12－開－03

保存本

データベースの構築および技術開発促進事業

情報可視化によるドキュメント構造化の調査研究

平成13年3月

財団法人　データベース振興センター

委 託 先　株式会社日本総合研究所

KEIRIN

この事業は、競輪の補助金を受けて実施したものである。

12－開－03

データベースの構築および技術開発促進事業

情報可視化によるドキュメント構造化の調査研究

平成13年3月

財団法人　データベース振興センター

委 託 先　株式会社日本総合研究所

序

データベースは、わが国の情報化の進展上、重要な役割を果たすものと期待されている。今後、データベースの普及により、わが国において健全な高度情報化社会の形成が期待される。さらに海外に対して提供可能なデータベースの整備は、国際的な情報化への貢献および自由な情報流通の確保の観点からも必要である。現在わが国で流通しているデータベースの中でわが国独自のものは約半数であるが、わが国データベースサービスひいてはバランスある情報産業の健全な発展を図るためには、今後もわが国独自のデータベースの構築およびデータベース関連技術の研究開発を強力に促進し、データベースの拡充を図る必要がある。

このような要請に応えるため、(財)データベース振興センターでは日本自転車振興会から機械工業振興資金の交付を受けて、データベースの構築および技術開発について民間企業、団体等に対して委託事業を実施している。委託事業の内容は、社会的、経済的、国際的に重要で、また地域および産業の発展の促進に寄与すると考えられているデータベースの構築とデータベース作成の効率化、流通の促進、利用の円滑化・容易化などに関係したソフトウェア技術・ハードウェア技術である。

本事業の推進に当って、当財団に学識経験者の方々で構成されるデータベース構築・技術開発促進委員会（委員長　東海大学教授　上條史彦氏）を設置している。

この「情報可視化によるドキュメント構造化の調査研究」は、平成１２年度のデータベースの構築促進および技術開発促進事業として実施した課題の一つで、当財団が株式会社日本総合研究所に対して委託実施したものである。この成果が、データベースに興味をお持ちの方々や諸分野の皆様方のお役に立てば幸いである。

なお、平成１２年度データベースの構築促進および技術開発促進事業で実施した課題は次表のとおりである。

平成１３年　３　月

財団法人　データベース振興センター

平成１２年度　データベース構築・技術開発促進事業委託課題一覧

分　　野	No.	課　　題　　名	企　　業　　名
一　　般	1	学びに活用する簡易ＰＣサーバ・システム、マルチメディア・データベース	（株）トライアード・プロジェクト
	2	ＧＩＳと全文検索エンジンを連携した文章管理システム	（株）創建
	3	情報可視化によるドキュメント構造化の調査研究	（株）日本総合研究所
	4	先端産業の企業検索用シソーラス作成と企業検索システム構築	（株）日経リサーチ
	5	インターネット環境における博物館型地図画像データベースの構築	（財）地図情報センター
地域振興	6	地域住宅地図情報システム〔地域住宅ＧＩＳ〕	石井測量設計（株）
	7	阿蘇の楽しみデジタル図鑑作成	（財）阿蘇町地域振興公社
	8	Ｂ to Ｓ共働情報マッチングＤＢパイロットシステムの作成	（株）八幡コンピュータセンター
	9	Ｗｅｂ型ＧＩＳを利用したバリアフリーデータベースのプロトタイプ作成	（株）札幌ネクシス

目　次

1. 本事業の目的 1

2. 可視化の基本技術 2

2.1. グラフィックライブラリ 3
2.2. データ表現 5
2.2.1.一次元、二次元、三次元 5
2.2.2.多次元 5
2.2.3.木 6
2.2.4.ネットワーク 6
2.3. 日本語パーサー 7
2.4. ドキュメントベクトル化 8
2.5. データベース管理システム 9
2.6. 統計分析 10
2.7. ニューラルネットワーク 11

3. インターネット上の情報と可視化 12

3.1. 情報可視化 12
3.2. 可視化のプロセス 13
3.3. インターネット情報の可視化 13
3.3.1.ブラウジングと検索 14
3.3.2. Webデータと半構造性 16
3.3.3. Webデータの自動分類 16
3.3.4.ドキュメントの構造発見 18
3.3.5.評価 19

4. タスクとユーザ 20

4.1. ドキュメントオーサリングにおけるタスク 20
4.1.1.情報管理 20
4.1.2.仮説生成と検証 21
4.1.3.情報検索 22
4.2. 可視化におけるユーザ特性 24
4.2.1.経営者 24
4.2.2.情報管理者 24
4.2.3.一般ユーザ 25
4.2.4.顧客 25

4.2.5.ユーザとタスクの関係 27
5. 可視化技術とツール 29
5.1. 可視化技術のタキソノミー 29
5.2. 焦点・文脈化 30
5.2.1.情報のズーミング 32
5.2.2.ドキュメントレンズ 32
5.2.3.その他の焦点・文脈化システム 40
5.2.4.評価 40
5.3. インタラクション 41
5.3.1.動的質問 41
5.3.2.動的分析 42
5.3.3.評価 42
5.4. ドキュメント可視化 44
5.4.1.サルトンのシステム 44
5.4.2.リンのシステム 45
5.4.3. GalaxyとThemeScape 46
5.4.4. GalaxyOfNews 49
5.4.5. Starlight 56
5.4.6.評価 57
5.5. インフォメーション・アウトランニング 58
5.5.1.システム概要 58
5.5.2.評価 59
5.6. ソフトウェア可視化 61
5.6.1.ビジュアルプログラミング 62
5.6.2. SeeSoft 63
5.6.3.評価 67
5.7. オーサリングツール 68
5.7.1.システム概要 68
5.7.2.評価 70
6. インターフェースモデル 71
6.1. インターフェース設計ガイドライン 71
6.2. 設計プロセス 73
6.3. 可視化システムのモデル 75
6.3.1.構造化 75

6.3.2.情報入手の効率性 75
6.3.3.理解のしやすさ 77
6.3.4.可視化 79
6.3.5.操作性 80
6.4.ユーザ・タスク・機能 81
7. 結語 83
関連図書 84

図目次

図3-1 可視化の手続き 13
図3-2 完全二部グラフの例 18
図5-1 スクロールの問題点 31
図5-2 WebBookの画面例 33
図5-3 Magic lens出力 34
図5-4 各ノードの計算例 35
図5-5 フラクタルの自己相似性を利用した木のレイアウト 37
図5-6 フラクタルビュー 39
図5-7 フラクタルビューによるCプログラムの表示 39
図5-8 相互作用 41
図5-9 Film Finder 43
図5-10 サルトンのシステム 44
図5-11 リンのシステムの学習例 46
図5-12 ThemeScape 48
図5-13 Galaxy 49
図5-14 GalaxyOfNewsの初期画面 51
図5-15 GalaxyOfNewsのモデル 52
図5-16 GalaxyOfNewsのキーワード階層 53
図5-17 GalaxyOfNewsアーキテクチャ 54
図5-18 GalaxyOfNews ARN 55
図5-19 StarLight 57
図5-20 インフォメーション・アウトライニング 60
図5-21 SeeSoft 線によるコードの表示 64
図5-22 SeeSoftにおける3通りの表示 64
図5-23 バグフィックスの履歴 66

表目次

表2-1 データ形 5
表4-1 情報検索のタスク 23
表6-1 SIDEによるユーザインターフェース設計項目 72
表6-2 ナレッジマネジメントのタスクとユーザ 81
表6-3 タスクと可視化 82

1. 本事業の目的

キーワード作成をはじめとする、ドキュメントオーサリングにあたり情報可視化技術によって支援する方式に関して調査・研究を行う。

ここでの可視化とは、情報のスナップショットをグラフィカルに表示することではなく、ユーザとのインタラクションによってさまざまな側面や抽象度から表現される情報が動的に変化し、それを適切な様式で表現する機能を指す。

情報の可視化は、複雑な計算結果の表示やOLAP(On-Line Analytical Processing)における対話的分析に用いられてきたが、情報量が多い、さまざまな構造化情報が抽出可能である、などのことからドキュメントのオーサリングにも情報可視化が有効であると考えられる。さらに、書誌情報やメタデータなどのドキュメントデータを用いることにより、豊富な属性に基づく情報可視化が可能になると予想される。

本提案では特に、インターネット上の情報の構造化に焦点を当て、オーサリングを行うユーザの目的、収集されたドキュメントの状態、出力される構造化情報の各要因ごとに要求される機能と、インターフェースモデルに関して提案を行う。

また、ナレッジマネジメントなどの、企業における知識資産のオーサリングを想定し、そこで求められる機能に関する調査、および機能実現に関する提案を行う。

2. 可視化の基本技術

可視化を支える基本技術として、グラフィックライブラリ、データ表現、そしてドキュメント処理技法について、その概要を述べる。本章で概説を行う基礎技術は、以下の技術である。

- グラフィックライブラリ
- データ表現
  - 一次元、二次元、三次元
  - 多次元
  - 木
  - ネットワーク
- ドキュメント処理技法
  - 日本語パーサー
  - ベクトルドキュメントデータベース
  - データベース管理システム
  - 統計分析
  - ニューラルネットワーク

これらの技術のうち、グラフィックライブラリやデータ表現はドキュメントオーサリングに限らず、可視化全般に関与する技術である。また、ドキュメント処理技法はドキュメントオーサリングに特有の技法である。

データ表現やドキュメント処理技法に関して、ここ数年、大きな進展はない。一方、グラフィックライブラリに関しては、ハードウェアの低価格化に伴って、高性能な表現が可能なグラフィックライブラリが利用可能になっている。

2.1. グラフィックライブラリ

グラフィックライブラリの進展と機能を概観するために、OpenGL[1]および、その上位レイヤーであるVRML[2]の概観を行う。

OpenGLは、プログラマーが高品質なカラーイメージで3Dオブジェクトを生成するために開発されたグラフィックハードウェア（たとえば、SGIのワークステーション）のために用意されたソフトウェアインターフェースである。

OpenGLはレンダリング処理のみを行い、グラフィックハードウェアに依存しない、中立なAPIとして2Dと3Dのグラフィック関数(モデリングを含めて、座標変換、色、ライティング、スムース・シェーディング機能やテクスチャー・マッピング、ナーバス、フォグ、アルファー値ブレンドとモーションブラー)を提供している。OpenGLは、イミディエートモードとリテインドモード(ディスプレイリスト)の両方のモードを提供している。

さらに、OpenGLをインターネット上での情報提供と融合させることによって、様々な仮想三次元空間をインターネット上で構築することが可能になった。VRMLは3D CGを表現する言語であり、Web上のリソースと同じ感覚で、さまざまな画像的インタラクションを行える。そのため、以下のような特徴を有している。

（1） 3D CGのテキスト表現

3D CGを構築する三次元物体の座標・形状・色やテクスチャ・振る舞いなどに関する情報を、オブジェクトごとに文字列で表現する。表示用の画像データそのものは、ネットワークで送らないので、比較的大規模な3D CGでもVRML形式にすることで転送の負荷が軽減される。

（2） ハイパーリンクの設定

三次元オブジェクトごとにハイパーリンクを設定できる。三次元オブジェクトに設定されたハイパーリンクをたどることにより、より詳細な情報やグラフィックデータを取得できる。

（3） シーングラフによるオブジェクト間の部分全体関係表現

仮想空間の構成要素である各三次元オブジェクトの情報は、ノードという単位で表現される。親子関係は、親子関係にある節点同士をエッジで接続して表現する。このように、各構成要素をノードで、構成要素間をエッジで表現したものをシーングラフと呼ぶ。

（4） ウォークスルー

VRMLで表現された仮想空間は、ユーザが自由に動き回ることができ、三次元オブジェクトを任意の視点から見ることができる。このように、仮想空間中の視点を随時変えながら動き回ることをウォークスルーという。

（5） 詳細度制御

ユーザの振る舞いをモニターし、それに応じたアクションを起こすことができる。この機能の一つとして、オブジェクトの詳細度制御がある。あるオブジェクトに対してあらかじめ詳細度の異なる複数のCGデータを用意しておき、オブジェクトとユーザの距離に応じて、オブジェクトの詳細度を自動的に変更することができる。

（6） アニメーション機能

仮想空間内にアニメーションキャラクタなどを配置し、このキャラクタを動かすことができる。

2.2. データ表現

可視化は抽象的なデータを構造化し、構造化されたデータと可視構造を対応付けすることで実装される。ドキュメントの可視化は通常、ドキュメント中のキーワード等を、ドキュメントを特徴付ける変数として扱う場合が多い。

これらの変数に基づいてドキュメントやその関係が可視化される。可視化された情報は、空間表現される。この空間表現に使用されるデータ型には、以下の７つが考えられる。

線形	テキストやソースコード。
二次元	地図のような領域を持つ要素の集まり。
三次元	実世界の実体のように三次元の広がりを持つ要素の集まり。
時間的	線形の要素の中で特に時系列データに関するもの。
多次元	N 個の属性を持つ関係データベースの組のような要素の集合
木	根となる要素以外は、各要素がただ一つの親要素を持つような集合
ネットワーク	木よりも一般的な、要素間の任意の接続関係を持つ集合

表2-1 データ形

2.2.1. 一次元、二次元、三次元

一次元、二次元、三次元可視構造は、情報を直行した座標に配置することで表現される。この可視構造は、データ構造の次元が低い場合には、非常に有効であるとされている。

日常的に使われる二次元グラフは二次元情報を可視化するものといえる。また、三次元可視化はディスプレイへの投影は二次元であるが、三次元可視化と単なる投影図の相違点は、物体の回転とユーザ視点の移動をアニメーションによって対話的に行えることである。つまり、三次元可視化は、単に二次元投影図を見ているのではなく、必要に応じて視点を対話的に移動できる点が大きく異なるのである。

2.2.2. 多次元

データ構造の次元が多い場合には、一次元、二次元、三次元可視構造と同じに扱うことはできない。このため、多次元データ構造を可視化する技術は様々な種類が開発されている。多次元データ構造可視化には以下のような表現方法がある。

・平行座標
・投影
・幾何学的投影
・アイコンを基にした手法
・階層手法
・グラフを基にした手法
・ピクセル指向手法

平行座標は、複数の次元を平行に並べ、各データを連続したつながりの折れ線で表現したグラフィカル表現である。平行座標では、一度に複数次元のデータ全体の概観を直感的に捕らえることができる。

OLAP等では、平行座標を用いて、多次元データの可視化を行う場合が多い。

2.2.3. 木

木とは、階層的な構造を持つ集合を可視化する時によく用いられる可視構造である。親と根となる要素以外は、各要素がただ一つの親要素を持つような集合である。

木構造は、次項で述べるネットワーク構造と並んで多用されるデータ構造である。ノードを結ぶエッジの方向性、あるいはラベルの有無、平衡木・非平衡木といったトポロジカルな違いなどによってさまざまなバリエーションが存在する。

2.2.4. ネットワーク

ネットワークとは、いくつかの点とそれらの間を結ぶ線からなる可視構造である。木もネットワークの一つの種類である。電話網やWorld Wide Web(WWW)を表現する時によく用いられる。WWW群は、有方向、重み付きネットワークとして表現される場合が多い。

一般に、ネットワーク構造は複雑な対象間の関係の表現のしやすさという点では木構造よりも優れる。しかし、表現者以外のユーザが対象を理解したり、または管理しようとする際、問題が発生しやすい。すなわち、対象の複雑度が増加するにつれ、管理不能になる可能性をもっている可視構造といえる。

2.3. 日本語パーサー

ドキュメントの可視化を行うために、ドキュメントのテキストデータから、キーワードの切り出しなどを行い、テキストデータとは別にキーワード情報をドキュメントに付属する情報として管理することが多い。テキストデータからキーワードを抽出するために用いられるツールが日本語パーサーである。日本語パーサーのツールは以下２つが有名である。

