスケープ社の Netscape Navigator 、マイクロソフト社の Internet Explorer 等がある。

Netscape Navigator

Netscape Communications 社の Web ブラウザ。電子メールソフトや NetNews クライアントなども標準で装備している。ベータ版をインターネットを通じて無料で配布したことから一時は 9 割近いシェアを獲得していたようだが、 Microsoft 社の Internet Explorer により、近年のシェアは低迷している。1998 年 1 月に製品版の無償配布とソースコードの公開が行われ、オープンソースソフトウェアとして世界中のボランティアプログラマの手を借りて開発を行なっている。

Internet Explorer

Microsoft 社の Web ブラウザ。 Netscape Navigator と市場を 2 分していたようだが、現在では Internet Explorer がシェアのほとんどを握る。電子メールソフトの Outlook Express などが Internet Explorer とあわせて配布されており、インターネットや雑誌の付録 CD-ROM を介して無料配布されている。

2.2 Netscape3.04 でのブックマークの管理方法

次に記す図は、 Netscape3.04 でのブックマーク管理ウィンドウを図化したものである。

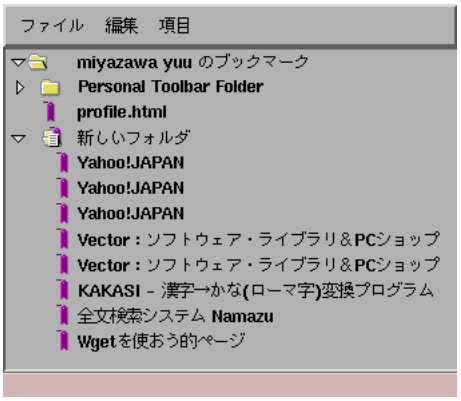

Fig. 1 Netscape3.04 のブックマーク管理ウィンドウ

ファイル	編集	項目
開く	元に戻す	プロパティ
読み込み	やり直し	ブックマークへジャンプ
名前をつけて保存	切り取り	ブックマークの並べ変え
最新情報	コピー	ブックマークの挿入
閉じる	貼り付け	フォルダを挿入
	削除	区切り線を挿入
	全てを選択	エイリアスの作成
	検索	新規ブックマークフォルダにする
	次を検索	ブックマークメニューフォルダにする

Table 1 Netscape3.04 のブックマーク管理ウィンドウのメニュー

Fig 1 にある重複したリンクはエイリアスによるもので、どれを選んでも同じ場所に行くことになる。エイリアスとは代理を意味する言葉で、MS-Windows でいうとショート

カット。コピーににているが実体をもたず、元のファイルを編集するとエイリアスも変更され、また、エイリアスを編集すると本体も変更される。エイリアスを消しても本体に影響はないが、本体を消すとエイリアスも消える。

Table 1 は Netscape3.04 での管理メニューではあるが、基本的な操作はどの Web ブラウザのブックマーク管理方法にも共通している。

ブックマークから URL を検索する場合、以下の性質がある。

- 探す条件として「名前」、「ロケーション」、「説明」がある。
- 大文字と小文字を区別できる。
- 完全に一致する単語だけを検索できる。
- 検索を続ける時には [次を検索] をつかう。

名前・ロケーション・説明・大文字と小文字の区別・完全に一致する、という五つの条件は同時に複数使うことができる。

ロケーションとは場所、サイトの URL のこと。インターネット上に存在する情報資源 (文書や画像など) の場所を指し示す記述方式で、インターネットにおける情報の「住所」にあたる。

「新規ブックマークフォルダにする」とはフォルダのみ選択可能で、次回のブックマーク登録時にその指定したフォルダにブックマークを入れる。

「ブックマークメニューフォルダにする」もフォルダのみ選択可能で、ブラウザ内のブックマークメニューをクリックした際に表示されるプルダウンメニューにそのフォルダ内のブックマークが表示される。

2.3 問題点

Web ブラウザでブックマークを整理するにあたり、問題点がはっきりした。

- ほとんどが手作業であること。

  自動でできる事といえば、更新チェック、並べ替えくらいである。整理するためには、新しくフォルダを作る、ブックマークを移動させる、などという手作業による操作が不可欠である。

