

# WWW ナビゲーションのためのグラフィカルブラウザの開発

島村 栄 久保 信也 高野 元  
NEC C&C 研究所

本稿では、WWW ナビゲーションを支援するツールであるマップブラウザについて述べる。マップブラウザは WWW ブラウザと連動し、その時点で WWW ブラウザでアクセスしているドキュメント周辺のノードリンク構造を表示する。画面上では、アイコンと線でノードとリンクを表現し、それぞれのアイコンの形でドキュメントのメディアの種類を示す。また、マップブラウザはそれぞれの WWW ドキュメントに対するアクセス頻度を計算し、ノードの情報の一部として格納する。アイコンの影の広さでアクセスの頻度を示す。表示するノードリンク構造はナビゲーションの際に取得した HTML ファイルを解析することで生成される。試用した結果、WWW ブラウザのみの場合よりも、マップブラウザを用いてナビゲーションを行なう方がより容易に情報を発見できることがわかった。

## A Graphical Browser for WWW Navigation

Hisashi Shimamura Nobuya Kubo Hajime Takano  
C&C Research Laboratories, NEC

This report describes a tool for supporting WWW navigation, which we call MapBrowser. MapBrowser works with WWW browser, and displays a node-link structure near the current document shown in the WWW browser. Icons and arcs in the screen means nodes and links, and each icon image is chosen according to its content type. Mapbrowser also counts an access frequency for each WWW document and stores it as one of node information. Width of the icon edge indicates value of access frequency. Besides, node-link structure displayed in the screen is generated by extracting each of HTML files opened through the navigation process. In experimental use, we proved that WWW navigation with Mapbrowser was easier to find information than only with WWW browser.

## 1 はじめに

この2～3年の間に World Wide Web(WWW)によって提供される情報は著しく増加している。このため、ユーザが必要な情報へ効率よくアクセスすることが非常に困難になってきている。

必要な情報の所在を探す一つの手段として、Yahoo! や AltaVista, NETPLAZA, TITAN, 千里眼といった WWW 上でのディレクトリサービスが用いられる。WWW ディレクトリサービスは、WWW ドキュメントを内容に基づいて分類したり、キーワード検索機能を提供することによってユーザの情報発見を支援することができる。

しかし、これらのサービスを利用しても、Netscape や Mosaic, Internet Explorer 等の WWW ブラウザでは、その時点で閲覧している WWW ドキュメントから直接リンクされている WWW ドキュメントしか把握できず、その先に関係付けられている WWW ドキュメントを知ることはできない。このため、必要な WWW ドキュメントが数段リンクをたどった先にあることがわかっている場合でも、リンク先の見通しが悪いためリンクをしらみつぶしに探すことになり非効率的であった。

そこで我々は、ユーザのナビゲーション操作を支援する一手法として WWW ドキュメント間のノードリンク構造を視覚化するツールを用意することを考え、マップブラウザを開発した。マップブラウザは、通常の WWW ブラウザ (Netscape) と連携し、ユーザが現在アクセスしている WWW ドキュメントを中心としたノードリンク構造を表示する機能を持つ。

ここで、ノードリンク構造情報の取得方法が問題になるが、WWW ドキュメントからリンク先を自動抽出してログファイルへ蓄積していく方法と、ディレクトリサーバから表示に必要な範囲のノードリンク構造情報を表示の都度取得する方法の2種類を実装した。

以下では、2章で現状の問題とその解決法について述べ、3章で我々が開発したマップブラウザについて述べる。また、ディレクトリサーバの利用については4章で説明する。5章では評価を行ない、6章でまとめる。

## 2 WWW ナビゲーションの問題と解決策

WWW は、情報の単位となる WWW ドキュメントがリンクによって相互に関連づけられている一種のハイパーメディアシステムとしてとらえられる。ユーザは WWW ブラウザを用いて、ネットワーク (HTTP: Hyper Text Transfer Protocol) を介して取得した HTML (Hypertext Markup Language) によって書かれた WWW ドキュメントを閲覧する (図 1 参照)。