CHASEN[3]
KAKASI[4]

CHASENは奈良先端科学技術大学院大学で開発された、日本語の形態素解析を行う本格的なソフトウェアである。単純にキーワードの切り出しだけを行うのであれば、KAKASIと比べて品質にそれほど差はないように見えるが、CHASENでは、形態素解析を行って得られたワードの品詞情報を抽出できるため、KAKASIよりも細かい日本語処理を行える。一方、両者の処理速度を比べると、KAKASIの方が3～4倍高速である。

日本語以外の言語に関しても、英語、ドイツ語、中国語、韓国語に対応したパーサー、または形態素解析エンジンが開発されている。それらのエンジンの提供形態は、フリー/シェアウェアから製品までとさまざまである。今後、多言語文書が混在した環境下でのドキュメント解析を行う機会が増加すると予想され、それに伴い、まざまな言語の形態素解析エンジンに対するニーズも増加すると予想される。

2.4. ドキュメントベクトル化

ドキュメントの特徴を、多次元ベクトル空間上に表現する手法がある。これは、ドキュメントに含まれる語彙などを座標軸として、語彙の出現頻度をもとに、ドキュメント内容を特徴付けるものである。多次元ベクトルの各次元は単語などに一致し、各次元の値は、単語の頻出頻度に比例する。これらのベクトルの各次元値が近い値の時、ドキュメントの類似性が高いと判断できる。

ベクトルドキュメント管理システムと、現在多く見られる、人手により構築されたドキュメント分類およびキーワード検索が基本的手法であるドキュメント管理システムは、以下の点で相違がある。ユーザ・フレンドリーな検索機能を提供する従来型システムの場合、十分に注意を払って構築された構造と、ドキュメント管理システムとの差はユーザから見て、ほとんど存在しない。むしろ、大きな相違は、システムの構築とメンテナンス性に現れてくる。従来型システムでは、事前に十分に設計されたドキュメントデータベースが要求されることが多く、その構築とメンテナンスには、多くの工数と専門性(DBMSと分類手法の双方)が要求される。

一方、ベクトルドキュメント管理システムでは、ドキュメントデータベースの構造をあらかじめ定義するのではなく、各ドキュメントのベクトル情報に基づいて、ドキュメントデータベースの「構造」をシステムが後天的に見出す。ベクトルドキュメント管理システムの持つこの特徴は、システム開発時、各種資源を大幅に圧縮する可能性を提供できる。さらに、ドキュメントの処理量に応じて、ドキュメントデータベースの「構造」が変化していくという特徴は、システムのメンテナンスにおいても、きわめて強力な機能である。

ジャストシステムの開発した概念検索システムであるConcept Baseは、検索文(日本語)の特徴をベクトル空間上に表現するプリ・プロセッサ部分と、検索対象となるドキュメントの特徴をベクトル空間上に表現する本体部分から構成されている。

2.5. データベース管理システム

近年の、パーソナルコンピュータ(あるいはワークステーションとも呼ばれる)の高性能化、それに比例した高機能化が急速に進展している。またそれを制御するOSも、以前のシングルタスクから、マルチタスク、マルチスレッドが一般的になっている。以前であれば、大型計算機や高性能なワークステーションのみで実行可能であったリレーショナルデータベースが個人利用の計算機で十分実行可能になっている。

大規模なドキュメント情報を管理する場合には、データベース管理システム(DBMS)の存在が不可欠である。データベース管理システムのシェアはリレーショナルデータベース管理システム（RDBMS）が圧倒的であるが、可視化を行うためには、インタラクション時のオブジェクトとデータベースオブジェクトとの親和性からオブジェクト指向データベース管理システム（OODBMS）が用いられることもある。

ある対象をアプリケーション化する際、最も課題となるのは対象を表現するためのデータモデルの構築である。

データモデルとは大量の情報を表現したり、操作したりするための枠組みである。アプリケーションを構築する上で、多少のモデル表現能力の違いがあっても、特定のデータモデルでないと表現できない、ということはない。文献検索やオーサリングの問題を考える際、不完全な構造をもち、情報単位もあらかじめ決めにくいような情報をどのようにして検索するかという課題の方が重要である。

さらに、データベース全体や検索されたデータ集合をいかに効果的に提示するかという技術が重要になる。また、あらかじめよく整理されたデータベースからデータを検索する技術ではなく、整理の悪い、つまり構造化が十分に行われていないデータベースから自分に必要なデータや埋もれている情報構造をいかに探査するかという問題も重要である。このため、情報可視化ツールと連動した、データモデリングツール、あるいはオーサリングツール、そしてそれらの基盤となる理論の開発が望まれる。

2.6. 統計分析

統計分析とは、データから役に立つ情報を正しく読み取るための技術である。ドキュメントからキーワードを取り出した後に、そのまま可視化するのではなくて、統計分析を行うことがある。ここで、使用される分析技法は、分布や平均値・分散を求めるといった単純なものから、多変量解析や時系列解析といった複雑な分析までと多岐にわたる。

統計分析の利用方法は、データの傾向を分析し、仮説を立てることから始まり、その仮説を検証するのに統計分析を用いるというように、仮説立案—検証作業において一貫して利用される。

また、統計分析の結果をグラフィカルな表示手段を用いて表示することは、市販の統計パッケージでは、必ずサポートされる機能の一つとなっている。

統計分析も、計算機の低価格化によって、高機能な統計パッケージがユーザのデスクトップで簡単に使えるようになってきている。現在、有力な市販パッケージとしては、以下のパッケージが挙げられる。

SPSS[5]
SAS[6]

基本的には、統計分析はユーザが立てた仮説を検証するための手段であり、統計分析がユーザに代わって仮説を立てたり、モデルを構築してくれるわけではない。統計分析パッケージが高度化するにつれ、あたかも統計分析が全てを行ってくれるがごとき幻想が生まれてきているのも、現実である。分析の主体者は人間であり、システムはあくまでも補助手段でしかない。

2.7. ニューラルネットワーク

ニューラルネットは生物の神経細胞の仕組みをモデル化し、コンピュータ上でその仕組みの一部を仮想的に実装したソフトウェアである。ニューラルネットは前提知識を持たない状態からスタートし、各ノード間の相互結合の重みづけを変更することにより学習を行う。

ニューラルネットはパターン認識処理に使われることが多く、テキスト中の同一パターンを発見すること等にも用いられる。その重み付けプロセスは他方法に比べ、データの一部欠損に対して強靭なので、特に入力が不完全なパターンを認識することに役立つとされている。

このような機能を実現するためにニューラルネット技術が注目されている。

ニューラルネットワークは、大別して以下の２つのタイプが存在する。

・ある入力と出力の関係を学習するタイプ
・ある状態を表現するタイプ

前者は、パーセプトロン[7]に代表されるタイプであり、後者はホップフィールドネットワーク[7]に代表されるタイプである。

ドキュメントを解析する際にニューラルネットワークを用いる方法として、ドキュメント間、単語間の関連をニューラルネットワークで表現することが考えられている。

また、コホネン(Kohonen)は、二層型の教師なし競合学習の一種である自己組織化マップを開発した[7] [8]。コホネンは自己組織化マップを、ドキュメント集合の可視化に適用している。自己組織化マップは、ドキュメントの特徴をキーワード集合によって表現し、学習アルゴリズムによって類似ドキュメントを近傍に配置する。類似ドキュメントの集まりであるクラスタを特徴づける単語を自動的に選択・表示し、ドキュメントの密度とクラスタの色の濃淡とを対応させている。

3. インターネット上の情報と可視化

3.1. 情報可視化

情報可視化とは、目で見ることができない事象を目に見えるような形に置き換えることである。従来のコンピュータシステムでは、科学技術計算の分野で、シミュレーション結果あるいは属性値の範囲・分布を理解しやすい形で表現する技術を中心に研究されてきた。

情報可視化は、同じ情報でもその提示方法やデータの表現方法よって、我々の認知容易性が異なるということに基づいている。従って、さまざまな種類の情報やタスクに応じて、最適な情報の可視化手法を選択できるように豊富な知見を蓄えることが重視されるようになった。たとえば、ユーザが記憶あるいは処理可能な資源の拡大、検索作業の負担の軽減、特定の情報パターンの検出支援、注意喚起によるモニタ支援、情報の視覚的な直接操作による使いやすいインターフェースの提供、といった目的で可視化手法が考えられている。

この可視化の技術をインターネット上の情報管理への適用という観点で見ると、まず、以下の２つの局面が考えられる。

・インターネット上の情報収集
・インターネット上の情報オーサリング

インターネット上の情報の収集に関しては、これまでも自動巡回ソフトやプリフェッチャーと呼ばれる自動収集用のソフトウェアが開発されている。それらのソフトウェアの大半は、収集目的のURLを再帰的に収集する機能を持っている。

情報収集で問題になるのは、対象となるドキュメントがどこにあるか、また、あるサーバには、どのような分野のドキュメントがあるかなど、ドキュメントのコンテンツとロケーションに関連する問題である。

コンテンツの把握は、ドキュメントの内容解析、すなわちドキュメントオーサリング技術との関連を検討する必要がある。ドキュメントオーサリングは、情報検索や仮説生成・検証と深い関連性を持っている。ゆえに、ドキュメントオーサリングの可視化技術による支援は、情報検索や仮説生成・検証への可視化技術の導入ともいえる。

3.2. 可視化のプロセス

ドキュメントデータなどのオーサリングなどで、一般に可視化を行うシステムは、図3-1のような手続きで行われる。まず、インターネット上などにあるドキュメントデータ（生データ）が収集される。それらから、コンピュータが管理しやすいデータテーブルへと変換が行われ、データテーブルから、可視化しやすいような構造化されたデータ（可視構造）に対応付けが行われる。次に可視構造を画面に表示するための画面変換を行う。

可視化とは、これらの一連のプロセスを支援するための技術である。

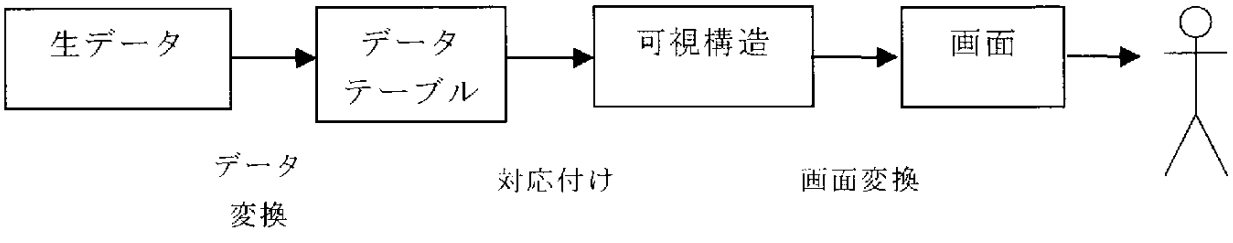

図3-1 可視化の手続き

可視化により、対象はより理解しやすい形で表示される。ユーザは画面に表示された対象を観察し、表示された対象に操作を行うことにより、新たな知見を得る。

可視化技術は、データベースやグラフィック機能の進展に伴って進展してきた。特に、近年の可視化の進展に寄与しているのは、グラフィック関係の機能の進展である。

3.3. インターネット情報の可視化

インターネット情報の可視化という観点からは、情報検索のための可視化に加えて、次の2種類の可視化手法が特に研究されている。

・参照リンクを持つハイパーテキストとしてのWebサイト
・あるURLを起点としたWebページの集合としてのWebサイト

前者の場合には検索条件に合致するWebページの集合を求めるだけでなく、検索結果のWebページと参照・被参照の関係にあるWebページの集合を求めたり、多数のWebページと参照・被参照の関係にあるWebページの重要度を上げて順序付ける方法が知られている。

後者の可視化としては、Webサイト紹介のためのヘルプ機能として提供される、サイトマップのさまざまな手法などが含まれている。このような可視化では、一般にページを節点に、リンクを枝に対応させたグラフ構造の可視化手法と考えられ、焦点・文脈化手法を適用したものが多数存在する。

世界中に数億ページあると予想されるWebページの検索エンジンは、おそらく最も多くのユーザが利用し、かつ最も高い性能が要求される情報検索サービスの一つであると考えられる。しかし残念ながら負荷のかかる可視化手法はほとんど採用されていない。いわゆるポータル・サイトの記事や情報サービスのレイアウトを個人向けにフィルタあるいはアレンジする手法を採用するものが多い。

Webページのメタデータに関しても、以下のようなインターネット特有のものがある。

- Webページの内容を多くのユーザが評価することで、そのページの有用度が推定できる。
- Webページは内容の更新頻度が高いページが多いため、更新履歴や最終更新日時が重要である。

前者は、協調フィルタリングと呼ばれる技術で、ニュースグループ記事の評価をメタデータとして可視化を適用したシステムPHOAKSなどで使われている。後者の例としては、いわゆるプッシュ技術によって更新された内容を記事単位で提供するシステムや、フレッシュアイのような過去1ヶ月以内に更新された情報のみを対象としたインターネット検索エンジンに利用されている。また、インフォメーション・アウトライニングをWebサイトに適用し、最新情報や更新履歴を可視化するサイト・アウトライニングという手法も報告されている。

3.3.1. ブラウジングと検索

情報にアクセスする方法として、ブラウジングと検索がある。ブラウジングとは、大量のデータを対象にして、全体としてどのようなものがあるのかを概略的に見るものである。情報が膨大になると、全体を把握する作業は容易でなくなる。一方、データベースシステムにおける検索とは、検索対象が満たすべき条件を問合せとして表現し、これをデータベースに対して送ることで、検索結果の集合を求める。問合せは、検索したいものが満たすべき条件が明確な場合には適している。多くのデータベースシステ

ムは、与えられた問合せを効率よく処理する方法を自身で選択する機能（クエリー・オプティマイザー）を有している。

可視化によって生成される仮想空間の特性を活かして情報検索を行う場合に考慮すべき事項は、以下のようになる。

（1） 情報の空間配置と動的な空間形成
仮想空間は、その場で変更したり拡張できる。そこで、仮想空間でユーザが行ったいろいろな振る舞いに基づいて、仮想空間を動的に形成し、その空間内に情報を効果的に配置することで、情報検索機能を効果的に支援する。

（2） インタラクションに基づく問合せ形式
インタラクションの結果からユーザの検索意図を推論し、問合せを自動的に生成する。

（3） 属性情報の視覚化制御
仮想空間内をユーザがウォークスルーできる環境で、情報検索を効果的に支援できる属性情報の提示方法の開発。

三次元仮想空間を用いた情報検索でまず重要なのは、情報の可視化である。膨大な情報をコンピュータ画面に提示する方法について、次のような技法が知られている。

（1） ドキュメントの分類階層の三次元アニメーション表示
ドキュメントやファイルは、階層的に分類されていることが多い。階層的な分類構造を三次元仮想空間内で動く三次元の木として表現したシステムとして、ConeTreeがある。このConeTreeをもとに、大規模なカテゴリ階層をもつドキュメントの検索インターフェースシステムとしてCat-a-Coneが開発されている[9]。Cat-a-Coneでは、大量のドキュメント群がどのようなカテゴリで分類されているかを、三次元の動く木によってブラウズできる。

（2） メタファによる検索結果の表示
データベースからの検索結果を、ユーザにとってなじみのある形式で表示する機能も重要である。Cat-a-Coneでもこのことが考慮されており、データベースから検索されたドキュメント群は本のような形式で見ることができる。

3.3.2. Webデータと半構造性

データベースとWebデータベースの世界では、通常、蓄積するデータ群の属性をすべて洗い出し、データベースの情報構造（スキーマ）を作成する。スキーマはデータベースの器であり、これがないとデータベースにデータを入れることができない。このスキーマを前もって定義しにくいデータのことを半構造データと呼ぶ。