- 全体が見通せないこと。

  階層が増えていく度に毎回そのフォルダを開きながら見ていくことになり、リンクが大量にあると表示が縦に長くなってしまいスクロールさせながら見ていくことになる。リンクの量が多くなるほど目的のリンクにたどり着くまでの操作が多くなり、時間がかかってしまう。

このことから、 Web ブラウザによるブックマークの管理は整理、使い易さという面で、十分とはいえないことがわかる。これらの問題点を改善できるような整理方法を考察していく必要がある。

3　ブックマークの仕組み

整理方法を考えていく上でブックマークの仕組みを理解する必要があると考え、調べた。

3.1　ブックマークの保存状態

ブックマークは一つにまとめられ、「bookmarks.html」という名前の HTML 形式で自動で生成され保存されるため Web ブラウザで見ることもできる。また、 bookmarks.html にはフォルダとリンクの他に作成日時、最終訪問日時、最終変更日時、エイリアス等の情報が加えられる。

ブックマークを登録する際、デフォルトではそのページの HTML ソース内の<title>から</title>の間の文章が名前として使われる。

3.2　bookmarks.html の基本的な構造

bookmarks.html 内の最低限のタグを取り出し、次に記す。

```
<H1>ブックマーク</H1>

<DL><p>
    <DT><H3>フォルダ 1</H3>
    <DL><p>
        <DT><A HREF=" URL ">リンク 1-1</A>
<DD>説明
```

```
    </DL><p>
    <DT><A HREF=" URL ">リンク</A>
<DD>説明
    <DT><H3>フォルダ 2</H3>
    <DL><p>
        <DT><A HREF=" URL ">リンク 2-1</A>
<DD>説明
    </DL><p>
</DL><p>
```

ここでの行のとり方、インデントはブックマークの機能とは関係がないため、決まり事であると考えられる。

インデントの取り方

インデントは半角スペース 4 個を基本としており、階層が一つ下るごとに 4 つづつ足されていくが、 <DD> にインデントは無い。

bookmarks.html のつくり

bookmarks.html のつくりを Fig.2 に記す。

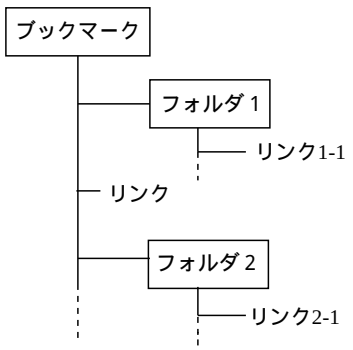

Fig. 2 ブックマークのつくり

ブラウザで見た場合は、ブックマーク管理ウィンドウとは違いフォルダの下のリンク、説明を全て表示しているが、作成日時、最終訪問日時などは表示されない。字下げにより階層を表しており、この状態でもリンクリストとして使うことができる。

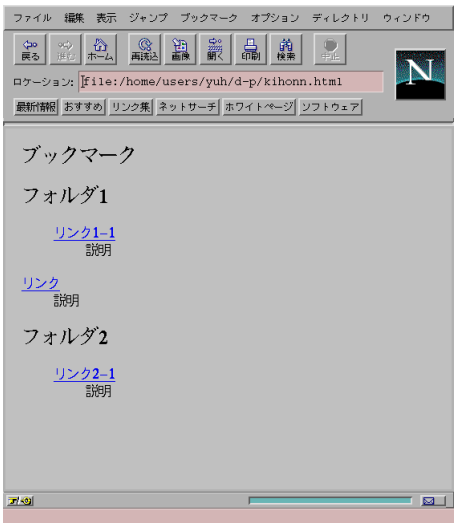

Fig. 3 ブラウザで見た場合

タグの働き

ブックマーク機能に関係のあるタグを紹介する。

<H○ >フォルダ名</H○ >

これはフォルダを示しており、 bookmarks.html 内では 1 と 3 が入る。1 は、そのブックマークの最上階層となるフォルダに入れられ、html 内のフォルダの記述でも最初に書かれる。3 が入る場合は <H3 FOLDED ADD_DATE="xxx">フォルダ名</H3> となる。

