情 報 学 基 礎 41－7
（1996．4．18）

異種データに対する統合的情報検索

諸橋 正幸　武田 浩一

日本アイ・ビー・エム株式会社 東京基礎研究所

テキストを中心とした情報検索の枠組の中にマルチメディアデータや数値データなどの異種データを取り込んで、統合的に検索するためのパラダイムを提案する．この試みにおいて問題となるのは，データ属性の不均一をどう取り扱うか，データの種類に強く依存した検索アルゴリズムがそれぞれの分野で開発されてきているが，統合化されたパラダイムの中でこれらをどう生かすか，異種の該当データが混じって戻ってくる結果をユーザにわかりやすく表示するにはどうするか，といった問題を含む．これらの問題に対する解決策を提案するとともに，解決策に沿った検索実験システムの構造を述べる．

Uniform Way for Retrieving Heterogeneous Data

Masayuki Morohashi, Koichi Takeda

IBM Research, Tokyo Research Laboratory

1623-14, Shimo-turuma, Yamato-shi, Kanagawa 242 Japan

{moro, takeda}@trl.vnet.ibm.com

We propose a uniform way of retrieving heterogeneous data that include not only text data, but also so-called multi-media data, such as still pictures and motion pictures, and numerical data that are represented by RDB. The proposed paradigm solves the problem how to represent datatype-dependent non-uniform attributes, the problem how to enable datatype-dependent search engines, and also the problem of presenting the characteristics of the retrieved data set that includes heterogeneous data.

1. はじめに

様々なデータの電子化が進むにつれて，従来のテキストを中心とする情報検索（文献検索）システムの中に，マルチメディアデータ（静止画，動画など）を取り込んだシステムが見受けられるようになってきている[1][2]．また，インターネットにおけるｗｅｂブラウザの利用者などのように，色々なサイトの情報を気軽に覗きにゆくユーザにとっては，データの種別に関係なく，電子化された情報ならば何であっても（もちろん，公開されている物に限定はされるが）アクセスしたいという要求が強い．こうしたユーザの要求に呼応する形で，高速の全文検索サービスを可能にする試みが行われ[3]，HTML形式で表現されたテキスト，画像，音声などのマルチメディアデータが検索可能になってきている．ただし，この場合には，サーチの対象がテキストであるため，画像などのデータを表示するには，一旦，テキストを中心とするホームページを経由してから，ボタンやアンカーなどを辿るケースが多いこと，検索機能自体がまだ，従来の情報検索で提供されているものより劣る点があること，などの制約がある．

一方，数値データを中心とするリレーショナルDBMSでは，頻繁に変更のあるデータを対象として，データのインテグリティの問題に重点をおいたシステム作りがされてきており，「情報検索」とは一線を画するものであったが，最近，ユーザの視点からモデルを見直す動きが出てきて，OLAP[4]のように，個々のレコードではなく表全体としてのデータの検索に重きをおくようなモデルが提唱されている[4]．しかし，これと情報検索とを積極的に結びつけようという試みはまだなされていないようである．

ここでは，数値データを含めたマルチメディアデータの検索を統合的に行いながら，なおかつ，個々のデータ種別（テキスト，画像，表で表す数値情報など）に特化して発達してきたシステムの長所を損なわないような仕組を実現するための情報検索モデルを提案する．これにより，ユーザは，データの種別が，単に表現法のバリエーションに過ぎないという感覚で，種々のソースから必要な情報を集めることができ，得られた情報から多角的な理解をすることが可能になる．

2. Attribute-Value Pair

多様なデータを対象にして統一的な検索を可能にするには，対象とするすべてのデータに対して統一的パラダイムを与える必要がある．筆者らは，この統一的パラダイムとして，すでにRDBなどで広く用いられているattribute-value pairのパラダイムを導入する．

実際，書籍を始めとしてビデオやレコードのような異種データを扱う図書館においては，伝統的な情報整理の技術である書誌情報を記したカードによってこれら異種データを統一的に扱ってきているが，この書誌情報の記述法は，一種のattribute-value pairでデータを統一的に捉えていると考えられる（図1の書誌情報の例で，第１列がattributeで，第２列がvalueとなる）．

書名	xxxxxxxxxxxxxxx
著者名	xx xxx
発行年月日	19xx年xx月xx日
発行所	xxxx出版
キーワード	xxxx, xxxx, xxxx
分類コード	xxxx-xx
ISBN	x-xxxx-xxxx-x