現在のWebデータの大半は、HTMLというマークアップ言語により記述されており、関連のある他の記述をハイパーリンクによって参照するハイパーテキストである。データの種類を示すタグをもち、かつ、階層的な構造をもつという意味では、Webデータは情報構造をもっているが、すべてのWebデータが厳密な構造をもっているわけではない。この意味で、Webデータは半構造的といえる。

3.3.3. Webデータの自動分類

Webで公開されている情報の量は圧倒的で、我々が必要とする情報を効率よく検索することが困難である。Webデータを、そのカテゴリや内容によって自動的に分類することは難しい。また、検索エンジンを利用しても問合せの結果は、膨大な量になることがあり、結果として得られたWebページを一つ一つ閲覧していかねばならない。このような問題点を解決するために、従来からWebデータ全体の把握や曖昧検索、分類作業の支援などを目的として、Webデータ全体の集約的な情報を抽出する研究やシステム開発が数多く行われている。

大量のドキュメント群の自動分類等にニューラルネットワーク技術を使用したシステムの研究も行われている。ここでは、ニューラルネットワークの一種である自己組織化マップ（Self Organized Map : SOM）を用いた、Webデータの自動分類について、以下でその概要を述べる。

まず、SOMのアルゴリズムの概要は、以下のようになる。

（1） SOMは、複数のセルが二次元の六角格子状に配置されたものである。

（2） 各セルは、その特徴量をn次元ベクトルで表した形で持っている。このベクトルのことを特徴ベクトルという。

（3） 各セルの特徴ベクトルの各要素は、最初は、適当な値で初期化される。

（4） 用意されたデータによって、SOMにこれを学習させることができる。学習の結果、各セルの特徴ベクトルの値が変化する。

（5） 学習に用いる各データもその特徴を表すn次元の特徴ベクトルを

持っている。

（6） 各データdに対して、そのデータの特徴ベクトルと最も似た特徴ベクトルを持つセルcを見つけ、そのセルの特徴ベクトルの値を、このデータの特徴ベクトルの値に近づくように修正する。さらに、このセルの周囲のセルに対しても、その特徴ベクトルの値を、このデータの特徴ベクトルの値に近づくように修正する。修正する度合いは、セルcやセルcに近いものほど大きくする。

（7） ステップ（6）のベクトルの修正を、すべてのデータを用いて行う。さらに、このプロセスを繰り返す。

（8） 上記の修正プロセスが終わった時点で、各データに対して、そのデータの特徴ベクトルと最も近い特徴ベクトルを見つけ、セルとデータの対応づけを行う。

（9） この結果、特徴のよく似たデータどうしはSOM上で近い位置にあるセルに対応づけられるようになる。

特徴が似ているデータは、SOMの近くのセルに対応づけられることから、データ集合を与えると自動的にデータの分類が行える。さらに、SOMは、セルに対応づけられたデータの位置関係によって、類似しているデータかどうかが直感的に理解しやすいという点から、情報の可視化にも有効である。

SOMを三次元表示可能にした3D - SOMも開発されている[10]。これによって、テキストデータや映像データの効果的な自動分類や類似検索が可能となる。インターネットなどから収集したWebページ群に対して、キーワード付けや内容記述作業を行うかわりに、収集したページの内容を解析し、ページごとにその特徴ベクトルを自動的に生成する。すなわち、検索エンジンの検索結果として得られるWebページ群を、この3D - SOMを用いて自動分類できる。

各Webページから単語抽出を行い、単語の出現頻度情報などを用いて、そのWebページの特徴ベクトルを生成する。すなわち、特徴ベクトルの各要素は、Webページから抽出された単語に対応する。特徴ベクトルの値は、その単語のこのWebページに対する重要度を表している。こうして生成された特徴ベクトルをSOMに学習させて、Webページ群を自動的に分類した分類マップを得る。

3.3.4. ドキュメントの構造発見

CLEVERと呼ばれる検索エンジンが開発されている[11]。CLEVERでは、Web間のハイパーリンクを表現したWebグラフ上の完全二部グラフを求める機能を持つ。完全二部グラフとは、グラフが２つの節点グループに分かれており、節点グループ内には一切枝がなく、片方のグループの節点からもう一方のグループの節点すべてに枝が張られているようなグラフである。

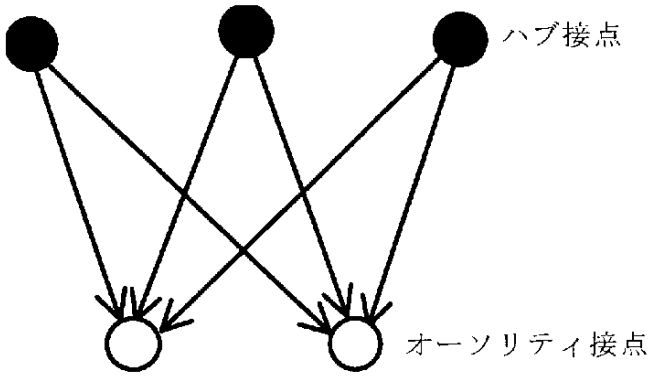

図3-2 完全二部グラフの例

Web全体を大規模なグラフとみなした場合に、その中に存在する完全二部グラフにおいて、枝の出ているグループの各節点をハブと呼び、枝が入ってくる方のグループの各節点はオーソリティと呼ぶ。ここで完全二部グラフは、ハブ節点のグループとオーソリティ節点のグループの間に、そのすべての節点の組み合わせに対してハイパーリンクが張られているようなグラフである。完全二部グラフにより、ハブ節点に対応するWebページは、重要なWebページすべてに対してリンクを張っているページとして認識される。オーソリティ節点は、あるテーマに関して重要で信頼性のある内容をもち、関連するようなすべてのハブ節点からリンクされている。

また、各Webページを節点、リンクを枝とするWebグラフ中で、指定された各キーワードに該当するページがすべて含まれているような部分グラフを検索する方法が提案されている[12]。この方式によると、従来の、各Webページを検索の単位とする検索エンジンで、キーワードのAND検索やOR検索では不可能であった、Webページの固まり（部分グラフ）を発見できる。

3.3.5. 評価

インターネット上の情報可視化は、ドキュメントオーサリングの一部として位置付けられる。しかし、これまでのドキュメントに無い特性として、ドキュメント間のリンクや特定の話題に関して、集中的に情報を蓄積しているWebサイトといった半データベース的サイトの存在などが挙げられる。

これらリンクやサイト情報などをいかに可視化し、表示するかを考える上で課題となるのは以下の2点である。

・情報の空間配置と動的な空間形成
・属性情報の可視化制御

これらの点で、情報の空間配置と動的な空間形成は、対象の大規模化に伴い、より主要な課題となると考えられる。すなわち、現状のシステムで情報の表示方法として使われている方式は、対象の関係を二次元、あるいは三次元上のグラフとして表現している場合が多い。しかし、表示対象が増加するにつれて、これまでの方式では限界がくることは容易に推測できる。

この問題に対応するためには、表示方法に階層的構造を持ちこむ、あるいは色分け表示のような属性情報の可視化制御を行うなど、他の手法との組み合わせが必要である。

また、Web情報の構造の発見に関しては、機械学習的手法が多く用いられているが、その殆どが発展途上のシステムであり、即効性のある解とはなっていない。

4. タスクとユーザ

情報可視化のインターフェースモデル構築にあたり、まず、ドキュメントオーサリングにおける情報構造化の各要因から要求される機能と、ユーザの目的に関する概観を行う。

4.1. ドキュメントオーサリングにおけるタスク

ドキュメントオーサリングという視点で、ユーザニーズを考える時、以下の場合を中心にニーズの分析を行ない、要求される機能との関連を明らかにする必要がある。

・ナレッジマネジメントからのニーズ
・情報ブラウジングにおけるニーズ

ここでのナレッジマネジメントとは組織内部、および外部の情報を収集し、そこから形式知を得て、組織内部に配布し、ユーザからのフィードバックに基づいて知識を洗練化する作業とする。

4.1.1. 情報管理

情報管理は組織の内外にあるデータを収集し、組織内の構造化則に基づいて、それらを構造化し、組織内に知識として提供する作業である。または、すでに蓄積された知識やデータの状態をモニターし、必要に応じて、再編成する作業である。

情報処理の立場から見た場合、データ、情報、知識はそれぞれどのような特徴を持ち、どのように関連し合うのかを明確にしておく必要がある。

データ、情報、知識はそれぞれ以下のように定義できる対象である。

- データはビジネスプロセスの中の様々な点で集められた、生の観測結果の集積である。ユーザや組織にとって、データはそのままの形ではあまり価値を持たず、活用されるためには次に示す情報の形に変換されなくてはならない。
- 情報は様々な決断を行うための、あるいは方針を変更させるためのデータである。情報として認識されるには、ユーザのビジネスに対するメンタルモデルの観点で何らかの意味を持たなくてはならない。また、それに基づいてユーザの決断を変更、あるいは補強するものでなくてはならない。

・知識とは、組織全体の観点から見た時、ビジネスプロセスを変更、あるいは補強させるような情報である。言い方を変えれば、知識は組織がその製品やサービスを提供する時に使っている手続きやテクニックを変更する効果を持たなくてはならない。

利用する対象がデータから知識へと移行するにつれて、価値が増加する。この流れは以下のように進む。

・データ処理
生の観測結果をコンピュータが読めるようなフォーマットで獲得し、その正確さを検証し、関係の無いデータを除外して残りのデータを一貫した構造に再構成する。
・情報管理
データをユーザのビジネス行動に対するメンタルモデルにマッピングできるような有効な表現に変換する。
・知識の構築
動的な情報の中で組織の目的を達成するために、組織の構成要素を再組織化する時に有効なパターンを発見する。

このフローは8つの活動のサイクリックな実行として表現される。データ処理は、データの獲得、変換、統合から始まる。それに続く情報のマネジメントはパッケージ化、発行、決断である。最後に、知識は提供され、コラボレーションが行われる。

これら一連の作業の中で必要とされる機能の内、可視化で支援可能な機能として以下の機能が挙げられる。

・インデックスなどの構造化情報抽出機能
・収集したドキュメントやデータの傾向を概観する機能
・視点を切り替える機能

4.1.2. 仮説生成と検証

ユーザが与えた仮説が成立するかどうかを調べる方法が仮説検証作業であり、従来から統計パッケージなどで採用されている手法を用いることが多い。どのような仮説を与えるかまでをアルゴリズム的に求める方法が仮説生成型であり、人工知能分野で研究されているアルゴリズムを用いる場合が多い。

データから何らかの規則やパターンなどを見つけ出すこと、すなわち、情報マイニングを行う技術は仮説生成にあたって重要な技術の一つである。また、情報として発掘された多数のパターンや規則に内在する構造を見出すこと、すなわち、知識発見を支援する技術も重要な技術である。知識発見プロセスをモデル的に示すと以下のステップになる。

- 知識発見の対象となる領域に特有な性質を理解し、特性に注意しながらデータ収集を行い、データベースもしくはデータウェアハウスを構築する。
- 蓄積されたデータに対して、不要な属性や属性値を取り除くデータクレンジングを行う。
- サンプリング処理などの前処理や、一般化などによるデータ変形操作を行うことによって、知識発見アルゴリズムを効率よく実行できるサイズのデータへと加工する。
- 知識発見アルゴリズムによるデータマイニングを行い、ルールやパターンを求める。
- 求められたルールやパターンに対して解釈や評価を行い、データベースに蓄積されたデータとの照合、そして検証を行う。
- 情報可視化技術なども用いながら、ルールやパターンの評価を行い、最終的な知識表現を求める。

仮説生成と検証作業において、可視化システムに求められる機能は、主に以下の機能である。

- 素早いインタラクションの確立。
- 柔軟な条件変更機能。
- ユーザが持っているメンタルモデルとシステムモデルとの違いを明示する。

4.1.3. 情報検索

ドキュメントオーサリングを行う場合、ユーザの基本的行動は、まず対象とする情報源のコンテンツの状態を確認することである。この行為の基本として、情報検索がある。情報管理、あるいは仮説生成‐検証は何らかの目的に基づいて、情報検索を行い、検索された対象をブラウズ、あるいは吟味する。さらに、その結果に基づき、新たな検索を行う。といった一連の行為と考えられる。

情報検索は、以下の７つのタスクから構成される。

概観	データ集合の全体を把握する。
ズーム	注目すべき要素や部分を拡大する。
フィルタ	興味のない要素や部分を除去する。
オンデマンド詳細化	個別の要素や特定のグループから詳細な情報を得る。
関連付け	要素間の関係を表示する。
履歴	ユーザの操作履歴を管理する
抽出	セッション管理や検索結果の保存を行う。

表4-1 情報検索のタスク

4.2. 可視化におけるユーザ特性

前節では、可視化システムの一つの側面であるシステムに求められる機能（タスク）に関して、その概観を行った。本節では、可視化システムのもう一つの側面である、ユーザに関しての概観を行う。

一口にユーザといっても漠然とした対象であるので、ここでは、企業におけるナレッジマネジメントをその主軸におき、ユーザのタキソノミーを構成する。

4.2.1. 経営者

ナレッジマネジメントにおいて、経営者層が果たすべき役割は、ナレッジマネジメントに取り組む組織や予算などの運用基盤と、報償や人事といった文化的なインフラストラクチャを提供することである。

経営者にとって、知識は組織内にある一つの資産であると同時に、組織のリストラクチャリングを実施する際の原動力となる。

また、経営者は組織内外の知識の概要を、常に把握しておく必要がある。

4.2.2. 情報管理者

組織がナレッジマネジメントを実施する上で、ナレッジマネジメントを担当する役員、および実質的にナレッジマネジメントを推進する担当者が必要となる。このような、いわばナレッジマネジメント専門職の役割は以下のようになる。

- 知識レベルの明確化
  - 現状の分析と確認—課題—レベルアップなど。
- 社内外のナレッジの監視と収集
- 社内ナレッジ発掘方法の立案
- 情報の調査の実施
- 競合他社の動向、顧客の動向、および市場の動向の監視
- 通常とは異なる事象が発生した場合、その分析
- 特に注意を要する事象が発生した場合、社内関係者に通知
- 表彰制度の確立と運用
  - ナレッジマネジメント推進力の整備
- 知識リポジトリの構築とメンテナンス

4.2.3. 一般ユーザ

一般ユーザに対しては、知識を醸造し活用する場を提供する必要がある。この場としては、以下のものが挙げられる。

- 一般ユーザが各自の責任、関心、および業務上の協力関係などを確立する上で必要となる、情報とコンテンツ。
- いつ、何にアクセスするかを従業員がコントロールでき、意志決定を必要な時点で、それを行えるようにする技術と人的プロセス。
- 一般ユーザが、構造化されていない業務を実施する時に、必要な協調と共有を達成するために、従業員のコミュニティ、およびチームに結び付ける手段。このメンテナンスには、以下のような運用基盤からの継続的なサポートが必要となる。
  - コンテンツ、様々なコミュニティとの協力関係、および専門知識に対するユーザのニーズを継続的に分析する。
  - マッピング、索引付け、およびそれらの統合化を通じて、コンテンツとプロセスをワークスペースに組み込む。
  - コンテンツの伝達性と完全性、参加の原動力の維持、および技術とプロセスの先端性など、ナレッジマネジメントをユーザニーズの変化に合わせて存続させる。

4.2.4. 顧客

顧客の知識、ニーズ、カスタマーヒストリーから、自社製品、経営のための知識を抽出する。たとえば、以下のようなシステムが近年注目を集めている。

- カスタマーヒストリー
- SFA

これらのシステムに共通することは、顧客から得られる知識を最大限活用し、ビジネスプロセスの革新につなげたいという願望がある。顧客の知識に基づいた、組織のビジネスプロセスの革新は最終的に顧客にも利便性や利得をもたらす。