従来 WWW ブラウザによるナビゲーションの問題点と本稿での解決策は以下の通りである。

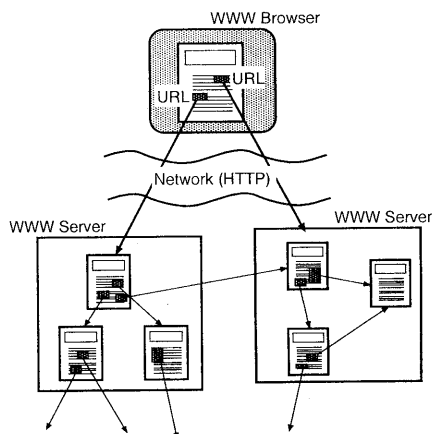


図 1: WWW ドキュメントのハイパーメディア構造

### リンクの見通しが悪い:

WWW ブラウザでは、表示中の WWW ドキュメントから直接リンクされた先の WWW ドキュメントしかわからない。そのため、何段か先の WWW ドキュメントを探す場合、全てのリンク先をしらみつぶしに調べなければならなくなる。

そこで、現在表示している WWW ドキュメントからリンク数段分の範囲のノードリンク構造を表示することにより、周囲の WWW ドキュメントを一覧可能とすることで、リンク先の見通しを良くすることができると思われる。

### ナビゲーション時の手がかりが貧弱:

ナビゲーション時の手がかりとして、WWW ブラウザでは以前にアクセスしたリンク先のアンカー文字列の表示色を変化させている。しかし、全てのリンク先にしらみつぶしにアクセスしたり、リンク先へアクセスしたが実際には必要がなかった場合が考えられるため、一度アクセスしたからといって、それが必要な情報であるとは限らない。

これに対しては、それぞれの WWW ドキュメントへのアクセスの頻度を記録することで、その WWW ドキュメントが自分にとってどの程度有用であるかの指標とすることができると考えられる。

### 3 マップブラウザの開発

この章では2章で挙げた解決策に基づいて開発したマップブラウザについて説明する。

#### 3.1 機能概要

マップブラウザはMicrosoft Visual C++2.0で実装され、Windows95/NT上でNetscape Navigatorと連携して動作する。常にNetscape上で取得したドキュメントを中心としたノードリンク構造を表示し、マップブラウザ上でのマウスクリックに応じてNetscapeは対応するWWWドキュメントを取得する(図3参照)。

このマップブラウザはナビゲーション支援のために以下のような機能を持つ。

##### ノードリンク構造の表示

Netscapeで表示しているWWWドキュメントを中心とした、限定した範囲のノードリンク構造のみを表示する。また、表示するノードリンク構造情報は、ユーザが取得したWWWドキュメントから抽出し、アクセス履歴として蓄積しておく。

##### アクセス頻度の記録方式、表示の方法

マップブラウザは、ユーザのWWWドキュメントへのアクセスを監視して、そのアクセス頻度をノードリンク構造と組み合わせてアクセス履歴として記録し、ノードリンク構造上に表示する。

#### 3.2 ノードリンク構造の表示

##### 3.2.1 表示形式

マップブラウザでは、Netscapeで表示されているWWWドキュメントを中心ノードとして、中心から2段階までの範囲でリンクされているノードを表示する。このように表示範囲を限定することによって、過度に複雑なノードリンク構造の表示を避けることができる。

表示形式は文献[3]を参考に、同一階層のノードを同心円状に並べる円形階層グラフによる表示を採用した。この方式によって第1に、中心となるノードからのノードリンク構造的に一定の距離にあるノードがわかりやすくなる。第2に、WWWドキュメント間の関係は、おおむね木構造をなしていることが多いが、同一階層間のリンクなどの不規則なリンクも存在するため、同一階層のノードを円形に並べることによって同一階層間のリンクが重なることがなく、サブツリーをまたぐような不規則なリンクも見やすくなる。