<DL>,</DL>

<DL> はフォルダのすぐ後に書かれ、これがつくと階層が一つ下がる。 </DL> がつくと階層が一つ上がり、無い場合は前にあるフォルダの下に続くことになる。つまり <DL> から </DL> までの間がフォルダの中身となる。

<A HREF=" URL ">リンク名</A>

これはリンクを示している。こう記述することによりリンク名が表示され、そこをクリックするとそのページへと移動できる。また、このタグの中に追加日時などの情報が入る。

<DD>

これは説明を示す。タグを閉じる必要がなく、この後ろに入力した説明が並ぶことになる。

3.3 Netscape3.04 との違い

同じ Web ブラウザでもバージョンが違えば、ブックマークにおいて機能の追加、変更されている場合があるため、Netscape3.04 との違いを調べた。

- Netscape4.08

  場所、最終訪問日時、作成日時の表示と、それによる並べ替えができる。Netscape3.04 でもそれらの情報を持っているが、Fig1 をみてわかるように表示と並べ変えのためには使えない。使えるのは Netscape4.08 以降である。

- Ntscape7.0

  キーワード、説明による並べ替えができる。Netscape7.0 では説明の他にキーワードを入力できる。特に違いは無いが、説明は <DD> に入るのに対し、キーワードは `<A HERF=" URL ">` の中に入る。

3.4 Netscape7.0 でのブックマーク

Netscape7.0 はエイリアス機能が無いが、エイリアスを含んだ HTML は表示でき、使うことができる。しかし、ブックマーク管理ウィンドウでエイリアスを含んだブックマークを見てみると Fig.4 のようになる。

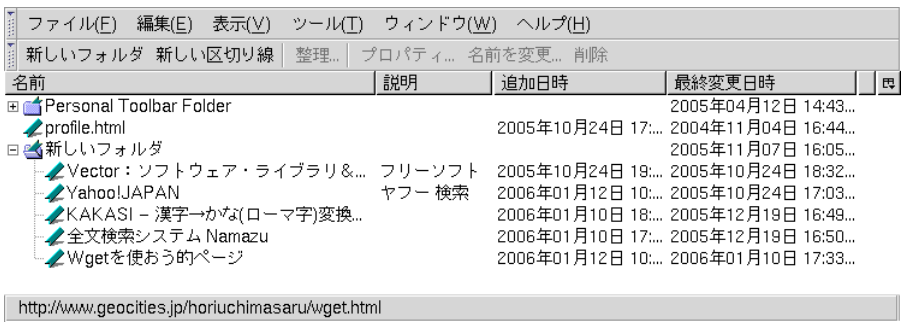

Fig. 4 Netscape7.0 のブックマーク管理ウィンドウ

Fig.1 と同じ bookmarks.html を表示しているがエイリアス機能がないため表示されない。

4　整理方法の提案と考察

リンクが大量にある場合の整理方法について考察した。

4.1　一般的な整理方法

一般的な整理方法として、Web ブラウザでのブックマークの管理機能を使い

1. リンク切れを探し削除。
2. 同じジャンルでまとめ、フォルダを作る。
3. 使用頻度が高いであろうリンクを階層の上の方に持ってくる。

ということが挙げられる。

これらの方法のうち 1 は削除はされないが更新をチェックする機能がある。しかし、 2 、 3 は先に述べたように効率が悪い。そこで、少しでも作業を減らせるような整理方法を考えた。

4.2　同じサイトでまとめる方法

bookmarks.html 内のリンクの URL を参照していき、同じ Web サイトの違うページがあった場合、一つにまとめるという方法である。

例として、

```
http://takeno.iee.niit.ac.jp/~foo/
http://takeno.iee.niit.ac.jp/~foo/#news
http://takeno.iee.niit.ac.jp/%7Efoo/Lab/Lab.html
```

という URL は

「http://takeno.iee.niit.ac.jp/」

という同じ Web サイト内にあるページのリンクである。このようなリンクがあった場合に有効ではないかと考えた。この Web サイトのトップページのタイトルをフォルダ名とし、その中にリンクを入れる。という方法である。