図1 書誌情報による属性表現

ここで，通常は，文献の所在情報を示すのが，分類コード（コード順に，本が棚に置かれている場合）あるいはISBN（発注などの場合）であり，その他の属性は検索用の条件指定に用いられる．すなわち，ISBNを主キーとするレコードとして表現され，主キー以外の属性から必要とするレコード（書誌カード）を検索するリレーションとなっている．

ただし，書誌情報をそのまま検索対象データに対するattribute-value pair表現として検索システムを構築する，いわゆる図書館システムには，いくつか問題がある．たとえば，

1) キーワードのように，あらかじめ最大数の設定されていない多値を値として持つ属性がある．

2) 書名などのようなテキスト情報に対して高速な部分文字列のサーチ機能が要求される．また，文字列一致に対する特殊な条件設定（隣接する，出現順序を保つ，など）が要求される．

3) 書誌カードには，コンテンツが記述されていないので，全文検索のように内容のテキスト部分で検索するような検索が出来ない．

4) 雑誌や年鑑類などのように通読しない文献について，該当項目／論文での検索が出来ない．

5) データの種類に依存しない共通の属性のみがカードに記述される．この場合，データ種別に依存する属性で検索ができない．

などである．

1)，2)を解決するには，RDBMSのような単一の検索機構でなく，後述するように，attributeごとに異なる検索エンジンを指定できるようなマルチエンジンのシステムにすればよい．また，オブジェクト指向のパラダイムにのせるならば，「テキスト」というデータタイプで定義される属性には，キーワードに基づく検索演算や全文サーチに基づく検索演算がメソッドとして用意されていると見ることになる．

3)も1)，2)と同様，従来の紙を中心とする処理やRDBMSの枠組からくる制約であり，既存のシステムでインプリメントすることにこだわらなければ，問題とはならない．逆に，電子図書館としての新たなシステムを構築するという考えに立てば，電子的な書誌カードとしては，コンテンツ（特にテキストで表現されたコンテンツ）も属性の一つとして考えるべきである．

4)は，登録するデータの単位をどうするかという問題であり，コンテンツが完全に電子化されて，実態としての書籍と必ずしも一対一になっている必要がなくなれば，自然と避けられる問題である．ただし，検索対象としてのデータが，元の本の単位よりも小さい物としたとしても，実際にコンテンツを見る場合には，元の本の単位が再構成される必要はあろう．たとえば，学術雑誌などにおいては，同じ号に載った別の論文は，通常は，検索により見つけた論文とは何の関連もないが，ある特定のテーマについての特集号だった場合には，前後の論文を，あるいは，「巻頭の言」を参照する必要が出てこよう．また，新聞における記事の検索を例に取っても，ユーザによっては，単に関心のある記事を読むだけでなく，当日の新聞における記事の扱い方（たとえば，第何面のどの辺りで掲載したかという情報から，新聞社のその記事に対する軽重が判断できる）をも問題にするかもしれない．

こうした要求に応えるには，検索の単位としてのデータの範囲を細かくすると同時に，リンクという形で，データ間の前後関係を保存したり，あるいは，「目次」にあたるデータから，本を構成する各要素データへのリンクを保存する方法をとる（あるいは，その両方のリンクを保存する）．

5)は，valueとしてN/A (Not Applicable) を導入することで解決する．図２において，検索システムが扱う，すべてのデータが持つ属性のunionをとったattribute-value pairを拡張書誌情報カードと見なせば，全データに共通な属性部分については，正規化された表の形で表現でき，データタイプに帰属する検索メソッドによる検索が常に可能である．

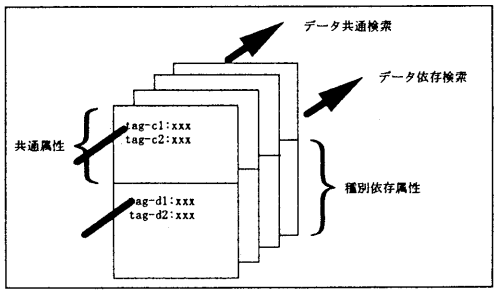

図2 拡張書誌情報をベースとする検索概念

データ種別に依存して現れる属性部分について，N/Aを値として持つ属性は，そのデータには存在せず，従って，検索の際にそのデータはサーチの対象から外れるという扱いをとれば，共通属性部分と同様に，これらの属性に対して