顧客との接点が、このような梃入れ効果を持つためには、以下のいくつかの要因が前提となる。

- 顧客がビジネストランザクションを制御し、カスタマイズできる。

・顧客とコンタクトする時、組織が適用できる資源がある。この資源は、それまでのカスタマーヒストリーの蓄積から構成される。
・ビジネストランザクションの質が保証されている。ビジネストランザクションの質は以下のような要因に依存する。
  - 顧客の要求は予想されている。
  - 顧客の要求は簡潔に記述されている。
  - 関連する顧客の要求とそれに対する解が示唆されている。
  - いろいろな要求に対する反応は素早くカスタマイズできる。
  - 製品(サービス)は素早くデリバリされる。
  - 製品が最初の宣伝手段となる。
・顧客が困難と感じる点は素早く、そして顧客に苦労させることなく解決できる。
・顧客が金銭的問題を感じること無く、ほとんど顧客の目から金銭的問題が見えない。
・製品のライフサイクルは、顧客の立場で管理される。

4.2.5. ユーザとタスクの関係

前節では、ナレッジマネジメントの中で想定されるユーザのタキソノミー、その特性に関する議論を行った。本節では、そのユーザと、ユーザが求める機能を明らかにする。

(1) 経営者

ナレッジマネジメントにおいて、ユーザとしての経営者は組織内外の知識の概要を、常に把握しておく必要がある。ここで、可視化システムに求められる機能として以下のような機能がある。

- 組織内外の知識を概観できる機能。
- 注目を払うべき事象が発生した場合、それをユーザに迅速に知らせる機能。
- 経営者の問題点と、対象とを結び付ける機能。

(2) 情報管理者

社内で主にナレッジマネジメントを担当する専門職には、監視、収集、処理の対象となる情報の量を拡大するツールが提供されるべきである。それらのツールを用いることにより、ナレッジマネジメント専門職は収集した情報を要約、分析して従業員に提供する。また、ナレッジマネジメント専門職が、知識の獲得と利用に関するモニタリングを行うためには、様々な技術が必要となる。その中で特に重点をおくべき機能として、以下の大規模な情報源を監視する機能が挙げられる。

- 公式および非公式の情報を大量に獲得する機能
- パターンを検出し、大量の情報の意味を把握する機能
- 情報源を系統的に編成する機能
- 必要な際、情報にアクセスできるように保存する機能
- 保存された情報を使用してレポートを作成する機能
- 裏付け情報をユーザの求めに応じて利用できるようにする機能
- 必要に応じて、アクセスできる形でレポートを保管する機能

(3) 一般ユーザ

情報管理者が情報管理を行うように、一般ユーザも情報管理を行う。しかし、その範囲はユーザ個人の業務、あるいは興味の範囲に限定される。

また、検索に関しても情報管理者が行う検索よりも、不定形である場合が多い。

可視化に対する一般ユーザからの機能要求として、以下のような情報管理者と同じ機能要件が挙げられる。

・情報源を系統的に編成する機能
・必要な際、情報にアクセスできるように保存する機能
・保存された情報を使用してレポートを作成する機能
・裏付け情報をユーザの求めに応じて利用できるようにする機能
・必要に応じて、アクセスできる形でレポートを保管する機能

(4) 顧客

ナレッジマネジメントという枠組みから見ると、顧客は情報あるいは知識源の一要素である。可視化の役割は、それによってもたらされるWebサイトのイメージアップ、顧客によるビジネストランザクションのカスタマイズのしやすさなどが考えられる。

しかし、先に述べた企業と顧客との関係から考えるならば、これらの機能は二義的なものであり、顧客と可視化の関係は婉曲な関係に留まっている。

5. 可視化技術とツール

5.1. 可視化技術のタキソノミー

情報可視化技術を考える際のタキソノミーとして、以下のような分類が考えられる。

以下に示した項目の内、(1)と(2)に関する概観は3章で、(3)と関連するサーベイを4.1節で行った。本章ではそれ以外の項目に関して、研究開発が行われている技法やツールに関して概観を行う。

なお、以下の各項目の後ろに示した番号は、各項目に関連する技術を述べた節、または項である。

(1) 基礎理論
(2) 空間表現
(3) 焦点・文脈化【5.2】
(4) ユーザ・システムインタラクション【5.3】
(5) ツール、オブジェクト【5.2、5.4】
(6) アプリケーションとインプリケーション【5.4】
(7) 情報可視化に向けたデータマイニング【5.5】
(8) ユーザの作業支援ツールとしての可視化【5.6、5.7】

5.2. 焦点・文脈化

テキストを二次元の図に変換するということは、以下の２つの作業から構成される。

・ページをタイル化する
・合成ページを生成する

たとえば、タイル化された略記は、ドキュメントの集合を表記するために一般的に使われている。合成ページの例として、ハイパーテキストの集合から新聞の見開きのような画面を生成する情報検索の技術がある。

文献データのオーサリング支援という観点で考えると、主要な技法は、焦点・文脈化技法である。

ユーザが膨大な情報を取り扱う時に、いかにして情報の中からユーザの見たいところだけを見せるかという問題がある。これに対する一般的な解は画面スクロールである。画面スクロールとは、上下あるいは左右に移動可能な仮想的ウィンドウを通して情報の一部を見る手法で、エディタや図形システム等で一般的に利用されている。

画面スクロールの問題点は、表示される情報が局所的であり、全体の構造がわかり難くなりやすい点である。たとえば、図5-1のような大きな木構造のレイアウトを考えると、木の先端部では着目ノードとその子供は見えるが、それ以外では着目ノードしか見ることができない。

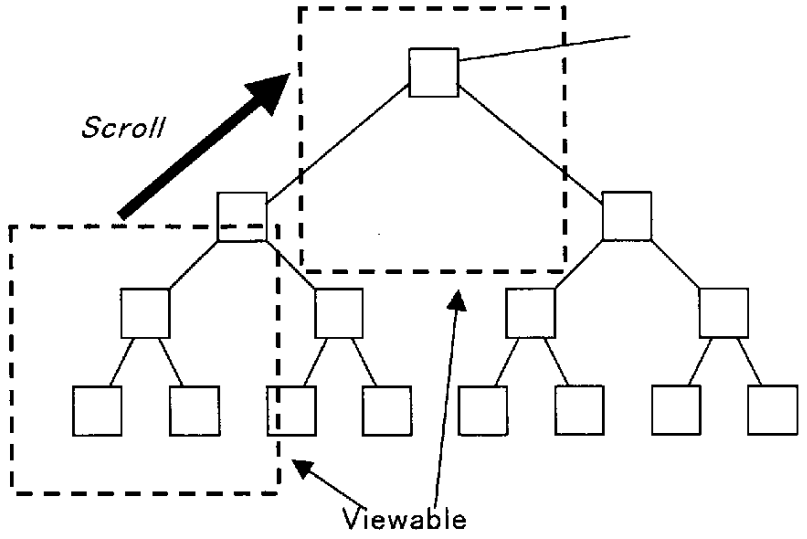

図5-1 スクロールの問題点

この例のように、可視化する情報が画面に収まりきらない時、着目点(焦点) の付近は詳細に見ながら、全体の構造(文脈)も同時に見たいという要求がある。この要求を実現する技術を焦点・文脈化と呼ぶ。

焦点・文脈化は、ブラウジングやナビゲーションのための大域的な情報を保持しつつ、興味のある情報を必要な詳細度で参照するために有効であるとされている。また、焦点・文脈化は三次元可視化では透視投影図法によって自然にこの効果が得られる。ただし、三次元可視化よる焦点・文脈化の実現は、三次元空間における物体の物理的配置に依存するため、配置を変えずに他の焦点・文脈化を得ることが難しい。

これに対し、三次元可視化に限らず一般的に利用できる焦点・文脈化手法がこれまでにいくつか提案されている。

5.2.1. 情報のズーミング

可視化された情報は、ユーザの理解を助ける手がかりを与えてくれる。我々が複雑な対象を理解する時は、対象を理解可能な単位に段階的に分割し、分割した対象を相互に関連付け、分類し、抽象化するといった作業を繰り返す。

我々は、よくわからないものを見る時、目を寄せて、より詳細な情報を得ようとする。これはズーミングといわれる作業である。膨大な量や複雑な構造を持つ情報を可視化する際に問題となるのが、ディスプレイの大きさから受ける制約である。情報を可視化するソフトウェアは、この制約を考慮しながら、かつ必要十分な情報を読み手に与えなければならない。

情報ズーミング技法は以下の二種類に大別できる。第一の技法は、情報の可視データを拡大表示する物理的なズーミングであり、第二の技法は、情報の意味や構造を詳細化して表示する論理的なズーミングである。いずれの技法も、情報の焦点・文脈化を行うための技法である。

（1） 物理的なズーミング

魚眼レンズ法[13]は興味の対象となる情報だけを拡大し、他の不要な部分を極小化して表示する技法である。ユーザがレンズを動かすことによって、拡大表示される対象も変化する。情報を拡大表示した時にも、拡大表示された対象と他との関係が示されているため、情報の間の関係をユーザが見失うことはないとされている。

（2） 論理的なズーミング

情報が階層構造を持っている場合、概要から詳細へと画面を変えながら情報を表示するインターフェースが使われることがある。このような場合、画面の切り替えによってユーザの思考が中断されるだけでなく、ユーザは、概要情報と詳細情報の間の関係を、ディスプレイ上の位置関係と共に常に把握していなければならない。論理的なズーミングでは、ユーザの興味の対象だけが詳細化され、他の情報は極小化して表示され、情報の間の関係と部分的な詳細情報とを同時に参照する工夫がなされており、概要から詳細への視点を移動する際に、ユーザの思考が中断されることはない。

5.2.2. ドキュメントレンズ

ドキュメントレンズ[13]は、物理的なズーミングの一つであり、ドキュメントをページ単位で概観するために、二次元平面上に配置したページを三

次元的な遠近法で投影するものである。ユーザが興味をもっているページを読むためのツールとしてドキュメントレンズが使用される。ユーザは各ページとそれに隣接するページを俯瞰することで、大域的な特徴をもとに興味のある箇所を特定することができる。また、キーワード検索の結果、指定したキーワードがどのようなページに分布しているのかを確認できる。ドキュメントレンズをおいたページは画面の中央におかれ、文字が読めるサイズに拡大される

また、魚眼レンズのように中央ページの周辺は歪められ、注目しているページを邪魔しない程度の文脈を提供する。ドキュメントレンズでは、このような手法で焦点・文脈化を実現している。

焦点・文脈化技術を用いて、リンク構造や木構造の可視化を行うWebBook[14]というシステムが開発されている。WebBookも魚眼レンズを採用している。図5-2に、WebBookの出力例を示す。

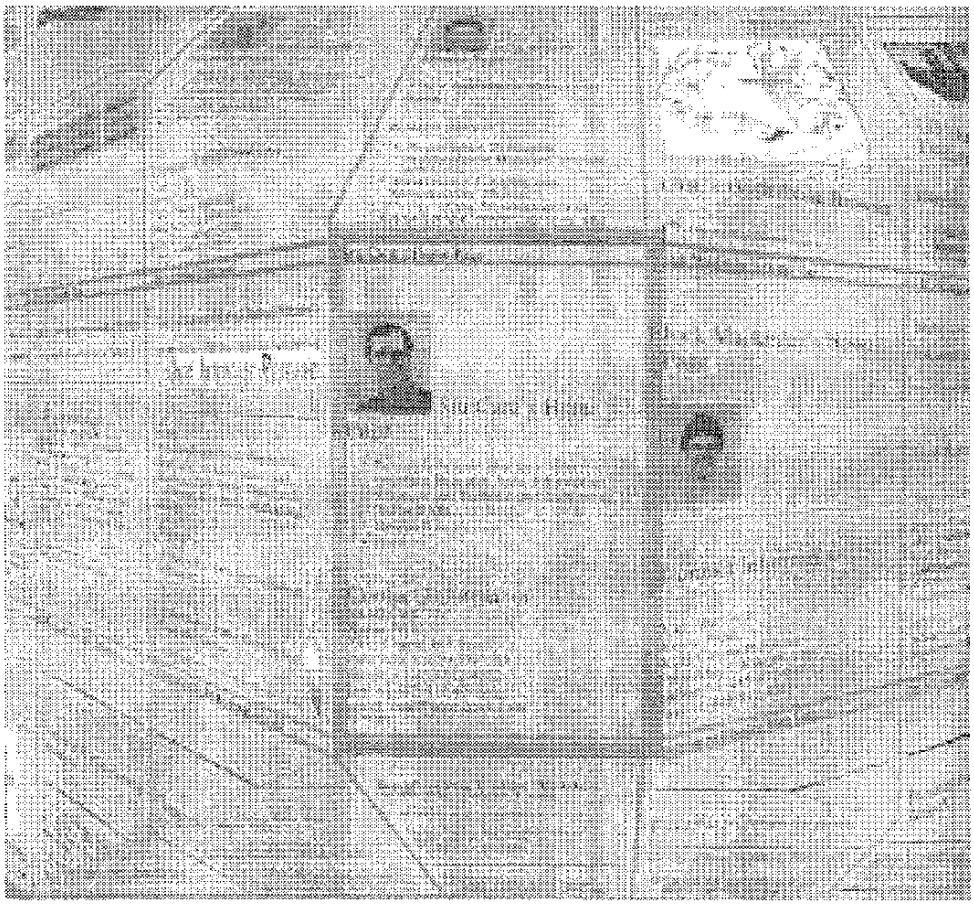

図5-2 WebBookの画面例

(1) Magic Lens

ユーザレンズをより柔軟に制御でき、またレンズによって注目している部分に情報を表示できるようにしたMagic lens[14]というツールが開発されている。図5-3に、Magic lensの出力例を示す。

WebBookの焦点化の粒度が“離散的”であるのに対して、Magic lensの焦点化の粒度は“連続的”であるといえる。Magic lensで使用されている焦点化技法は、グラフィック作成ツールで採用されている“拡大鏡”と、ほぼ同じインターフェースを提供する。