このようなまとめ方をした場合、例のように複数個リンクがあれば、まとめることによりすっきりとしたブックマークになると考えられる。

考察

この整理方法は、同じ Web サイトのリンクが複数あることを前提としているが、そのようなリンクが少ない場合にはリンクが 1、2 個しか入っていないフォルダが大量に作られることになる。

また、リンクは同じ Web サイト内にあってもジャンルが異なるページの可能性があり、このようなリンクが同じフォルダに入っている場合には、目的のリンクを探すのが余計困難になってしまうと思われる。

このことから、この整理方法では整理したことにはならないといえる。

4.3 検索結果でまとめる方法

フォルダ名、リンク名、 URL 、説明を対象として、あるキーワードで検索した結果を表示する。表示の例として、

- [フォルダ]–...–[キーワードを含むフォルダ]
- [フォルダ]–...–[フォルダ]–(キーワードを含むリンク)

となる。また、そのキーワードをフォルダ名としてフォルダを作り、その中にキーワードを含んだリンクをいれる。という方法である。

考察

自分でキーワードをいれるため、自分にとって理解しやすいファイル名になるといえる。また、この整理がジャンル別に整理したことにもなり、使いやすいブックマークになると思われる。しかし、次のような場合、検索しても目的のリンクが見つからない可能性がある。

- フォルダ名、リンク名がページの内容と異なっている。
- 「説明」が入っていない。

この場合、検索をする際に特に重要であると思われるのが 「説明」 であるが、ブックマークを登録する際に 「説明」 をいれることが少なく既存のリンクの中に検索で使用するキーワードが入っていないことが多い。

そのため、説明を自分で入れるのではなく、自動的にもってくることができればよいと考えた。そのために以下のものを利用する。

- w3m
- KAKASI
- Namazu

これらを利用すれば、その Web サイトに含まれる文章を「説明」とすることで問題点を解決できる。

w3m

フリーソフトウェアで、テキストベースの Web ブラウザである。UNIX 環境では, Web ブラウザとして Netscape が定番となっているが w3m を使うと kterm 上で Web を閲覧することができる。また、URL ファイルをファイルとして保存することもできる。本研究では 「-dump_source」 というオプション使用し、

```
w3m -dump_source [URL] > file.html
```

とすることで指定した URL を持つ Web サイトの HTML ソースを HTML ファイルとして保存する。

KAKASI

フリーソフトウェアで、漢字かなまじり文をひらがな文やローマ字文に変換することを目的として作成されたプログラムと辞書の総称である。日本語の単語のわかち書きにも使えるため、 Namazu でわかち書き処理にも用いられている。

Namazu

フリーソフトウェアで、全文検索システムである。インデックスの作成を行う mknmz コマンドと、検索を行う namazu コマンドから構成されている。

mknmz でどの単語がどの文書で使われているかを記録した索引ファイル (インデックス) を作成し、namazu で作成されたインデックスに対して検索を行い、利用者が入力したキーワードと、インデックスの単語を比べて、その単語が含まれている文書ファイルを表示することができる。ここで、 mknmz のオプションを

```
mknmz --html-split -O [index] [dir]
```

とすることにより `--html-split` で HTML ファイルを`<a name="...">` 単位で分割処理し、`-O` でインデックスの出力先を指定する。

また、 namazu は

```
namazu -h [検索文字列] [参照インデックス]  > file.html
```

と使用し、 `-h` とすることで検索結果を HTML で出力するため、HTML ファイルとして保存する。

検索結果でまとめるために、これらをどのように使うかを Fig.5 で記す。

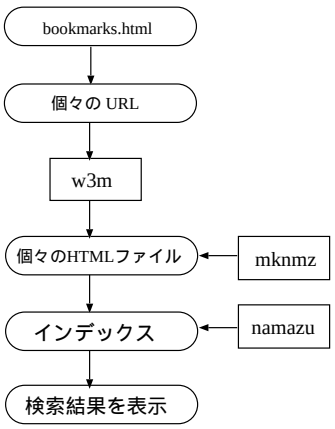

Fig. 5 検索結果でまとめるために

bookmarks.html から個々の URL を取り出し、 w3m でそれぞれの HTML ファイルとして保存する。そして、その HTML ファイルを元に mknmz を使いインデックスを作成すれば検索できるようになり、その結果を Web ブラウザにまとめて表示できるようになると考えた。