##### 3.2.2 ノード配置のアルゴリズム

マップブラウザの表示におけるノード配置のアルゴリズムは以下の通りである(図2参照)。

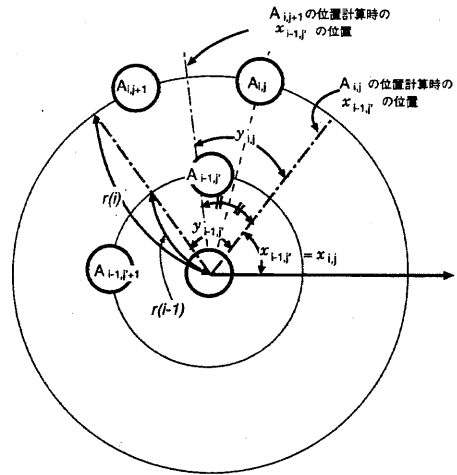


図2: 配置アルゴリズム

内側から  $i$  番目の同心円に表示される  $j$  番目のノードを  $A_{i,j}\{i: 0, 1, 2, \dots\}\{j: 1, 2, 3, \dots\}$  で表す。ただし、 $A_{0,1}$  を中心ノードとする。

1. ノード  $A_{i,j}$  は、 $i$  の関数  $r(i)$  の半径の同心円上に配置する。ノード  $A_{i,j}$  の基準角  $x_{i,j}$  はその親ノードの基準角  $x_{i-1,j'}$  で初期化する。
2. ノード  $A_{i,j}$  に割り当てられる角度  $y_{i,j}$  を以下の計算式で求める。

$$y_{i,j} = \frac{(\text{ノード } A_{i,j} \text{ のマップ上の子孫のノードの数})}{(\text{マップ全体に表示するノードの数})} * 2\pi$$

3. ノード  $A_{i,j}$  は中心から  $r(i)$  の距離、 $x_{i-1,j'} + \frac{y_{i,j}}{2}$  の角度の位置に配置する。
4. ノード  $A_{i,j}$  の親ノードの基準角の値を以下のように更新する。

$$x_{i-1,j'} = x_{i-1,j'} + y_{i,j}$$

以上の操作で、マップの中心となるノードから外側へ向かって順に計算することで、各ノードの位置を決定する。

表示するノードの配置が全て終わったらリンク関係があるノード間にリンクを描画する。リンクの参照、被参照関係を矢印で表す。

このアルゴリズムでは、末端ノードを均等な中心角で配置するため、局所的な表示ノードの集中が防げる。さらに、サブツリーごとにある程度のまとまりを持たせることができる。