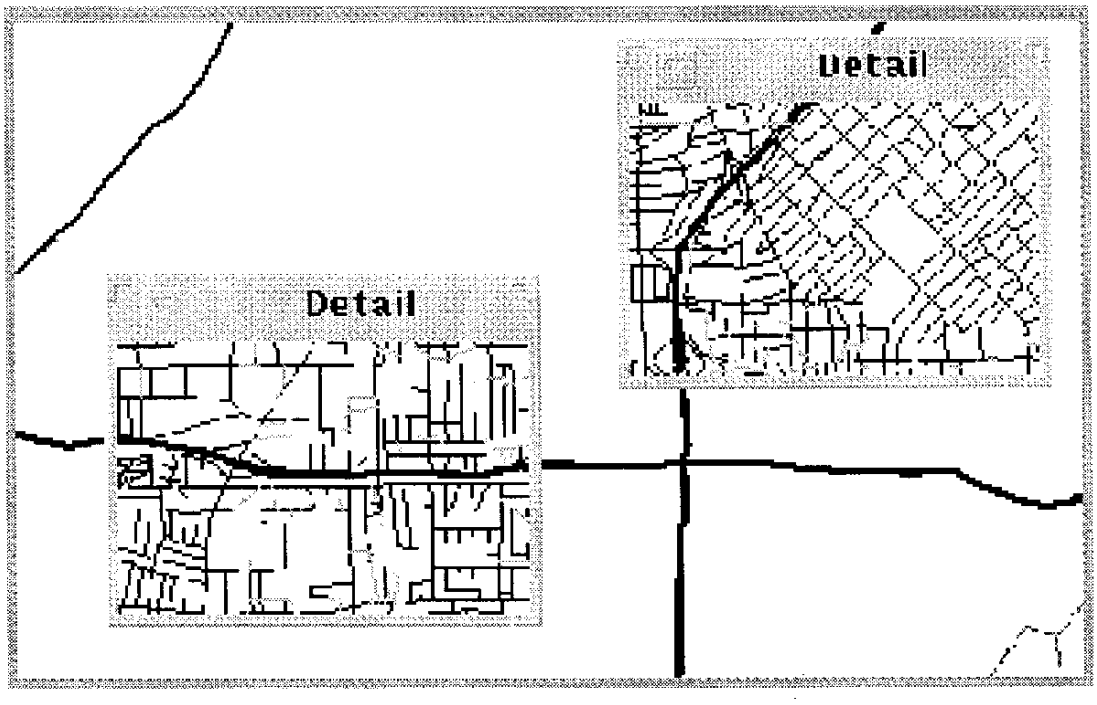

図5-3 Magic lens出力

(2) 一般化魚眼ビュー

ファーナス(G.W.Furnas)の提案した一般化魚眼ビュー(Generalized Fisheye Views)は、木として表現される情報構造に適用できる焦点・文脈化手法である。図5-4のように、本手法は木の潜在的重要度(API)と焦点からの距離(D)を用いた以下の式に基づき木の各ノードの重要度(DOI)を決定する。

$$DOI = API - D$$

ただし、潜在的重要度はルートを0とし、ルートから離れるにしたがい、−1, −2,... と単調に減少するものと定義される。

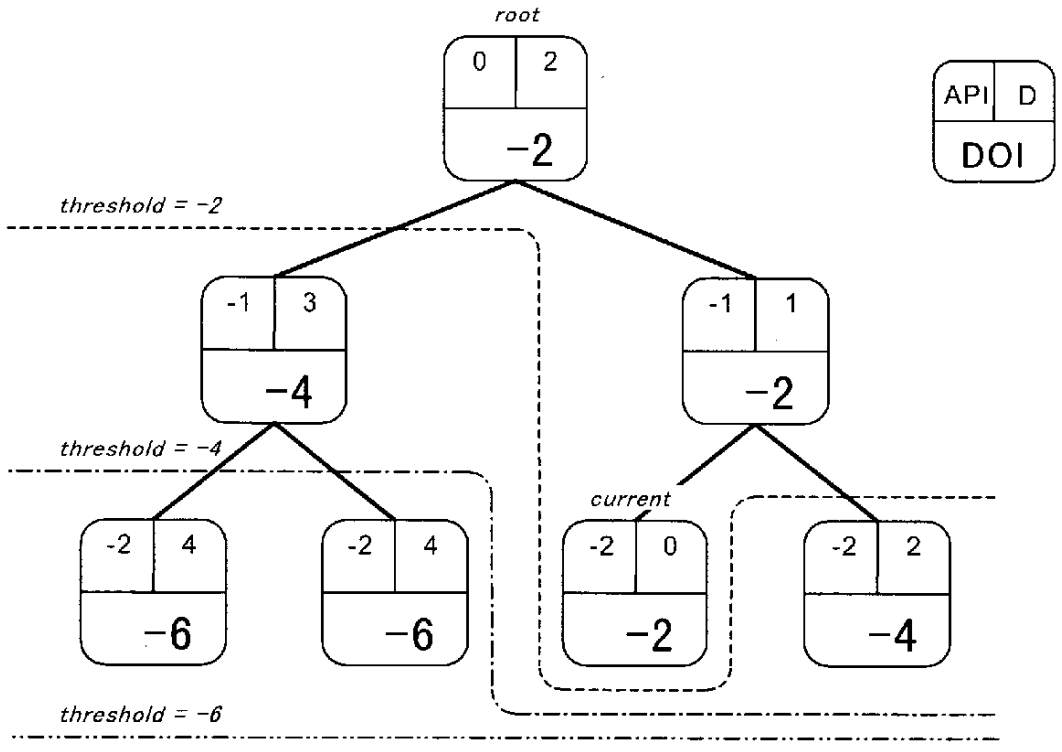

図5-4 各ノードの計算例

各ノードの重要度を計算後、閾値を設定し、閾値未満のノードは消去することにすると、着目点近傍の詳細と全体の文脈がミックスされた表示が得られる。また、閾値を増減することで、表示の範囲を拡大縮小できる。

(3) 一般化魚眼ビューの応用例

ファーナスは、構造化エディタ、スケジュール表、SemNetでの情報表示量削減に一般化魚眼ビューを適用した。構造化エディタへの適用では、インデントされたCプログラムを木構造とみなし、魚眼ビューを適用している。普通のテキストエディタでの表示と異なり、現在着目している行の前後数行と、着目行からの距離は遠いがプログラムの制御構造を知る上で重要な行（たとえばfor文、if文、switch文等）の同時表示を実現した。

またSemNetでは、CAD(Computer Aided Design)のオクトツリーの概

念を利用し、これに魚眼ビューを適用することで情報削減を行った。つまり、表示空間である三次元立方空間を8分割し、さらにそれら部分立方空間を8分割した8分木を作る。そして、各ノードをそれが所属する部分空間の子として登録する。ユーザの着目している部分立方空間を焦点として魚眼ビューを適用している。論理的に距離の遠い部分空間は空間の存在を示すマーカーだけを表示し、ノードは消去している。ただし、この手法はユーザの視点が部分空間を移動する時の表示の変化が激しく、ユーザに負担をかけることが指摘されている。

サルカー(N.Sarkar)とブラウン(M.H.Brown)は魚眼ビュー[15]をグラフ描画に応用した。彼らのシステムは、アメリカの主要都市をノードとするグラフで、着目都市とその近傍都市を大きく表示し、遠方の都市を小さく表示している。また、重要な都市にはあらかじめ重みを与えておくことで、小さくなりすぎることを回避している。一般化魚眼ビュー自身はノードの重要度を計算するだけで、これらがグラフ描画に応用された場合のレイアウト問題は扱っていない。これに対し、サルカーとブラウンは魚眼ビューをレイアウトに利用するための一般的枠組みに関する議論を行っている。

この他に、三末と杉山はKJ法エディタ[16]で用いられる網図と領域図が共存する複合グラフに魚眼ビューを適用した。ノイック(E.G.Noik)も同様の研究を行っている[17]。また、フェアチャイルド(Fairchild)らはConeTree[18]で、木の各ノードに対し抽象度の異なる数種類のアイコンを用意し、ノードの重要度に従い表示アイコンを変化させるシステムを提案している。

(4) フラクタルビュー

一般化魚眼ビューでは、簡単なアルゴリズムで効果的な焦点・文脈化表示を実現できる。しかし、表示される量が対象とする木の分岐数に依存するといった問題点がある。たとえば、同じ閾値を設定したとしても、着目点の近くの分岐数が少ない場合には少数の情報しか表示されない。一方、分岐数が多い場合には多量の情報が表示される。

これに対し、フラクタルの持つ抽象化機能に着目した情報量制御手法が提案され、フラクタルビューと呼ばれている。フラクタルビューは木として実現できる情報構造に適用可能で、次頁の式によって焦点と各ノードの類似性を計算し、その類似性を重要度と見なす。

$$Fv_{focus} = 1$$

$$Fv_{children_of_x} = r_x \cdot Fv_x$$

$$r_x = C \cdot N_x^{-\frac{1}{D}}$$

Fv:ノードxの重要度（類似性）
C:定数（０～１）
Nx:ノードxにおける枝の数
D:焦点からの距離

ただし、この式に従って各ノードに値を伝播すると、閾値に対して分岐数の少ない場合はより先まで、分岐数の多い場合には近くだけを表示することが可能となる。さらに、離散的な情報構造を連続系として理想化した状態では、この表示量が分岐数に関係なく同程度となる。

(5) フラクタルビューの応用例

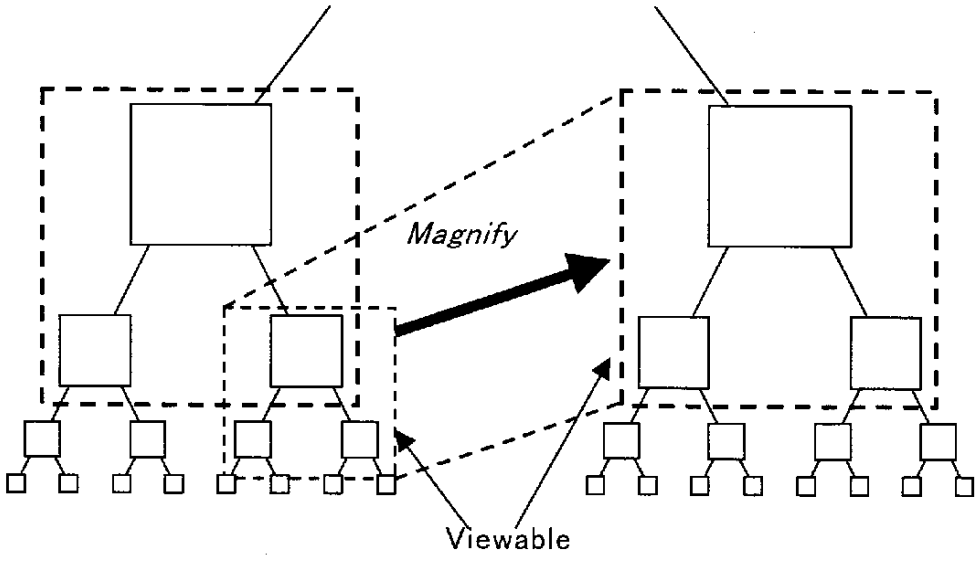

図5-5 フラクタルの自己相似性を利用した木のレイアウト

UNIXディレクトリに代表される大規模な階層構造を可視化するシステムの潜在的有用性は高い。しかし、これまでに提案されている木レイアウトの枠組みで、大規模な階層構造を可視化することは難しい。すなわち、木の高さが深くなるに従って末端の幅が指数関数的に増大し、画面から溢れてしまうからである。この時、画面スクロールはあまり意味をなさない。ノード同士の間隔の狭い葉の部分ではいくつかのノードを見ることがで

きるが、木の上部では親子、兄弟ノードさえ見ることができなくなるからである。

小池[19]らは、これらの問題点に対しフラクタルビューを適用し、その解決を図った。第一の改善点は、木の枝の長さとノードの大きさを上述の式に基づき縮小するフラクタルレイアウトである。図5-5に示すように、自己相似性を利用することにより、木のどのノードに着目していても統計的に同じような視点が得られ、常に着目ノードとその周辺を見ることができる。

第二の改善点は、着目点を中心としたフラクタルビューによる表示ノード数の制御である。この結果、表示されるノード数が削減されるだけでなく、ほぼ同程度の量のノードが表示されるため、システムの反応速度が同程度に保たれる。図5-6に、小池らが試作したシステムの画面例[19]を示す。どのレベルにおいてもほぼ似たようなディレクトリ表示が得られている。また、図5-6の画面右側にあるGUIスライダを用いて、提示情報量の制御もできる。

一般のテキストエディタでプログラムを作成する場合、着目点のある行とそれに連続する上下数行を見ることができる。しかしこの時、プログラムの制御構造に関する、より大局的な視点は失われている。

小池らは、フラクタルビューを応用したエディタを開発した。このエディタは、通常viエディタとして機能するが、モード変換キーを押すことにより、Fractal View(ON/OFF)モードに移行する。つまり、着目行から各行の重要度を計算し、ある閾値以下の重要度をもつ行は消去する。このモードでは着目点周辺の情報は連続して表示されるが、着目行から離れるに従い、switch、ifといった制御構造を示す行だけが表示され他は消去される。

この表示法の欠点は、着目行を移動した時の表示変化が激しいことである。特に、現在表示されていない行へ移動した時のユーザに与える認知負荷が大きいことが実験の結果、わかってきた。この問題を解決するために作成されたのが、図5-7に示すマルチスケーラブルフォントモードである。各行はその重要度に対応する大きさのフォントで表示される。基本的に省略される行がないので、連続的変化を認識しやすい。実際にはこれらのモードはモードキーで変更できる。

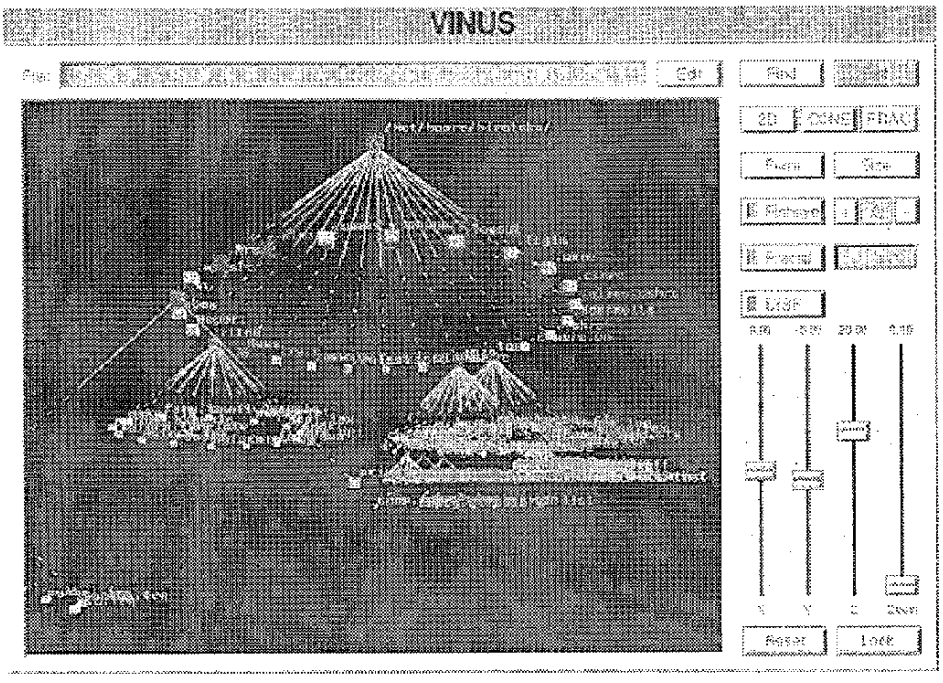

図5-6 フラクタルビュー

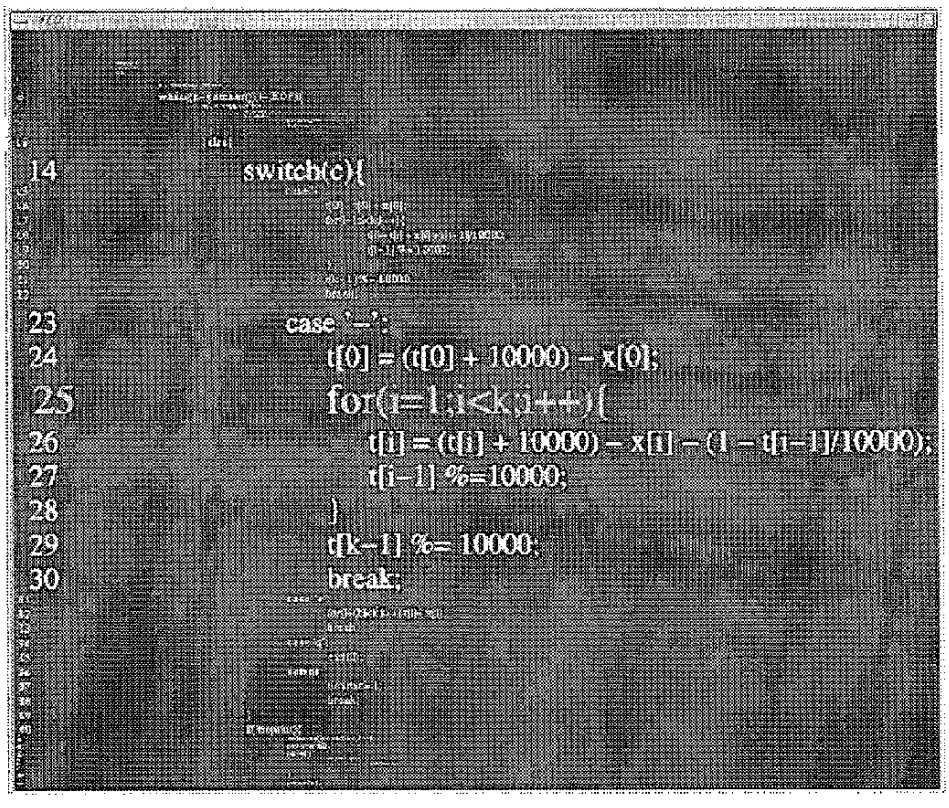

図5-7 フラクタルビューによるCプログラムの表示

5.2.3. その他の焦点・文脈化システム

ペルリン(K.Perlin)らは焦点・文脈化技術を利用したシステムPad[20]を開発した。Padは仮想的な無限大の描画領域を持つ。その描画領域上にテキストやグラフィックが異なるスケールで描画される。ズームアウトした位置から見ると、上位レベルの情報記述が見え、ズームインすると、個々の情報の詳細を見ることができる。