4.4 使用頻度別に整理する方法

これはブックマーク管理ウィンドウを見ながら、使用頻度が高いリンクを取り出していく方法である。リンクの動きを Fig.6 に記す。

1. まず、「1. 未処理」、「2. 使用中」、「3. 保留 」、「9. 保存」というフォルダを作る。フォルダ名の先頭にくる数字は、並べ換えをしても順番が変わらないようにするためである。

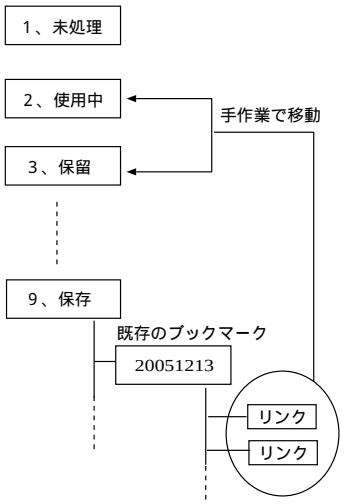

Fig. 6 使用頻度別に整理する

2. 「9. 保存 に適当な名前のフォルダを作り、全てのブックマークをそこへ移動させる。ここでは 「20051213」 とする。
3. 「20051213 から最も使用頻度が高いと思われるリンクを 「2. 使用中」の中に移動させる。
4. 「20051213」から次に使用頻度が高いと思われるリンクを 「3. 保留」の中に移動させる。
5. この状態でしばらく使い、状況に応じてフォルダを作り足していく。また、「4.XXX」、「5.XXX」 ... と作っていくこともできる。
6. 残りの「20051213」の中にあるリンクは使う機会が少ないことになる。「2001」や「2002」など年ごとのフォルダを作り、そこへ移動させていく等、整理していく。

以後、新しいブックマークを登録する時は「1. 未処理」に入れ後で整理できる。という方法である。

考察

ある程度使用頻度を決めることにより、以後の作業を少しでも減らすことができると考えられる。また、よく使うリンクだけを持ってくることで、使いやすいブックマークになると思われる。しかし、次のような問題点がある。

- 全て手作業である。

- 大量のリンクを移動させる可能性がある。

この方法では全体を見ながら作業をしなければならず、一つづつ移動させていくため効率が悪いといえる。特に手順 3、4 において、使用頻度が低くなってきた場合「3. 保留」や「9. 保存」へリンクの移動の必要がでてくる。

その解決策として、それぞれのリンクは最終訪問日時、最終変更日時などの情報を持っているため、それらの情報を活かし使用頻度に反映させていき、自動的にリンクを移動させることができれば良いと考えた。自動的に移動させる場合のリンクの動きを Fig.7 に記す。

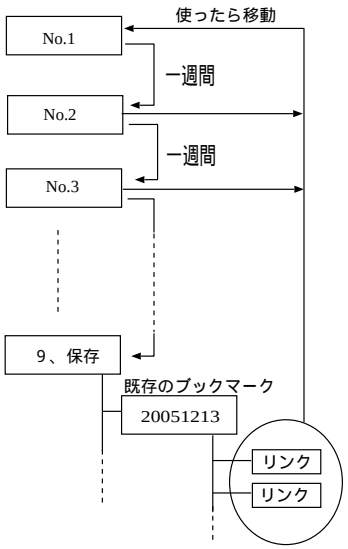

Fig. 7 リンクを自動的に移動させる

使用頻度が高いリンクを自動的に移動させるためには

- 使ったらとりあえず一番上のフォルダにいれる。
- 一週間使わなかったら一つ下のフォルダへ移動させる。

の繰り返しでできると考えた。

5 実行結果

これまでに提案、考察してきた整理方法を実際に試すことでどのようなブックマークになるのかを調べた。また、プログラムを作成するにあたり、 Perl、csh を用いた。その理