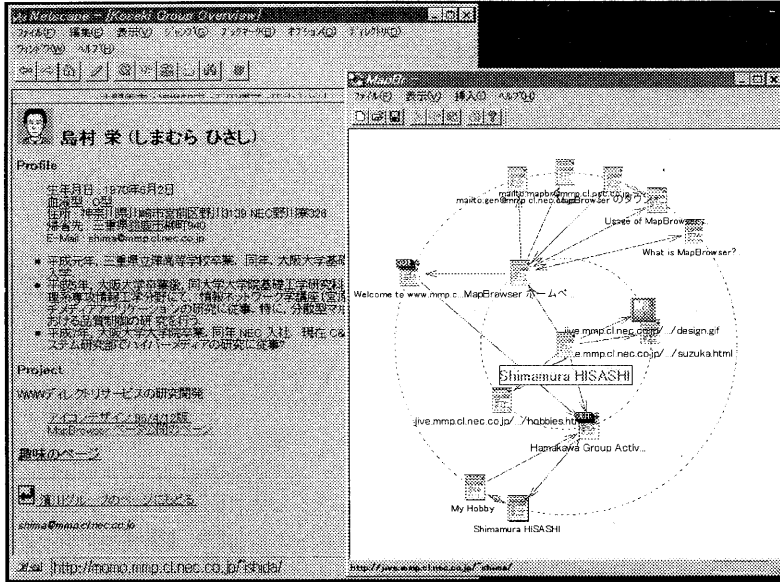


図 3: マップブラウザと Netscape の動作外観

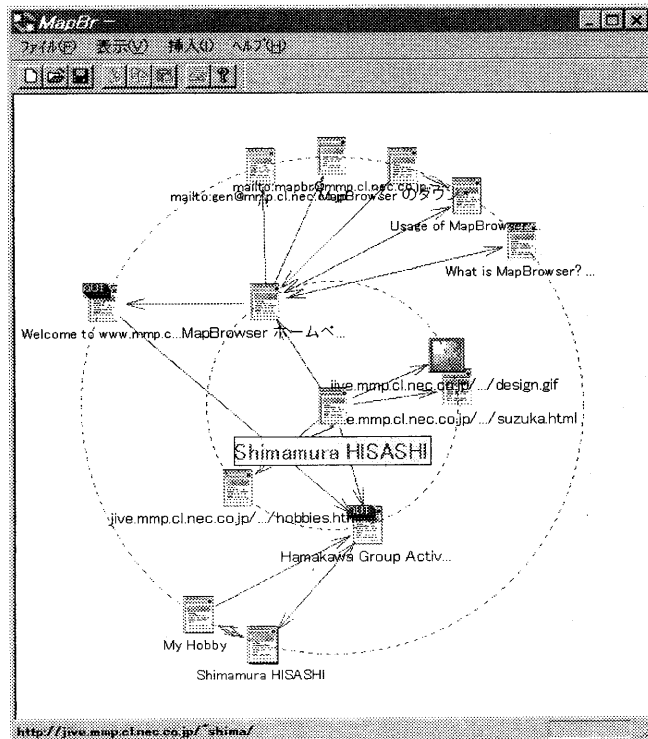


図 4: ノードリンク構造の表示例

### 3.2.3 表示の工夫

マップブラウザでは、WWWドキュメントのメディアの違いによって、マップ上でノードとして表示するビットマップを変化させている。図5はそのビットマップの一覧である。左からHTMLドキュメント、動画ファイル、静止画ファイル、音声ファイルを表す。

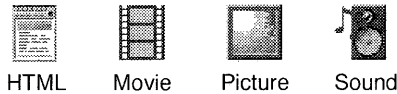


図5: メディアによるビットマップの違い

また、特定のノードからの参照リンクが非常に多くなる場合にマップ表示が見にくくなることもある。このため、一つのノードからの参照リンクの本数があらかじめ決められた値を越えた場合、その参照リンクおよび参照先のノードの表示を省略することとしている。同様に、被参照リンクが多すぎる場合もリンクの表示を省略している。

図6はリンク表示を省略したノードのビットマップの形状の変化を示している。あるノードからの参照リンクの表示を省略した場合、そのノードに「out」と書いたビットマップを張り付けている。また、被参照リンクを省略した場合は「in」と書いたビットマップを張り付けている。

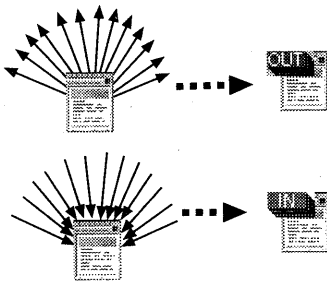


図6: リンク表示の省略

### 3.3 ノードリンク構造の取得

マップブラウザは、取得したWWWドキュメントの内容からノードリンク構造情報を自動的に抽出し、それを履歴情報として蓄積している。この履歴情報はマップブラウザ終了時に履歴ファイルに保存され、再起動時にはメモリ上に読み出される。

ノードリンク構造情報の取得の際のマップブラウザの動作手順は以下の通りである。

1. ユーザが新たなWWWドキュメントをNetscapeで取得(Open)する。
2. マップブラウザが保持する履歴情報内に、このドキュメントのノードリンク構造情報があるかどうかを調べる。
  - (a) ない場合、取得したWWWドキュメントを解析してノードリンク構造情報を抽出する。
  - (b) 新たに抽出したこれらの情報を履歴情報へ追加する。
3. 得られたノードリンク構造情報を反映したマップ表示を行なう。

上記の手順で収集されたノードリンク構造情報は、ユーザ情報とあわせて履歴情報として記録される。それぞれの項目は以下の通りである。

- ノードリンク構造情報
  - URL
  - タイトル
  - ドキュメントのメディアの種類
  - ドキュメント内容の最終更新時刻
  - リンク先のURLのリスト
- ユーザ情報
  - ユーザ定義のコメント
  - 最終アクセス時刻
  - アクセス頻度