ペルリンらはPadを従来のWIMP (Window, Icon, Menu, Pointer)に代わる新たなインターフェースとして位置付けている。WIMPのクリック・アンド・オープンパラダイムでは、たとえば、フォルダを開いて見なければその内部情報はわからない。これに対しPadでは、インタラクティブにズームすることで徐々に詳細が見えることにより、クリック・アンド・オープンパラダイムの問題点を解消している。

Xerox PARCではPerspective Wallの知見を活かし、その後、Document Lens[13]、Spiral Calendar[9]など焦点・文脈化システムが開発された。また、階層構造を双曲平面上に描画するシステム[9]も開発されている。

リバーマン(Lieberman)の開発した焦点・文脈化システムPowers of Ten Thousand[21]は、半透明表示を利用して、全体図と詳細図を重ねて表示するシステムである。ただし、その表示は非常にみづらいとされている。

5.2.4. 評価

焦点・文脈化システムの利点は、ズーミング機能によってユーザの注目点を、それがおかれている文脈情報とともに表示できる点にある。これにより、ユーザは詳細に見たい部分と、それを取り巻く文脈の双方を、思考の中断無しに参照することができる。

ただし、現在のシステムでこの利点が発揮されるのは、ファイルシステムやソースコード、地理的情報など、詳細と全体表示に関して統一的なバックグラウンドがある場合が多い。

このバックグラウンドのモデルが一貫性を持たない、あるいは文脈となる部分の表示アナロジーが認知的不協和をもたらすような場合、焦点・文脈化はユーザーインターフェースを阻害する要因となってしまう。

5.3. インタラクション

コンピュータでデータを取り扱うことの長所の一つとして、インタラクションが考えられる。可視化は、図5-8に示すように、生データから可視構造へ対応付けを行い、次に可視構造化されたデータを画面に表示することにより行われる。

ユーザが画面を操作し、操作結果をデータテーブル、あるいは可視構造にフィードバックし、再び、画面表示に反映させることをインタラクションと呼ぶ。ここでの操作とは、スライドバーをスライドさせることなどである。

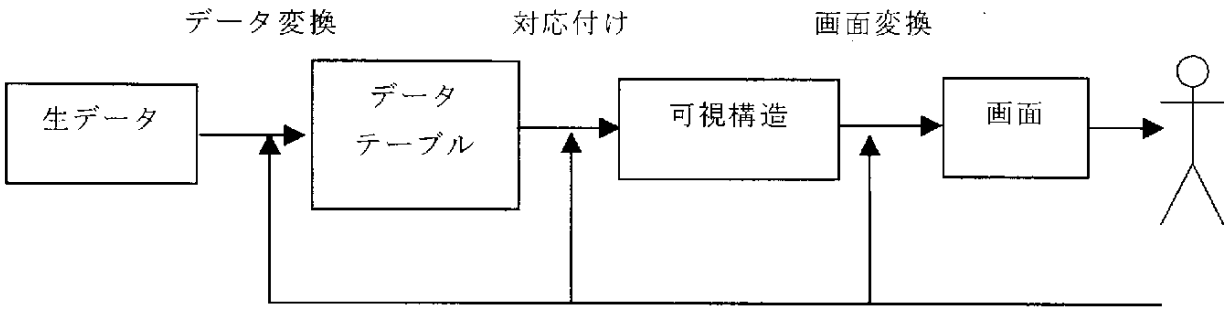

図5-8 相互作用

5.3.1. 動的質問

動的質問とは、画面の操作をデータベースへの問合せに変換することである。可視化の対象となるデータは階層的になっている場合が多い。動的質問は、生データに近い階層に対して問合せを行う手法である。

たとえば、動的質問はユーザがスライドバーを利用してデータベースの検索条件を連続的に変更し、その結果、得られる検索結果表示の動的な変化を元にして、最適な解の集合に到達するという仕組みになっている。比較的データ集合の静的な観点の可視化に比べて、動的質問では、アニメーション効果とでも呼ぶべき連続的な変化を可視化の重点としていることに特色がある。ここでは、概観を制御するとともに、不必要な情報を除去するフィルタの役割をスライドバーという可視化要素によって実現している。また地図などを背景として、検索結果を輝点で表示すると視認性もよく、色別の情報分類も可能となり、有効な可視化を実現できる。

このような動的質問の可視化手法はOLAPやデータウェアハウスの分析

手法とむすびついて、より強力な対話的データ分析機能を実現するものと考えられる。大規模なデータベースに対して高速に動的質問を処理することは困難であるかもしれないが、検索結果がある程度絞り込まれている場合には、有効な手段となる。このことは、対象データのサイズや可視化手法のスケーラビリティに応じて、複数の可視化戦略を組み合わせて最適なインターフェースを構築するという要求につながる。

テキスト情報に対する動的質問の例としては、映画のタイトルや俳優などで検索を行うFilm Finder[14]という検索システム（図5-9参照）が試作されている。

5.3.2. 動的分析

動的分析は画面の操作により、データを分析し、分析結果を画面に表示する手法である。動的分析は、データベースに問合せを行うのではなく、ユーザーインターフェースを提供するアプリケーション上で行われる。具体的には、対象とするデータは変更せず、見え方を変更するような作業を行う。たとえば、ユーザの目的によってドキュメントの類似性にスポットを当てて可視化を行うことや、ドキュメントの時間的データにスポットを当てて可視化を行うことなどが考えられる。

5.3.3. 評価

あるインタラクションが、あるユーザにとって本当に便利な機構であるか否かを決定する要因として以下の各点が考えられる。

・システムの反応の良さ。
・間違った指示を与えた時の回復機能。
・ユーザが与えた指示を蓄えるヒストリー機能。
・表示された情報と、実際の対象との対応性の良さ。

これらの要因が相互に補完し合うことにより、良好なインタラクションが実現される。

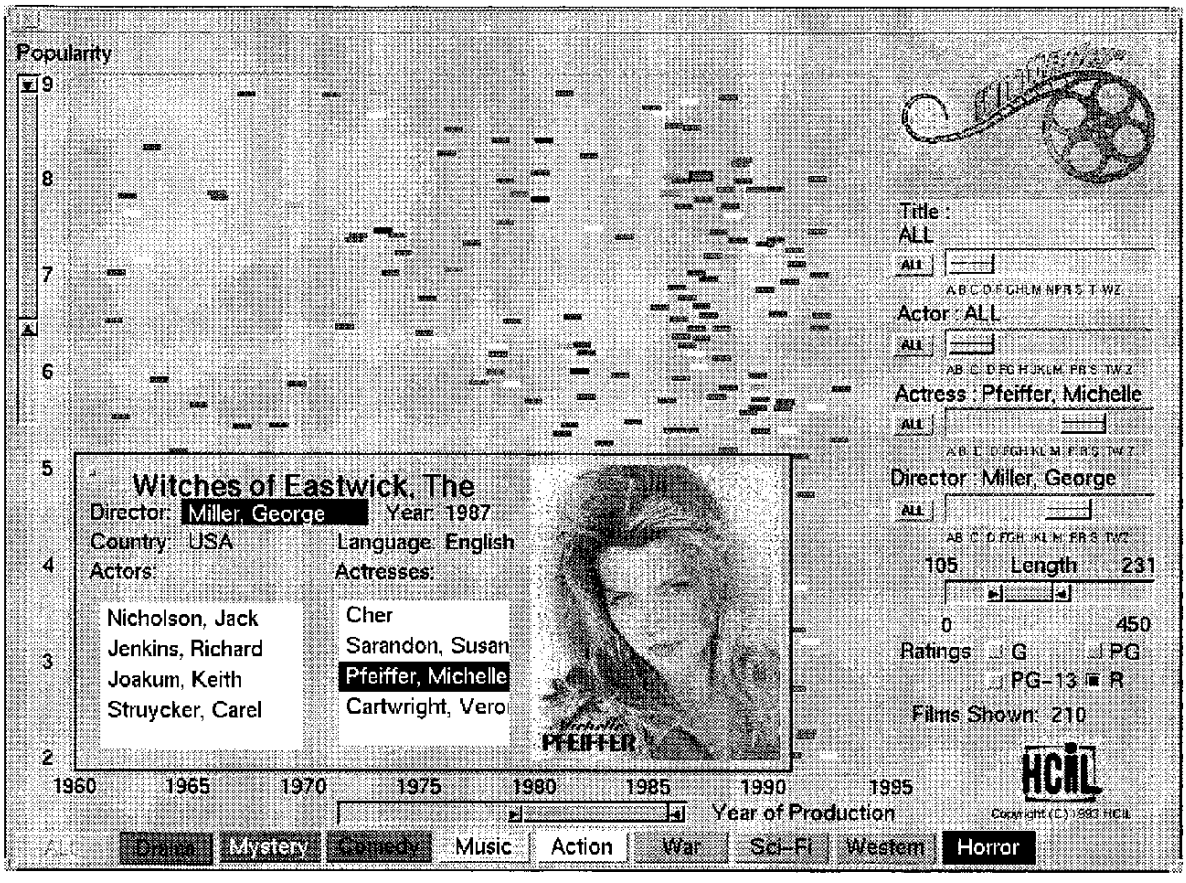

図5-9 Film Finder

5.4. ドキュメント可視化

5.4.1. サルトンのシステム

サルトン(Salton)は、ドキュメント内容をドキュメントに含まれる単語によって特徴付ける手法、すなわちドキュメントのベクトル化表現を提案した。さらに、ベクトル化されたドキュメントをデータベース化することにより、ドキュメントが表現している、内容やセマンティックスに基づいた検索を提供することを提案している。[22]

たとえば、ある本の各章の内容を単語ベクトルで表現し、各章ごとの類似度をベクトル空間分析で計算する。サルトンのシステムでは、ドキュメントの各章を円上の点としてマッピングし、類似度の高い章の間に線を引くことで、章ごとの関連を可視化している。サルトンの開発した手法のうちベクトル空間分析はその後のドキュメント可視化に大きな影響を与えている。

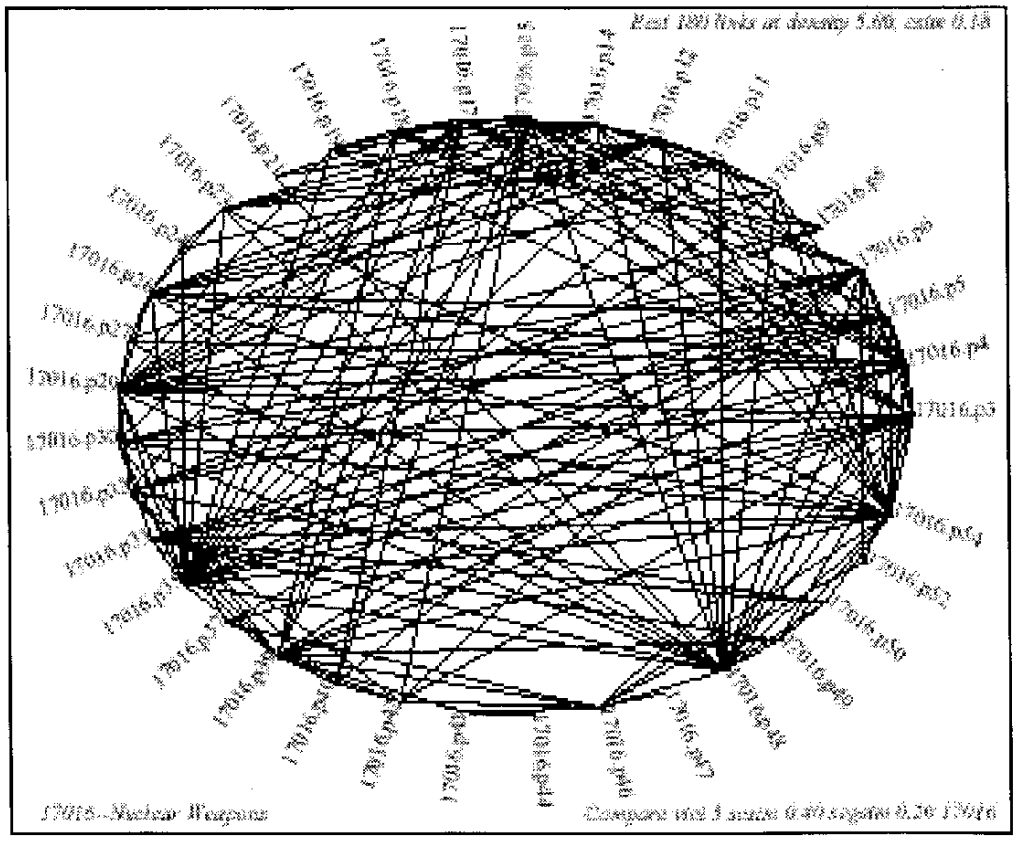

図5-10 サルトンのシステム

5.4.2. リンのシステム

巨大なドキュメント集合を行うために、関連するドキュメント同士を空間的に近い位置に配置することが考えられる。リン(Lin)は、ドキュメント集合をこのルールに従い二次元空間に配置するシステムを開発した[23]。このシステムの目的は、ドキュメント集合の各ドキュメント間の相互関係をユーザに理解させることであり、自己組織化マップを用いてドキュメント集合の可視化を行った。リンのシステムの特徴は、以下の通りである。

- ドキュメントの集合を二次元格子に配置する。
- 内容の近いドキュメントは、格子上の近くの点に表示される。
- ドキュメントはデータテーブル状で、ベクトルとして表現される。
- ベクトルの中身はドキュメントのキーワードである。
- 二次元平面上にマップする前に、これらのベクトルを自己学習させる。
- 図を領域に分割するために境界が描かれ、領域は名前付けされる。

リンの採用した学習アルゴリズムは自己組織化マップに基づくアルゴリズムであった、オリジナルのアルゴリズムには時間がかかるという問題があった。リンは学習を分割すること、およびスーパーコンピュータを用いることで、この問題を回避した。リンは分割した領域に対して名前付けを行っているが、このラベリングは類似性と相違性を複雑にする可能性がある。このラベリングを行う良いアルゴリズムは現在のところ提案されていない。