も検索が可能となる．いささか直感的な説明になるが，図２の任意の属性（tag）部分で串刺しにして該当カードだけが残るようにするには，N/A値部分は，そこに串を入れた時に，常に落ちるようなパンチを入れておけばよいということになる．

さらに，値がN/Aとなる属性については，書誌カードに記述する必要がないとしておけば，データ集合をオープンにすることが可能になる．

```
<meta>
<orgname>
<name>平成五年通商白書＜総論＞</name>
</orgname>
<orginfo>
<orgtype>ＳＧＭＬ</orgtype>
<orgsite>tg93.sgm</orgsite>
</orginfo>
<contents>
<pubbib>
<mtitle>
平成五年通商白書＜総論＞
</mtitle>
<puborg>大蔵省</puborg>
<pubdate>平成5年6月10日</pubdate>
<formtype>Ａ５</formtype>
<npages>419</npages>
<mnum>4-17-270356-9</mnum>
<otherbib>通商産業省</otherbib>
</pubbib>
</contents>
</meta>
```

図3 新産業創造ＤＢにおける書誌データ例

筆者らのシステム構築経験では，日経新聞１年分のデータを基にした実験システムの構築においては，図２にあるような，１行の最初のトークンとして属性名を書き，残りの部分に値を記述する形式のattribute-value pair記述を行い[5]，情報処理振興事業協会（IPA）の新産業創造データベースプロジェクトにおいては，図3に示すようなSGMLによる同様の記述を行った[6][7]．

図３のように，SGML（あるいはHTML）で書誌情報を記述した場合，この書誌データ自体が表示を目的とした形式を備えているという利点がある．言いかえれば，検索システム構築にあたって，特別のブラウザを用意しなくても，ユーザはいつでも，書誌情報を覗けるということになる．特にインターネットなどで公開する場合には，有利といえる．なお，この例では，実際の白書の内容は，物理的に書誌データに含まれているわけではなく，ファイル名によるリンクで辿るようになっいる．

もともと，RDBの表として記述されているデータの扱いについては２つの対処法がある．検索対象がレコード（複数可）の場合は，書誌情報の表形式表現と，今，扱おうとしているデータ表現は，全く同じ検索方法が意味を持つため，この枠組をそのまま用いることでデータの検索が可能になる．すなわち，表の各レコードが１つの書誌データと考えればよい．しかし，情報検索が相手にするデータの粒度から見て，検索対象となる数値データは，通常，かたまりとしての「表」ということになろう（ただし，この時の「表」は物理的な表でも，正規化された表でもなく，ユーザの要求に合わせたビューである）．その場合，表のattributeは，いわばその表に対する「メタ情報」（表の名前，表を構成する属性の数と各々の属性名，表の意味するところ，など）を，書誌データとして記述することになる．

3. Information Outlining Paradigm

筆者らは，文献[5]などにおいて，検索していく過程の中で，網にかかっているデータ集合の輪郭をつかむための情報を，常時，表示し，その中からユーザにとって価値のあるものだけに絞り込む次のステップに手がかりを与え，さらには，その手がかりをマウス等で触ることでナビゲートする方法を提案し，Information Outliningと名付けた．このパラダイムを前述のattribute-value pairの捉え方で図示すると図４のようになる．

3.1 検索

図の中心の位置にあるattribute-value pairの平面から上がいわゆる検索のためのパラダイムであり，下がデータ収集パラダイムとなる．検索パラダイムは，データ→モデル→ビューという繋りでユーザに表示され，ビューを通じて与えられた検索条件が，前述の繋りを逆方向に，すなわち，ビュー→モデル→データとわたり，条件にあったデータのみが，検索結果として保持されることになる．この条件にあったデータは再びシステムで用意されたモデルを通してビューに伝えられるため，検索過程において，網にかかっているデータをいつでも，複数のビューを通して，眺められるようにするため，ユーザはデータの海の中で迷子になることなく，探しているデータにたどりつけることになる．図においては，attribute-value pair平面上に描かれている点線部分がこのモデルに相当し，ビューを通して与えた検索条件が実際に働くのは，このモデル上である．