由として Perl は、テキスト処理、ファイル操作が容易である。csh は既存のコマンドを組合せて処理できるためである。

使用したブックマークについて

今回使用したブックマークはリンクが数 2371 個あり、リンク切れを含むものである。また、普段使うブックマークとは別に bookmarks.html を作ったため、情報が古いといえる。

5.1 検索結果でまとめた場合

プログラムを作成するために考えをまとめたものを Fig.8 に記す。

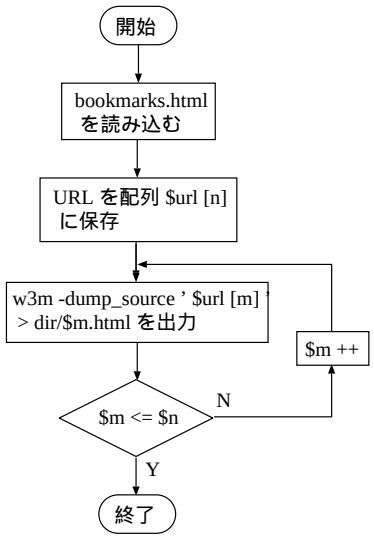

Fig. 8 検索結果でまとめるプログラムのフローチャート

これは、 bookmarks.html から個々の URL を取り出し配列 url[n] に保存し、

```
w3m -dump_source url[m] > dir/$m.html
```

を出力して終了となる。実際は出力を csh ファイルとして保存することにより、w3m でれぞれを HTML ファイルとして書き出すことを目的としたものである。これにより、指定したディレクトリに 1 番から m 番の HTML ファイルを作ることができ、後は Namazu を使うだけになる。

しかし、 w3m は指定された URL に接続していくため、 HTML ファイルの作成時に相当な時間がかかってしまう。もし作業を中止した場合、また最初から始めなければならない。そこで指定したディレクトリに何番まで保存されているかを調べ同じ HTML ファイルが作成されるのを防ぐようにした。

実行結果

作成したインデックスを元に namazu で「UNIX」をキーワードとして検索し、 Web ブラウザで表示させたものを Fig.9 に記す。

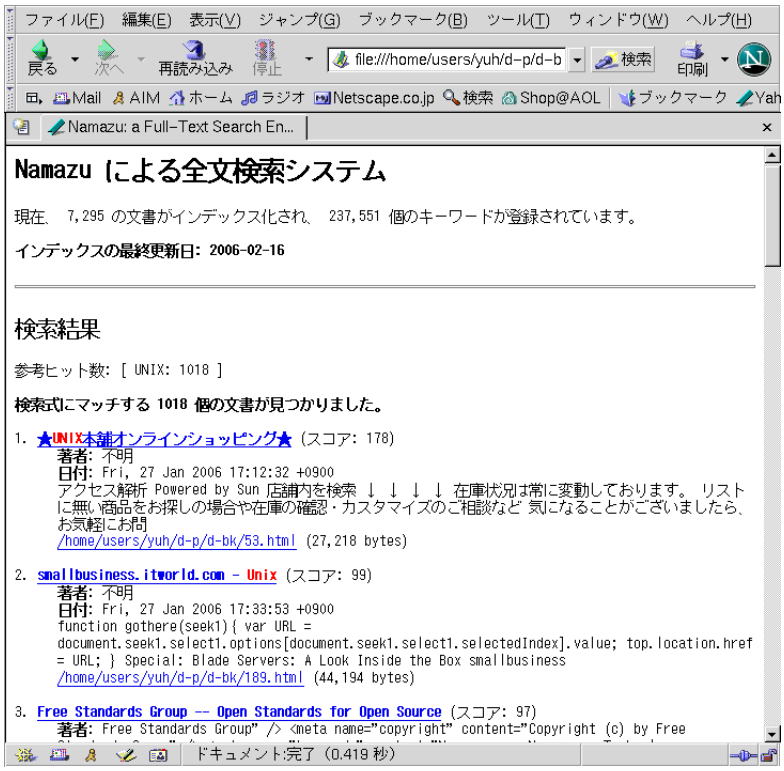

Fig. 9 Web ブラウザで表示

これはそれぞれのリンクの説明を表示していることになるが、全てのリンクに説明が入っているわけではないため不十分である。それは、リンク先の Web サイトの移転や、ソースが圧縮されており対応できない Web サイト、などにより説明として使えない内容となっていたり、ソースを保存できない Web サイトのリンクがあるため、空のファイルを作ることになる。そのファイルを参照してインデックスを作成したため、全てのリンクに説明が入っていないことになる。