図5-11にリンのシステムの学習例を示す。

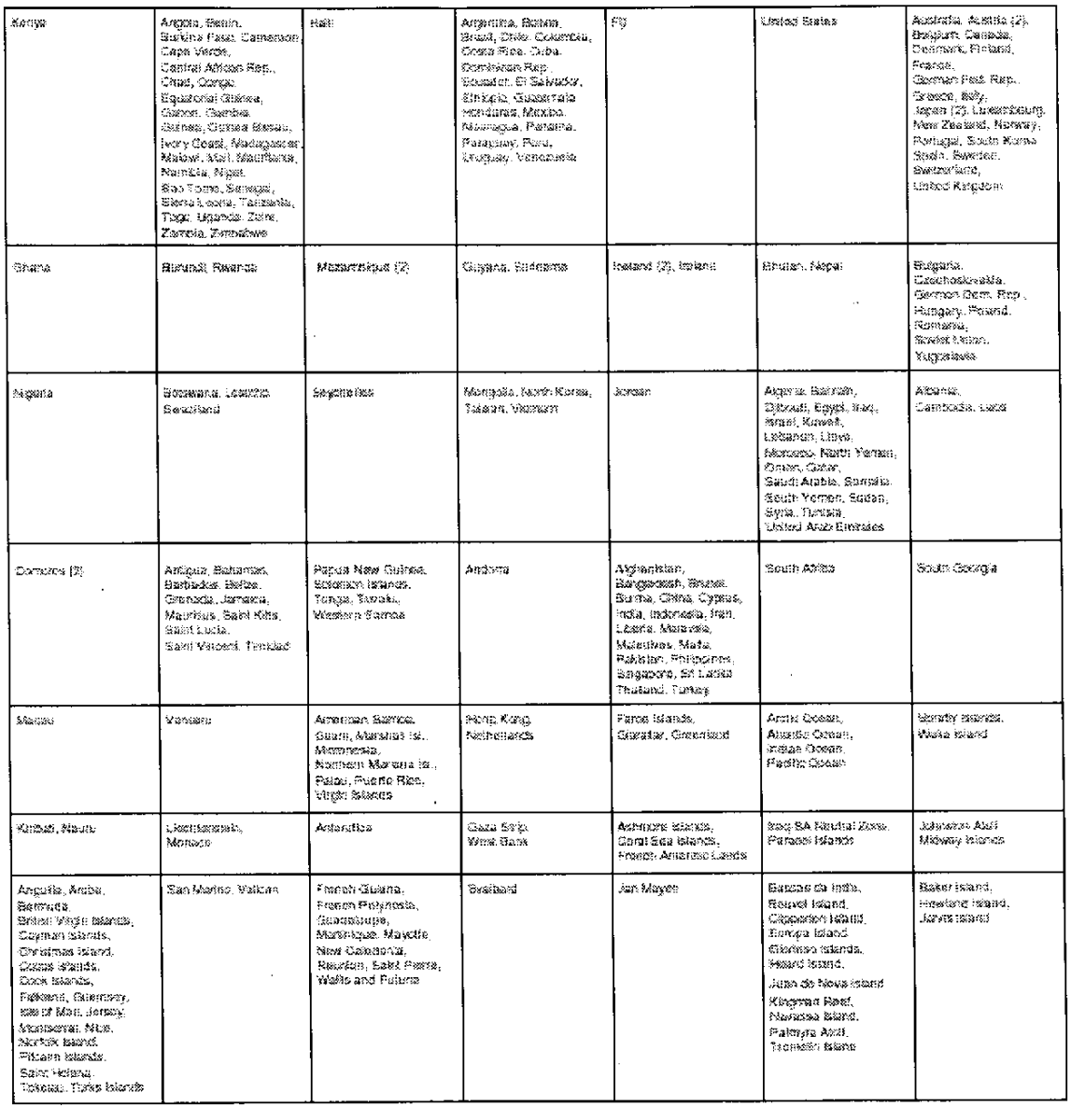

図5-11 リンのシステムの学習例

5.4.3. GalaxyとThemeScape

ワイズ(Wise)らは、有用な可視化情報を生テキストから生成するためには以下の4つの考慮点があるとし、GalaxyとThemeScapeを開発した[24]。

・テキストを構成するものと、それを情報の他の記号表現から区別する

方法の定義を明らかにする。
・生テキストから可視化表現への変換方法が必要である。視覚的探索と分析を各個人が行うことができれば、なお良い。
・意味ある可視化の基礎として、適切な数学的手法や分析のための測度を定義する必要がある。
・テキストやそこから導出される情報を管理するデータベース管理システムが必要である。

GalaxyとThemeScapeは、ドキュメント集合のオーバービューを三次元空間上に図示することを目的としており、以下のストラテジーに基づいて、ドキュメント空間の可視化を行なう。

・三次元レイアウト・アルゴリズムを使って、高次元空間で特徴付けられるドキュメント集合のクラスタリングを行う。
・クラスタの重心は、レイアウト・アルゴリズムの入力とする。
・レイアウト・アルゴリズムは主成分分析あるいは、多次元尺度法が用いられる。
・ドキュメントは、可視化するためにクラスタの重心近くに配置される。

クラスタリングにより、レイアウト計算の複雑さは少なくなるが、レイアウトした時に各ドキュメントの特徴が希薄になるという欠点がある。
こうして作成されたレイアウトを、風景画をメタファとして可視化したのがThemeScapeである。ThemeScapeは以下のような特徴を持つ。

・ThemeScapeからユーザはトピックスに関する情報を得ることができる。
・ThemeScapeは、テキストの二次元的風景画を与えられた領域における主題の強さを表すために、高さを表わす次元を導入する。
・ThemeScapeにはトピックスを分割する境界が存在する。

これらの特徴により、ユーザは複数のトピックスに依存する領域や、ドキュメント集合間の影響などの判断がしやすくなる。
Galaxyは、ドキュメントの集合を夜空の星のように表現した可視化システムであり、Galaxyは星空のメタファである。Galaxyは、ThemeScapeに対する概観を提供している。GalaxyとThemeScapeによる可視化によっ

て、データベースに何が入っているかという問い合わせに対して、迅速に答えを出すことができる。

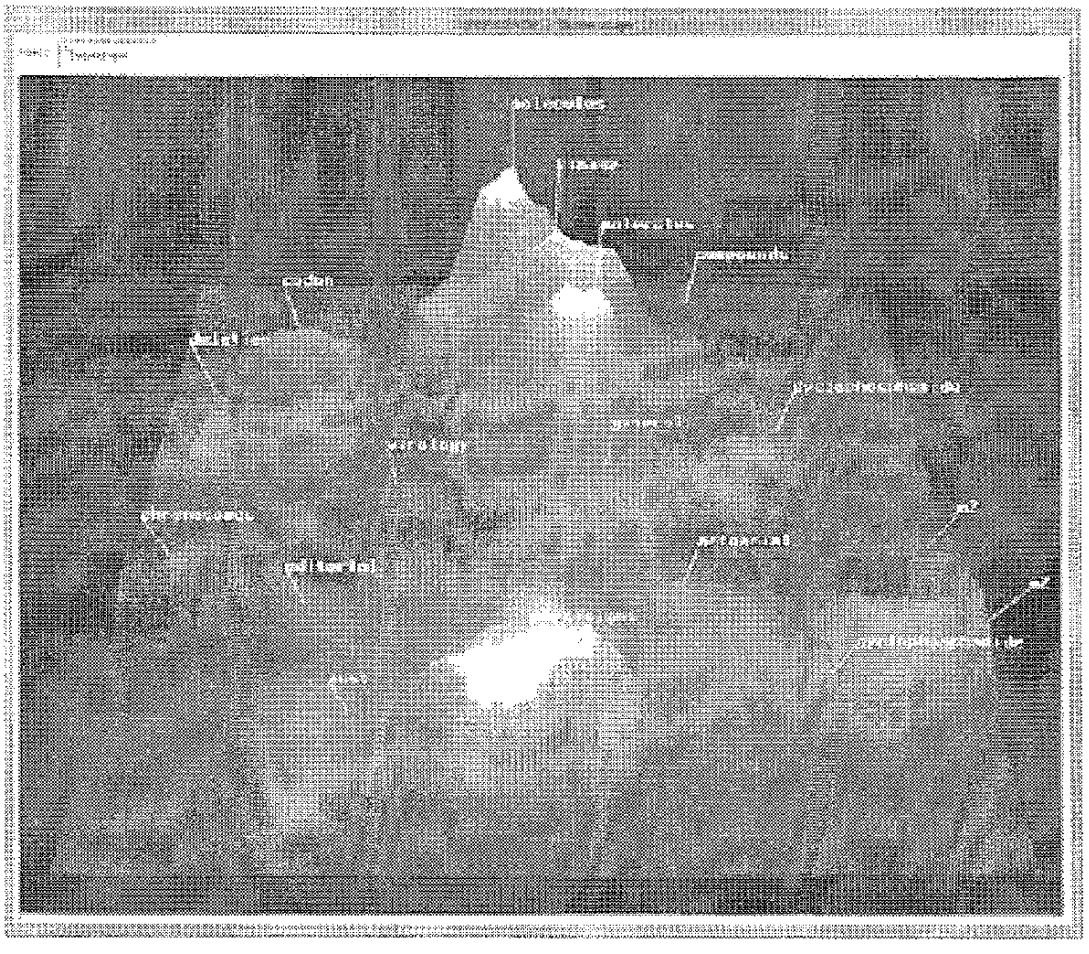

図5-12 ThemeScape

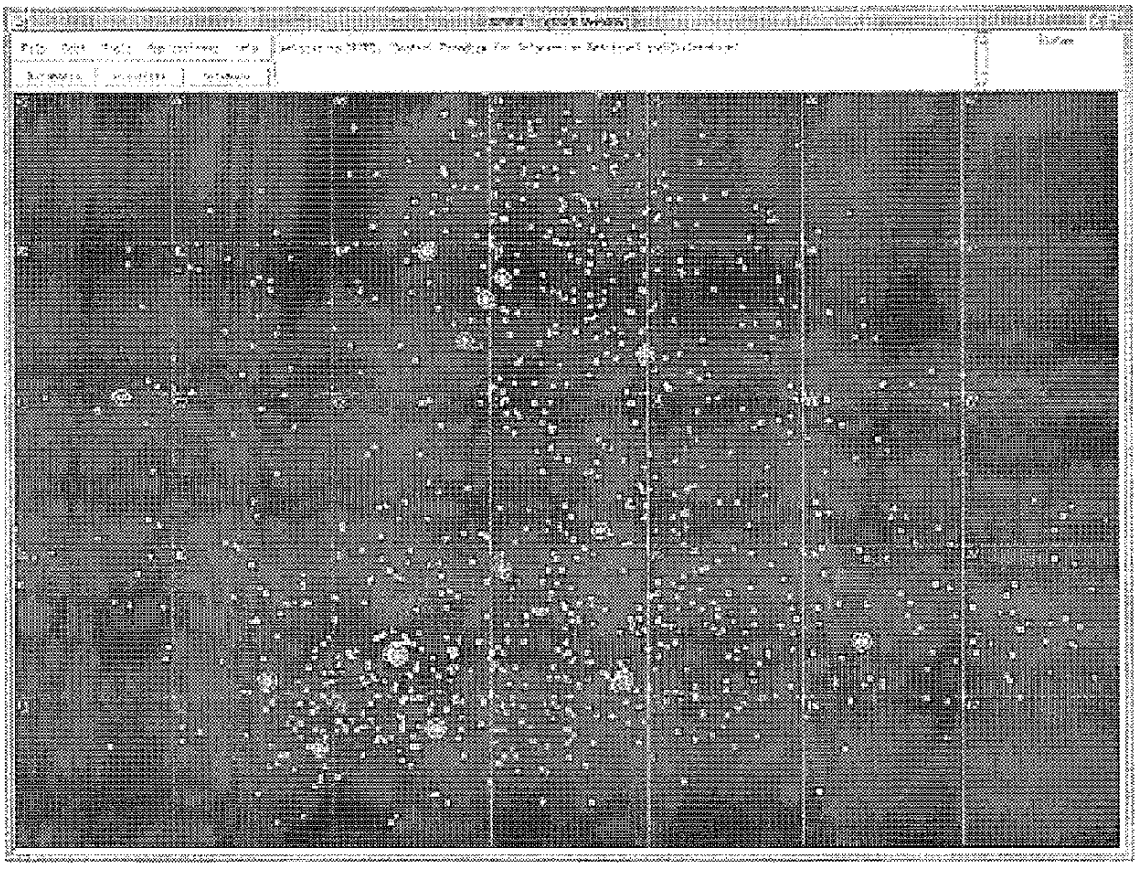

図5-13 Galaxy

5.4.4. GalaxyOfNews

(1) 概要

GalaxyOfNews[25]は、ニュース記事を探索するためのツールである。大規模なニュース記事の空間を探索し、目的とする記事を検索する可視化システムである。GalaxyOfNewsは、関連した記事間の暗黙のリンクを自動的に生成する連想関連ネットワークを構成する、関連構築エンジンをシステムの核としている。これらの関連はピラミッド構造で表現される。この関連を用いてニュース情報空間の可視化が行われる。

GalaxyOfNewsでは、ユーザは対話処理を通して、情報空間の詳細を知るとともに、ニュースの幅広い理解を行うことができる。

図5-14にGalaxyOfNewsの情報空間の初期ビューを示す。図5-14には、抽象的なニュース階層におけるニュースのクラスタの根となるキーワードが表示されている。ルートキーワードは、ニュース・データベース全体

の、抽象的で一つのスクリーンの上に表現可能な概要を提供する。ルート・クラスタは、抽象化と一般化プロセスを通して導出される。表示された空間は非線形であり、魚眼レンズのように中心要素が注目されるように表示される。

図5-14の例で、ユーザが「Media」クラスタの方へズームしたとする。この段階で、GalaxyOfNewsは、ズームされたクラスタの詳細をこの領域にある他のクラスタとともに詳細化する。

ユーザが「Media」クラスタにズームし続けると、空間上の他の領域にあるニュース記事はバックグラウンドに消えていく。このアニメーション効果により、ユーザがシステムの行っていることを認識しやすくなっている。この時点で、「Media」に対する字体は消えかかったような字体になるが、まだ、「Media」は見え続けている。これにより、ユーザが「Media」のサブスペースに入ったことを認識し、大域ナビゲーションマップを使わずに空間の範囲内にいるという文脈を認識できる。さらにズームを行うと、「Media」クラスタに関する記事の表題が表示され、キーワードと記事を結ぶ線が表示される。さらに「Media」クラスタにズームすると、キーワードと記事を結ぶ線が消えていき、記事の本体が項目の下から浮かび上がり、最終的には読み取り可能な大きさに拡大される。

(2) GalaxyOfNewsのモデル

GalaxyOfNewsは、ニュースの管理、探索、可視化を図5-15のモデル上で行う。このモデルでは、ニュース記事を自己構築された情報空間へと集積する。この情報空間に対して、ユーザはインタラクティブに、情報の可視化とアクセスを行う。このモデルの重要な点は、ニュースオーサリングを分けている点である。このプロセスは、受け渡し環境は動的かつインタラクティブであるという点以外、日刊紙編集のプロセスと似ている。このモデルは、以下３つの独立した視点を定義している。