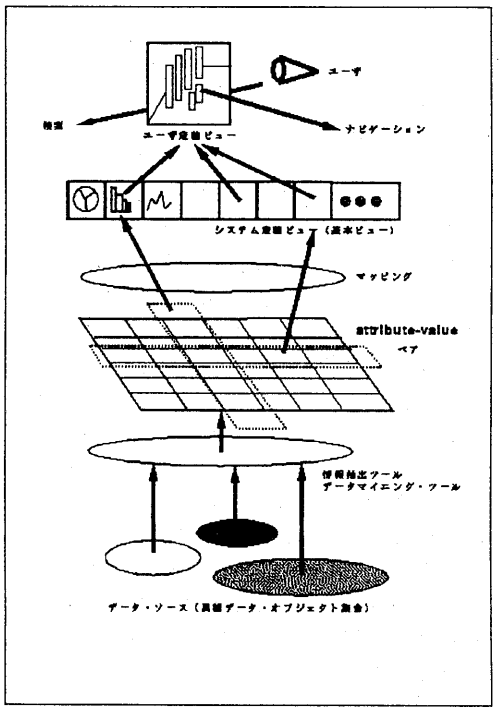

図4 Information Outlining Paradigm

モデルとビューの間は，m対nのマッピングになる．これは，同じattributeを表示するのにいくつかの表現方法がありうることを示している．たとえば，該当データの発行年を見るモデルが定義されていたとして，それを年ごとのデータ数で集計し，表形式で表示したり，棒グラフで表示したり，あるいは，発行データの多い順に並べたパイチャートで表示したりするには，1対nの関係を許す必要がある．逆に座標軸の意味さえ変えれば，グラフ表示などはデータタイプに独立な形で作ることができる．

3.2 モデルと検索エンジンの対応

検索は，ユーザがビューを通して与えた条件を，そのビューに対応するモデルのもとで行うことになるが，その際，どの検索エンジンでも実行が可能というわけにはいかないのが現実である．理想からいえば，検索のためのユニバーサル演算を与え，各データ種別について，その演算定義をあたえればよいのだが，そのようにして定義した演算は，今まで個々に発展してきたサーチ機構のほとんどを使えなくしてしまう結果になることは，ほとんど明らかである．現実的解決としては，データモデルと検索エンジンのそれぞれを，ユーザが別個に選べるようにして，意味のない組み合わせ指定をされた場合にのみ警告を出すようなシステム設計をとることになるであろう．ただし，その場合でも，各モデルに対して，エンジンの指定がない時に動くdefaultのエンジン設定はできるようにしておくべきである．

3.3 属性の構造化

属性値のデータタイプが階層をもつ場合がある．たとえば，年月日は年→月→日の階層を持つし，地名も，日本の行政区画でいえば，都道府県から市郡区，町村，番地まで階層化された構造を持つ．こうした，構造化された属性に対する検索は，従来の情報検索システムにおいては，同義語辞書（あるいは類義語辞書）の採用などで実現されてきた．しかし，ユーザは検索結果の一覧（あるいは，結果の総件数）というビューしか持たないため，場合によっては，自分がどのような条件で検索したかも定かでないというケースも起こりかねない状態にあった．

文献[8]では，階層を持った属性に対する色々なビューの試みが，かなり網羅的に紹介されているが，その他にも，表計算ソフトが表示する表形式におけるドリルダウンやディレクトリをファイル表示に用いられるフォルダーのメタファも有力な表示法である．

3.4 属性値の自動抽出

書誌データに立ち戻ってデータの入力を考えると，ほどんどの属性は，人手で振られているのが現状であった．キーワード検索や全文検索は，キーワードという限定された属性に対して，その人手による入力の手間を省こうという発想で開発されたシステムである．しかし，その他の属性に関しても，値の自動付与の可能性のあるものがある．たとえば，タグを手がかりとする属性の構造化文書からの抽出，語の出現頻度を手がかりとする自動分類[9]などが候補となる．

4. 検索システムの構築

以上，ここに提唱する検索パラダイムに則って，筆者らは検索システムを構築してきた．ここでは，その検索システムの概観を紹介するとともに，パラダイムの実現法を述べる．

4.1 システムの概観

システムの概観図を示す．

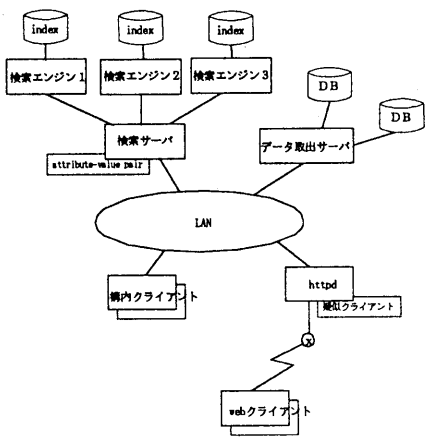

図5 システム概観図

検索サーバで扱うattribute-value pairは，contentそのものを含んでおらず，データのidを持っている．従って，クライアントからの（ある特定のモデルを通した）検索要求に対するサーバの答は，データidの集合である．この中から，実際に内容を見たいデータがある場合には，ユーザは，改めてクライアントからデータ取り出し要求を，データ取出サーバに向けて流してやることになる．