5.2 使用頻度でまとめた場合

プログラムを作成するために考えをまとめたものを Fig.10 に記す。

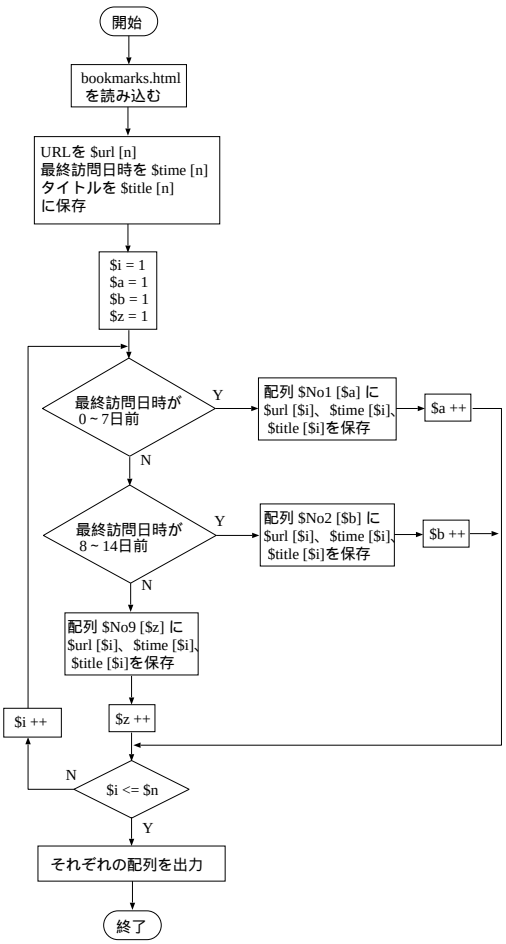

Fig. 10 使用頻度でまとめるプログラムのフローチャート

これは URL の他に 最終訪問日時とタイトルも配列に保存し、最終訪問日時の値を見てどこのフォルダに入れるかを判断し、それぞれの配列へ移動させ、出力して終了となる。この出力は HTML ファイルとし、 Web ブラウザ上で使用できるブックマークとなる。また、出力する文に手を加えることにより、ブックマーク管理ウィンドウでの表示ができる。