### 3.4 アクセス頻度の反映

マップブラウザでは、WWWドキュメントを取得するたびに、対応するノードのアクセス頻度を増加させる。また、各ノードのアクセス頻度をある一定の時間が過ぎると減少させ、無制限に値が増加することを防いでいる。現在の実装では、5日間アクセスをしなかったノードはその頻度値を半分にするようにしている。

また、マップブラウザでは、アクセス頻度の高いノードのビットマップほど影を太く描画するようにしている。これによって、より直観的なアクセス頻度の把握が可能となる。

図7のa)がアクセス頻度が最小、b)が最大のHTMLドキュメントのビットマップである。

### 3.5 マップブラウザとNetscape間の連携

マップブラウザとNetscape間の連携には、Netscape Client Application Interface (NCAPI)[1, 2]を用いている。



図 7: 頻度と影の大きさ

ノードアイコン上でマウスクリックをすると、マップブラウザはそのノードを中心としたノードリンク構造を表示し、そのノードに該当する URL を NCAPI を介して Netscape へ渡す。Netscape は渡された URL の指す WWW ドキュメントを取得、表示する。

また、Netscape での操作により WWW ドキュメントを取得すると、NCAPI はマップブラウザ側のコールバック関数を介してその WWW ドキュメントの URL を通知する。マップブラウザは通知された URL にあたる WWW ドキュメントを中心としたノードリンク構造を表示する。

このようにマップブラウザで表示しているノードリンク構造は、常に Netscape で表示している WWW ドキュメントを中心としている。

## 4 ディレクトリサーバの利用

この章ではノードリンク構造をディレクトリサーバから取得するマップブラウザの開発に関して説明する。

### 4.1 ノードリンク構造情報の管理の問題とその解決策

3章のマップブラウザのノードリンク構造情報取得方式には2つの問題がある。第1に、アクセス履歴ファイルが無制限に蓄積され、ログファイルが大きくなりがちである。第2に、過去にアクセスした先のノードリンク構造情報しか利用できないため、初めてアクセスする WWW ドキュメント周辺のノードリンク構造情報を利用することができない。

これらの問題を解決するために、ここでは、ディレクトリサーバ [4] を利用する。ディレクトリサーバは広域の WWW ドキュメントの URL や最終更新日時、検索を行なうためのキーワードなどを管理する。この情報を利用して、マップブラウザは表示に必要な範囲のノードリンク構造をディレクトリサーバに問い合わせる。

また、このようなノードリンク構造情報の共有に対して、アクセス頻度などのユーザ固有の情報はログファイルとしてクライアントで記録、管理する。

以下では区別のために、3章のマップブラウザを履歴型マップブラウザ、この節で説明するディレクトリ

サーバを利用したマップブラウザをクライアント型マップブラウザと呼ぶことにする。

## 4.2 ディレクトリサーバとの接続

### 4.2.1 システム構成

履歴型マップブラウザと同様、クライアント型マップブラウザにおいても Netscape とマップブラウザは常に連携する (3.5節参照)。

マップブラウザは、CGI プログラムを通してディレクトリサーバの情報を得る。この CGI プログラムはマップブラウザからの要求に応じてデータベースから必要なノードリンク構造情報をマップブラウザへ返す機能を持つ (図 8参照)。

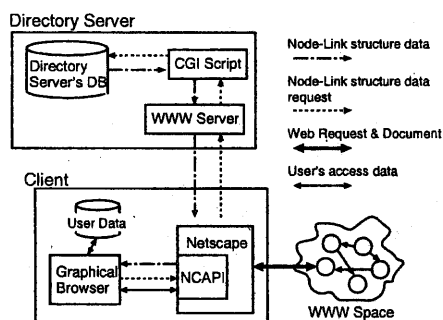


図 8: システム構成とデータフロー

### 4.2.2 ノード情報

ディレクトリサーバから取得するノードリンク構造情報は、以下の要素からなるノード情報の集合として扱う。

- URL
- タイトル
- ドキュメントのメディアの種類
- ドキュメント内容の最終更新時刻
- そのドキュメントからのリンク先の URL のリスト

マップブラウザは、動作中にディレクトリサーバから読み込んだノード情報を終了時までマップブラウザが確保したメモリ上にキャッシュする。

### 4.2.3 取得要求

マップブラウザは Netscape で表示されている WWW ドキュメントを中心とした、ある範囲のノード情報を

ディレクトリサーバに対して要求する。この範囲は中心のノードからのリンクの段数で指定される。このとき転送するデータ量を最小限にするために、マップブラウザはできるだけメモリ上にキャッシュしていないノード情報のみを要求する。以下ではこの方法について説明する。