このように検索サーバとデータ取出サーバを分離することで，データの送り出し操作は，検索エンジンとは独立になる．また，検索の過程において，書誌データの表示も可能にするために，書誌データそのものも，対応するデータとともにＤＢに入れておく．書誌データは，検索のための特殊な構造に変換された形で検索サーバが持っているが，ここから再構成して表示可能な形式に戻すことは，処理速度の面から実現が難しいので，記憶スペースの無駄にはなるが，ＤＢ中にもユーザへの表示を目的として保存する方式をとった．

webクライアントに対するサポートは，httpdが，疑似クライアントとして振る舞うことで，サーバに影響を与えない形で実現した．また，web browserを検索クライアントとして使う場合の問題点として，一回の検索過程（条件を出して該当するデータのリストをもらう過程）が１つのセッションとして完結するために，次の検索過程で，直前の状態を保存できないことが指摘されているが，これを避けるために，ホームページごとにそれまでの状態を覚えておき（該当データの集合を覚えるのではなく，検索条件の過程を覚えておき），それに引き続く検索にあたっては，それ以前に出した検索条件とのANDという形で検索する．この方式では，検索の過程が進むに従って，検索速度が遅くなるという状態を引き起こすが，実際には，2-3回追加条件を入れる程度で，該当データ件数が２桁以下になるケースがほとんどなので，速度の劣化はそれほど問題にならない．