実行結果

実際に Web ブラウザ上で見た場合は Fig.11 、ブックマーク管理ウィンドウでの表示

すると Fig.12 となる。

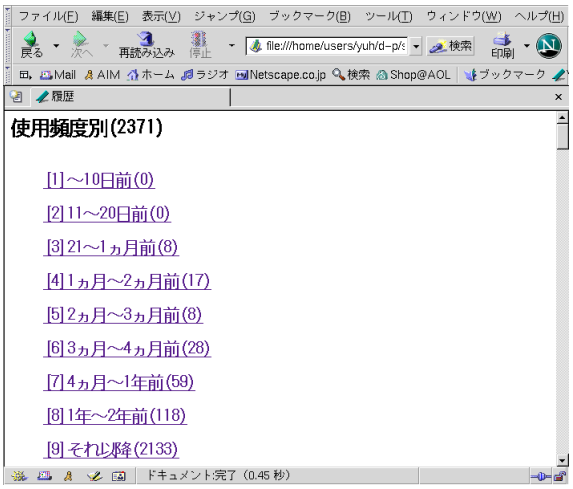

Fig. 11 Web ブラウザで表示

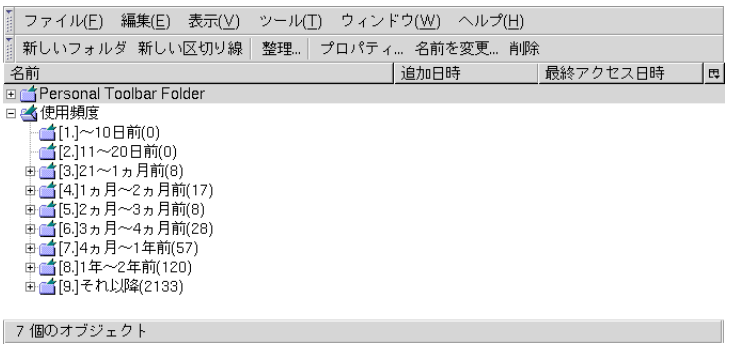

Fig. 12 ブックマーク管理ウィンドウで表示

ここでは一週間ではなく適当にフォルダを区切り、リンクとして表示させ、このリンクをクリックすることでフォルダの中にあるリンク一覧が表示される。そのリンク一覧は同じページに出力しているため同ページ内リンクとなるが、「それ以降」には大量のリンクがあるため違うページとなっており、新しいウィンドウが開くようになっている。また、括弧内の数字はその中にあるリンクの総数となる。

これで提案した整理方法をある程度実現できたといえ、最終訪問日時が更新されたリンクが一番上に移動し、されないリンクが下がっていくことになる。

5.3 考察

検索結果でまとめる方法

リンク先の Web サイトの移転や、対応できない Web サイトによる空ファイル、保存した HTML ファイルを説明としてどのように既存のブックマークに反映させるかという問題が残っており、完成にはほど遠いといえる。また、今回は w3m を用いて HTML ファイルを作成したが、wget というソフトの方がページ情報を持ってくることに関して高機能なため、それについて調べ、使用した結果も必要だと考えられる。

wget フリーソフトウェアで、ダウンロード支援ツールであり、ホームページの自動巡回ツールでもある。 URL を指定すると、その指定したページ (ファイル) をそのまま自分のハードディスク内にダウンロードでき、ページの文章のみでもダウンロードすることができる。

使用頻度でまとめる方法

既存のブックマークのフォルダ構造を無視して作成したため、使用頻度が低いリンクを一つのフォルダにまとめてしまっている。それによりこのプログラム単体では使いづらいといえるが、既存のブックマークの補佐的な役割としては十分に使えると思われる。そこで、この整理方法は有効なのか、 Web ブラウザとブックマーク管理ウィンドウでの使いやすさに違いがあるのか、など実際に使用してもらい、アンケートを取ることが必要だと思われる。

6 まとめ

本研究では、ブックマークが大量にある場合の整理方法を提案、考察し、実際にプログラムを作成した。まず、ブックマークの機能を調べた結果、問題点がいくつかあり、ブックマークが大量にある場合は Web ブラウザによるブックマークの管理は難しく、使いづらいことがわかった。

そのことから、わかった問題点を改善できるような整理方法を考え、基礎研究としてブックマークの仕組みを理解することで新たな整理方法を三つ提案、考察することができた。

その提案した整理方法のうち二つを実際に形にすることができ、他の問題点を見つけることができたが、その問題を解決できるまでに至っていない。また、使いやすさなどの利用者の視点から見た考察も必要だといえる。

今回はテストプログラムを作っただけで終わってしまい、提案した整理方法を実現できるようなものが完成していないため、その開発が今後の課題となる。また、さらに新たな整理方法や表示方法を提案していく必要があると思われる。

参考文献

[1] 岡本茂, 大島邦夫, 堀本勝久： 2001-'02 パソコン用語辞典 (技術評論社,2001)

[2] 深沢千尋： すぐわかる Perl (技術評論社,2004)

[3] 山森丈範： シェルスクリプト基本リファレンス (技術評論社, 2005)

[4] 馬場肇： Namazu システムの構築と活用 (ソフトバンク, 2001)

[5] Namazu Project： 全文検索システム Namazu
http://www.namazu.org/index.html.ja

[6] KAKASI project： KAKASI - 漢字→かな (ローマ字) 変換プログラム
http://kakasi.namazu.org/index.html.ja

[7] 伊藤彰則： W3M Homepage
http://w3m.sourceforge.net

[8] HORIUCHI Masaru： Wget を使おう的ページ
http://www.geocities.jp/horiuchimasaru/wget.html