1. 中心となるノードを0段目として、調査対象へリストアップする。
2. 調査対象としてリストアップされた  $n(= N)$  段目のノード情報が全てそろっているかどうか調べる。
  - (a) もし  $N$  段目で一つでもノード情報をキャッシュしていないノードが発見されたら、 $N$  段目以降のノード情報を全て要求する。
  - (b) すべてそろっていれば  $N$  段目のノードからリンクされている  $N + 1$  段目のノード情報を全てリストアップし、2. の操作を表示範囲の段数だけ繰り返す。もし、 $N + 1$  が表示範囲を越えていれば全てノード情報はそろっているとして取得要求は行わない。

上記の処理の結果、マップブラウザはディレクトリサーバの CGI スクリプトに対して、中心となるノードの URL、表示範囲段数、ノード情報を必要とする段数  $N$  を引数として取得要求を行なう。CGI スクリプトは、与えられた URL を中心とした表示範囲の段数以内のノード情報のうち、段数  $N$  以上のものをマップブラウザに返す。

### 4.3 ユーザ情報の管理

ユーザ定義のコメントやアクセス頻度からなるユーザ情報は、マップブラウザがログファイルとして記録する。

ノード情報は以下の項目からなり、このノード情報の羅列をユーザ情報としてログファイルへ記録する。

- URL
- ユーザ定義のコメント
- 最終アクセス時刻
- ユーザのアクセス頻度

クライアント型マップブラウザでは、アクセスをしたことのない WWW ドキュメントに関するユーザ情報は記録しない。また、履歴型マップブラウザと同様に各ノードのアクセス頻度をある一定の時間の経過で減少させるが、アクセス頻度が最低値まで落ちたノードのユーザ情報はログファイルから削除する。この機能によって、ログファイルのサイズが大きくなりすぎることを防いでいる。

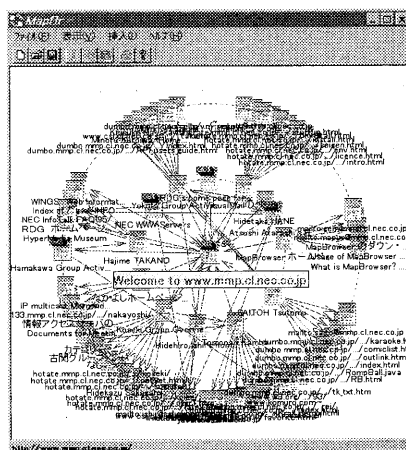


図 9: 複雑なノードリンク表示

## 5 評価と課題

ここでは開発したマップブラウザの表示方式とノードリンク構造の取得方式について評価する。

### 5.1 マップ表示

マップの表示に関しては、簡便なアルゴリズムを用いており、ノードリンク構造を取得する時間を除けば1秒以内に表示される。表示に関するオーバーヘッドは、Netscapeによるドキュメントの内容のダウンロードにかかる時間と比較した場合ほとんど問題にならない。

また、中心のWWWドキュメントからリンク2段階分のノードリンク構造の表示によってナビゲーションの際の見通しは良くなった。特に、円形階層グラフによるノードの配置により、階層構造において、サブツリーのまとまりが直観的に把握できる。

しかし、現状において以下のような問題点があげられる。

1. 多くのノードを含むページを表示した場合、ノードや文字列の表示が重なりあってわかりにくくなる場合が多い。リンクについても、バックリンクなどが多く張られている部分を表示した場合、マップが完全グラフ状になって非常に見にくくなる。(図9参照)。
2. あるノードからナビゲーションを行うと、マップ全体が一度に描き変わる。このため、前のマップからの遷移の様子がわからなくなり、たどった道筋がマップ上ではっきりしないため混乱することがある。