4.2 モデルの定義と階層属性の定義

属性の決定やモデルの定義は，データに依存するものであり，システム構築とは独立に行う

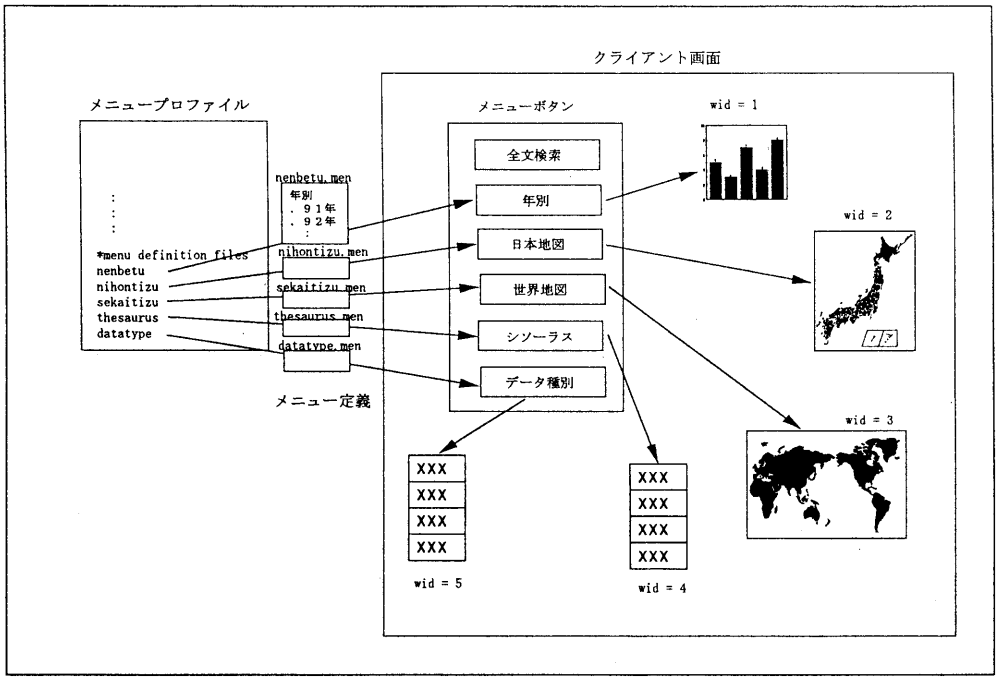

図6 モデル定義

べき作業であり，この定義が，どれだけシステムの制約から自由で有るかが，システムの汎用性を計る鍵となる．

我々の検索システムにおけるモデルの定義と属性の定義は，図6に示した形で行われる．図の中で，システムがどのようなモデルを使用するかを決めるのが，メニュープロファイル中の記述である．ここに示すメニュー定義ファイルの数だけシステムはモデルを用意し，クライアント画面には，これに基づいてモデル選択のボタンが表示される．各モデルの定義は，個々のメニュー定義ファイルに記述される．図7にメニュー定義の例を示す．点の数で階層の深さを示す形式で，検索条件を指定する属性名「日本」とその中の値の階層構造を指定する．この形式で指定できるのは，現時点では，外延的定義の可能なデータタイプを属性値として持つ属性だけであり，たとえば，任意のキーワードを許すような全文検索用の文字列入力フィールドは，あらかじめ，システム固有の定義が存在する．

日本
. 北海道
. 東北
. . 青森
. . 岩手
. . 宮城
. . 秋田
. . 山形
. . 福島
. 関東
. . 茨城
:
:

図7 階層型属性「日本の地名」による
モデルの定義例

なお，この任意文字列の入力フィールドとして設定されているモデルは，検索条件中にサーチすべき属性名を指定できるようになっているが，指定がない場合は，可能なすべての属性に対してサーチを行う．

現時点では，インプリメントしていないが，数値による比較条件を指定する属性に関しても，外延定義ができない場合がほとんどなので，上記と同じ扱いが必要となる．

4.3 モデル-ビューワ間マッピング

図6にあるように，現時点のシステムでは，このマッピングに関してはまったく自由度のない作りとなっている．すなわち，最初に定義する３つのモデルに対応して，３つのビューワ（棒グラフ，日本地図，世界地図）が対応する．それ以後のモデルに対しては，属性値の階層表示に従って該当件数を表示するビューワが対応する．

4.4 属性値の自動抽出指定

データ種別ごとに属性とその処理方法を指定する定義ファイルが存在する．図8において，左辺の大文字表示されているタグが，処理すべき方法を示し，左辺がその対象となる属性を示している．たとえば，「KWD」は，キーワード抽出すべき属性の指定を要求しており，「SWP」というデータ種別に対して用意される書誌データ中のタグ「volid author data」で囲まれたエリアからキーワード抽出を行うことを指示している．

```
DTD       xxxx.dtd
DATATYPE  orgtype
DATACAT   SWP
CONTENT   section chapter part
TITLE     title
LINEFEED  P
KWD       volid author date
```

図8 属性値の自動抽出指定の例

5. おわりに

本文中にも述べた通り，現段階では，まだ，パラダイムとして提唱している概念と現実のシステムの間には，ギャップがあり，完全にはデータ独立なシステムにはなっていない．そのギャップの中で，我々が当面重要と考えているのは，モデル-ビューワ間マッピングの問題である．すくなくとも，ビューワ自身をエンドユーザやＤＢ管理者が開発していくことは期待できないので，我々自身が品揃えを計るか，汎用のツールを期待するのが妥当であろう．インターネット環境でのビューワの品揃えという意味では，Java[10]などの利用が考えられよう．

参考文献

[1] J. Hong, J. Takahashi, M. Kusaba, "A Motion Picture Archiving Tech., and Its Appl. in an Ethnology Museum"

[2] NSF, "NSF ANNOUNCES AWARDS FOR DIGITAL LIBRARIES RESEARCH," Announcement Letter, Sept. 1994.

[3] http://www.opentext.com, http://www.lycos.com, など

[4] 特集「データウェアハウスの実践」DATABASE SYSTEM, Vol.2, No.3, 1996

[5] M. Morohashi, K. Takeda, et.al., "Information Outlining - Filling the Gap between Visualization and Navi. in Digital Libraries," Intl.Symp. on Digital Libraries, Tsukuba, Aug. 1995

[6] 浦本, 諸橋 "Information Outlining -検索情報の可視化-" 情処-情報学基礎40-7, 11/95

[7] M. Morohashi, N. Uramoto, "Information Outlining for Government-Issued Data," MultiMedia Japan, Yokohama, March 1996

[8] R. Rao, et.al., "Rich Interaction in the Digital Library," Comm. ACM, Apr. 1995

[9] M. Morohashi, S. Umeda, "Statistical Anal. of Japanese Text by Kanji-Usage Freq. & Its Applications," ASIS workshop, 11/91

[10] J. December, "Presenting Java," Sams.net Publishing, 1995

[11] 上田, 増田, 石飛, "CastingNet," ComSoftW. Vol.12, No.4, 1995