1の問題については、用途によっては必ずしもリンク関係を全て見せる必要はなく、特にバックリンクなどの表示は必要のない場合が多い。必要最小限のリンクを表示するような工夫が必要であると考えられる。また、タイトルやURLといったテキスト情報の表示の重なりに関する解決を図る必要がある。

2の問題については、ナビゲーションによるマップの再描画の際にアニメーションを導入して、マップ上のノードの位置変化を連続的にしたり、マップの変化に対してノードの表示位置が一定になるような描画アルゴリズムを導入することで解決することができると考えられる。

## 5.2 ノードリンク構造の管理

### ● 履歴型マップブラウザ

履歴型マップブラウザでは、アクセスを繰り返すにつれ履歴情報ファイルが非常に大きくなる。

このため、現状のテキストベースの履歴情報ファイルをバイナリ化したり、アクセス頻度の下がった情報の消去を行なうことにより、サイズをなるべく小さくする必要がある。

### ● クライアント型マップブラウザ

クライアント型マップブラウザでは、大量の履歴情報を管理する必要はなくなる一方、ディレクトリサーバのデータ量が増大するにつれてデータベースアクセスに必要な時間が大きくなってしまいうという問題が生じる。

この問題に関しては、データベース内のクラスタリングを行なうことによって、現在アクセスしているWWWドキュメントと類似した内容のクラスタや同一ドメインのクラスタの情報のみを視覚化の対象とすることによって、アクセスするデータ量を小さくできると考えられる。

## 5.3 アクセス頻度の記録、表示

アクセス頻度をグラフィカルに表示することは、個々のドキュメントがユーザにとって重要であるかどうかの手がかりとして非常に有効であることがわかった。また、絶対的なアクセス回数ではなく、一定期間で値が減少するアクセス頻度を視覚化することによって、よりタイムリーに重要だと思われるドキュメントのみをハイライトすることができる。

しかし、現状の影による頻度の表現はビットマップの表示サイズによっては分かりにくい。また、頻度値を減少させる期間、減少幅が実際の利用に適しているかどうかについては十分調査していない。これらの問題に関しては、より多くのユーザに利用してもらい、最適なビットマップサイズや頻度値の減少方法へ調整する必要があると考えられる。

## 6 おわりに

本稿は、従来のブラウザと連携し、ノードリンク構造とアクセス頻度をグラフィカルに表示する機能を付加したマップブラウザについて述べた。マップブラウザは全てのノードリンク構造を表示するのではなく、あるWWWドキュメントを中心とした限定された範囲のノードリンク構造をグラフィカルに表示することで局所的なナビゲーションを支援する。

このようなノードリンク構造のグラフィカルな表示によってナビゲーションが効率的になることがわかった。しかし、現在の表示方法ではノードリンク構造が少しでも複雑になると画面が混乱してしまう。今後は、クラスタリングも含めて、ノードリンク構造のよりわかりやすい視覚化方式の検討を進めていく。

## 参考文献

- [1] Garrett Arch Blythe, "Netscape's DDE Implementation", <http://home.netscape.com/newsref/std/ddeapi.html>, Mar. 22 1995.
- [2] Garrett Arch Blythe, "OLE Automation in Netscape", <http://home.netscape.com/newsref/std/oleapi.html>, Mar. 22 1995.
- [3] John Lamping, Ramana Rao, and Peter Pirlli, "A Focus + Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies.", CHI '95 MOSAIC OF CREATIVITY, May. 7-11 1995.
- [4] 高野, 久保, 島村, "WWWナビゲーション環境の試作(1) ~ディレクトリサービスの構成~", 第52回情処全国大会, Mar. 7 1996.
- [5] 久保, 高野 "WWWナビゲーション環境の試作(2) ~ノード/リンク探索ツールの作成と評価~", 第52回情処全国大会, Mar. 7 1996.
- [6] 島村, 高野 "WWWナビゲーション環境の試作(3) ~マップブラウザの試作と評価~", 第52回情処全国大会, Mar. 7 1996.
- [7] 島村, 久保, 高野 "ディレクトリサーバを利用したWWWグラフィカルブラウザの開発", 第53回情処全国大会, Sep. 4 1996.