- 著者
- 情報空間デザイナー
- 読者

著者の目的は、情報オブジェクトの作成である。情報空間デザイナーの目的は、情報空間の構築である。読者は、ニュースの内容から作られた情報空間のユーザである。

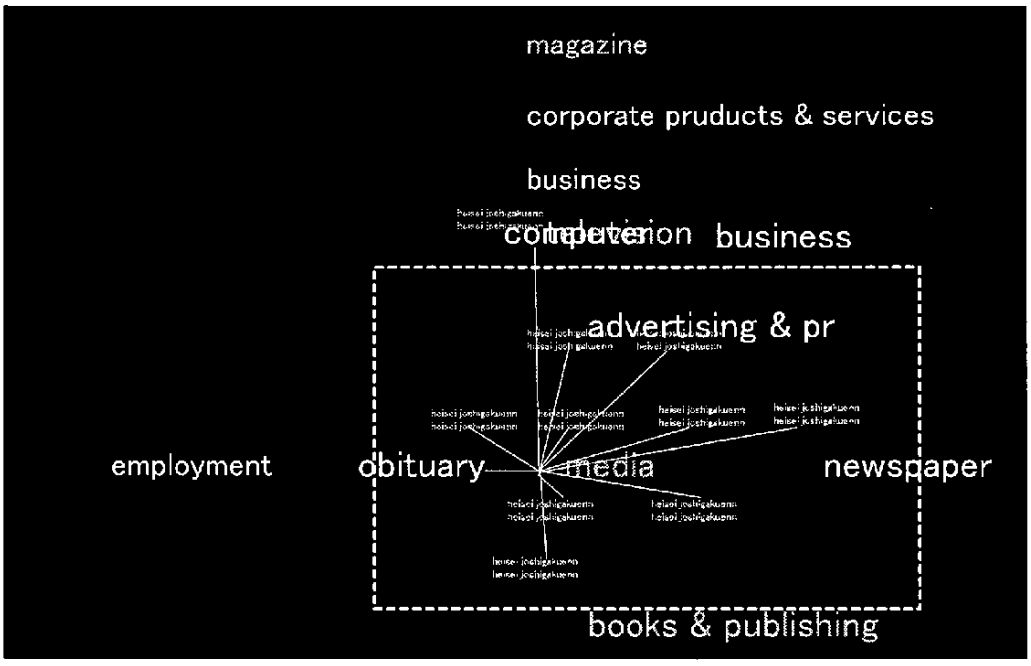

図5-14 GalaxyOfNewsの初期画面

GalaxyOfNewsのモデルは4つのレイヤーからなる。最も重要で基本となるレイヤーはニュース情報である。他の3つのレイヤーはどんな情報を可視化するか、あるいはどのように可視化するかを決定する仕様を定義している。以下で4つのレイヤーについて述べる。

・ニュースベース

ニュースベースは、情報オブジェクトとその注釈からなる。情報オブジェクトとは、テキストで書かれた記事や、音、ビデオ、写真等である。GalaxyOfNewsでは、これらは生データではなく、可視化に適するように処理されたデータとして存在する。注釈は情報オブジェクトの特徴、またはその記号表現から構成される。これらの注釈は、著者によって作成される場合と、情報オブジェクトのデータから自動生成される場合がある。注釈とは、キーワードや、記事の時刻、場所、題名などである。

・情報の関連

情報の集合、もしくは情報のクラスを定義するために、情報オブジェクトのインスタンス間の関連が定義されている。実際には、主題同士の関連、時系列等がこれにあたる。記事同士の関連は持たずに、記事

から抽出された注釈間に関連が張られる。情報空間デザイナーは、情報間の関係、あるいは関係付けを行うプロセスの記述に関して最終的な責任がある。

・空間構造特性

上記のように、情報間の関連をいかにユーザに空間的に見せるかを定義する必要がある。それを定義するのが空間構造特性である。この空間構造特性は情報オブジェクトそのものではなく、情報オブジェクトの集合や集合間の関連によって構成される。空間構造特性は、情報オブジェクトの集合から抽出される。

・インタラクション特性

インタラクション特性は、ユーザとの対話処理の表現と動作を定義する。ビューの詳細化レベルの適切な設定は重要である。

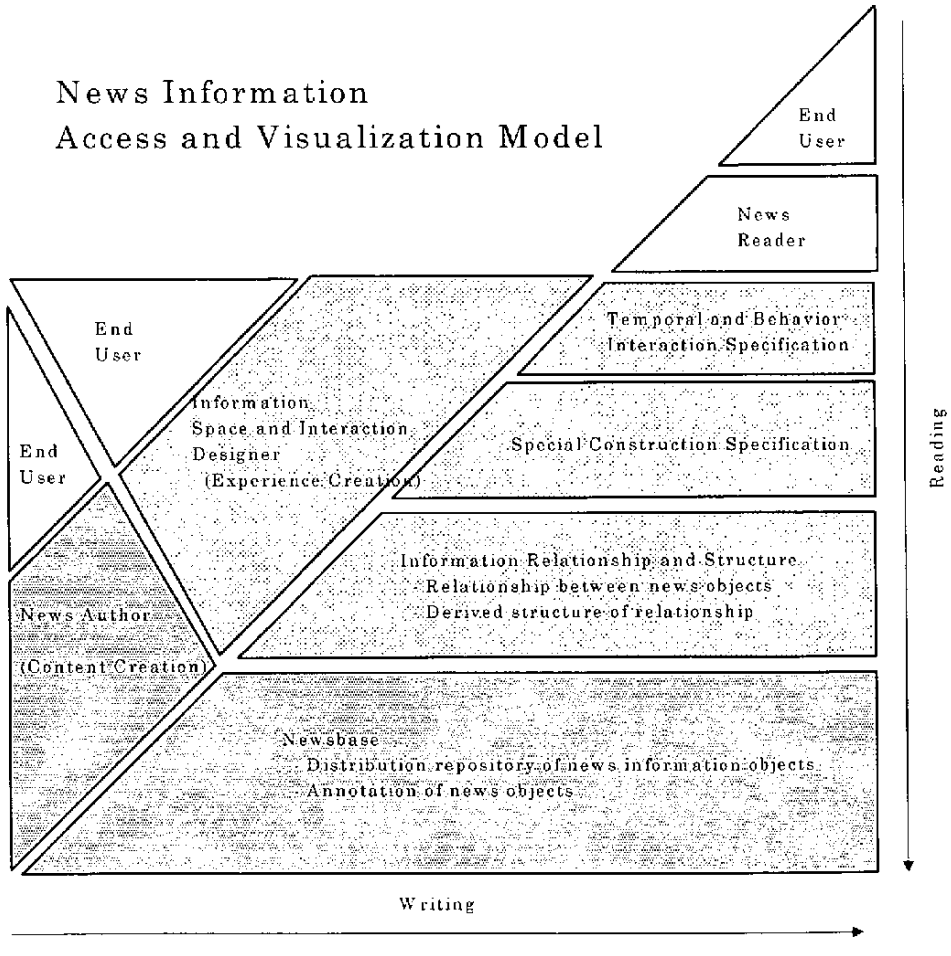

図5-15 GalaxyOfNewsのモデル

GalaxyOfNewsは、動的環境でニュース情報空間の領域と記事のオーサリングを行う。情報空間デザイナーの役割は以下を定義することである。

・ニュース記事間の関連のタイプと関連の構築プロセス。
・ニュース記事とルール間の関係に基づく多次元空間のレイアウトを作るためのルールと、動的に空間のレイアウトを管理する制約ネットワークを構築するルール。
・ユーザが空間を探索する時に、システムが返す応答。

また、情報空間には以下の3つのレイヤーが存在する。

・一般的なものから、より特殊なものへ至るキーワードの階層。
・記事の見出し。
・記事の本体。

キーワード群の階層は、ニュース記事間の関連から抽出される。自動的にニュースから抽出されたキーワード階層を図5-16に示す。

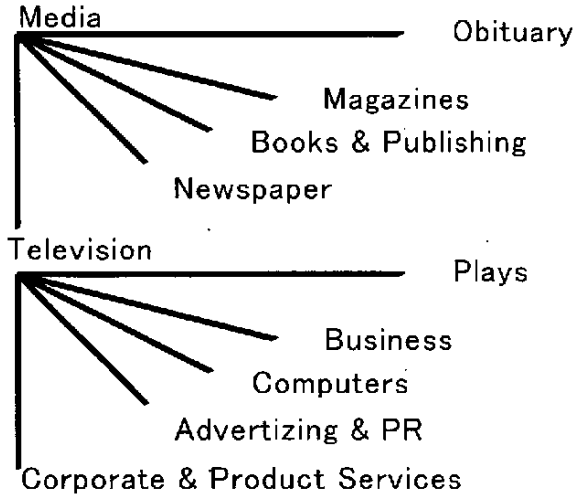

図5-16 GalaxyOfNewsのキーワード階層

GalaxyOfNewsシステムのアーキテクチャを図5-17に示す。図5-17に示すように、アーキテクチャには、以下3つの重要な側面が存在する。

・ニュースのオーサリング
・ユーザが探索する情報空間の設計
・ニュースを探索し、読むこと

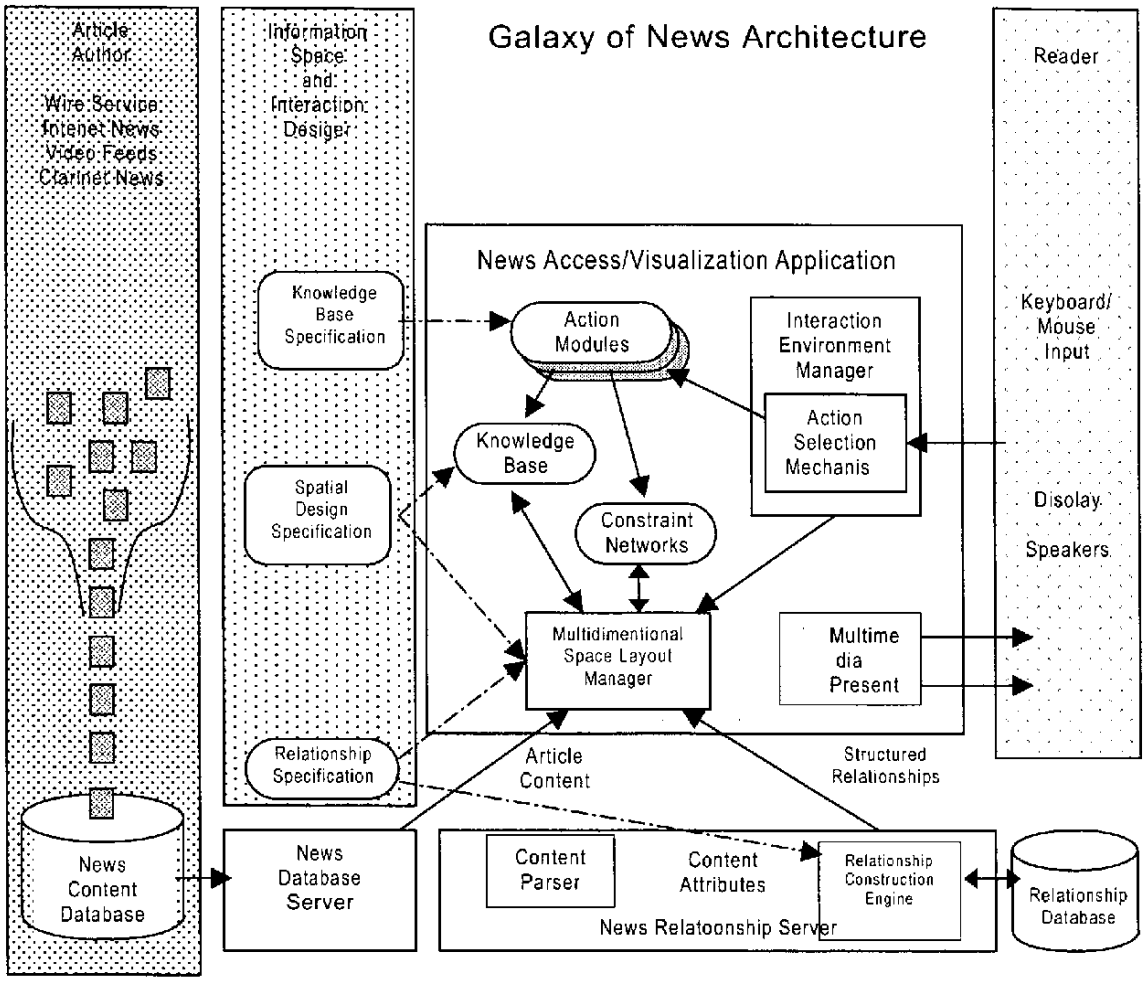

図5-17 GalaxyOfNewsアーキテクチャ

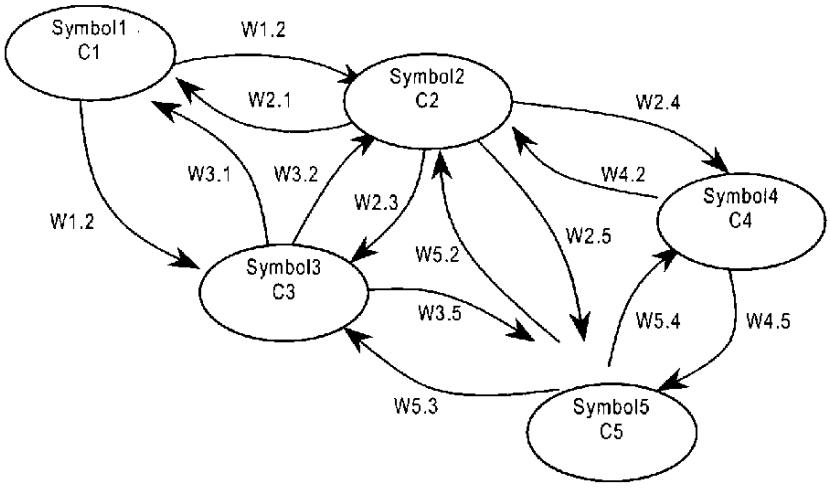

図5-18 GalaxyOfNews ARN

GalaxyOfNewsシステムの核は、ニュース記事間の関連を学習するためのメカニズムと、その表現方法である。GalaxyOfNewsにおいて、ニュース記事間の関連を維持する表現方法をARN（Associative Relation Network）と呼ぶ。ARNのノードにあたる部分は、ドキュメントから抽出されたシンボルである。シンボル間の関連はドキュメント間の関連を定義している。図5-18にARNの例を示す。ARNでは、シンボル間の重み付けしたリンクが維持されている。

ドキュメント集合Dが与えられたとする。このとき、ドキュメント集合Dには、シンボル集合Sが存在する。シンボルS_iの頻度c_iを以下のように定義する。

$$c_i = \sum D_x : \{S_i, \ldots\}$$

ここで、$D_x : \{S_i, \ldots\}$は、S_iを含むドキュメントを表わす。ネットワーク中のS_iとS_j間の重み$W_{i,j}$は以下のように定義される。

$$W_{i,j} = W_{j,i} = \sum D_x : \{S_i, S_j, \ldots\}$$

ここで、$D_x : \{S_i, S_j, \ldots\}$は$S_i$と$S_j$を共に含むドキュメントを示す。

このARNがニュースの構造を学習するために使われる。システムは初期状態では、ARNのネットワークは未学習の状態である。システムのデータベースにニュース記事が格納される時にARNの学習も行われ、システムのデータの増大と共に、ARNも学習を行う。

GalaxyOfNewsシステムとGalaxyとの違いは、Galaxyが静的にドキュメント空間を表示するのに対し、GalaxyOfNewsはドキュメント空間をユーザの動的な操作によって表示している点である。また、GalaxyOfNewsはその検索の入力としてドキュメント中のキーワードを用いている。Galaxyでも内部的にドキュメント間の類似性の計算にキーワードを用いているが、ユーザのフロントエンドにはキーワードをおいていない点が大きな違いである。また、GalaxyOfNewsはアニメーション効果によって焦点・文脈化を実装している点も他のシステムに無い特徴である。

5.4.5. Starlight

Starlight[26]は、テキストやデータベースを含むマルチメディア情報源を利用した諜報活動を支援するために、さまざまな観点から情報を関連付け、専門的な分析ができるように米国パシフィック・ノースウェスト国立研究所で開発されたシステムである。複数の可視化手法をワークスペースと呼ぶ三次元の仮想空間上に展開し、それぞれの可視化表現にまたがる要素間の関連はリンクとして明示することができる(このような手法はbrushing and linkingと呼ばれることがある)。リンクにはドキュメント間の類似度のように、動的に計算されるものや、URLのような直接的参照によるものが含まれる。階層的なデータモデルとリンクによって、ある要素に関連する要素をディレクトリのように入れ子状に展開し、可視化することもできる。これを関連検索と呼んでいる。

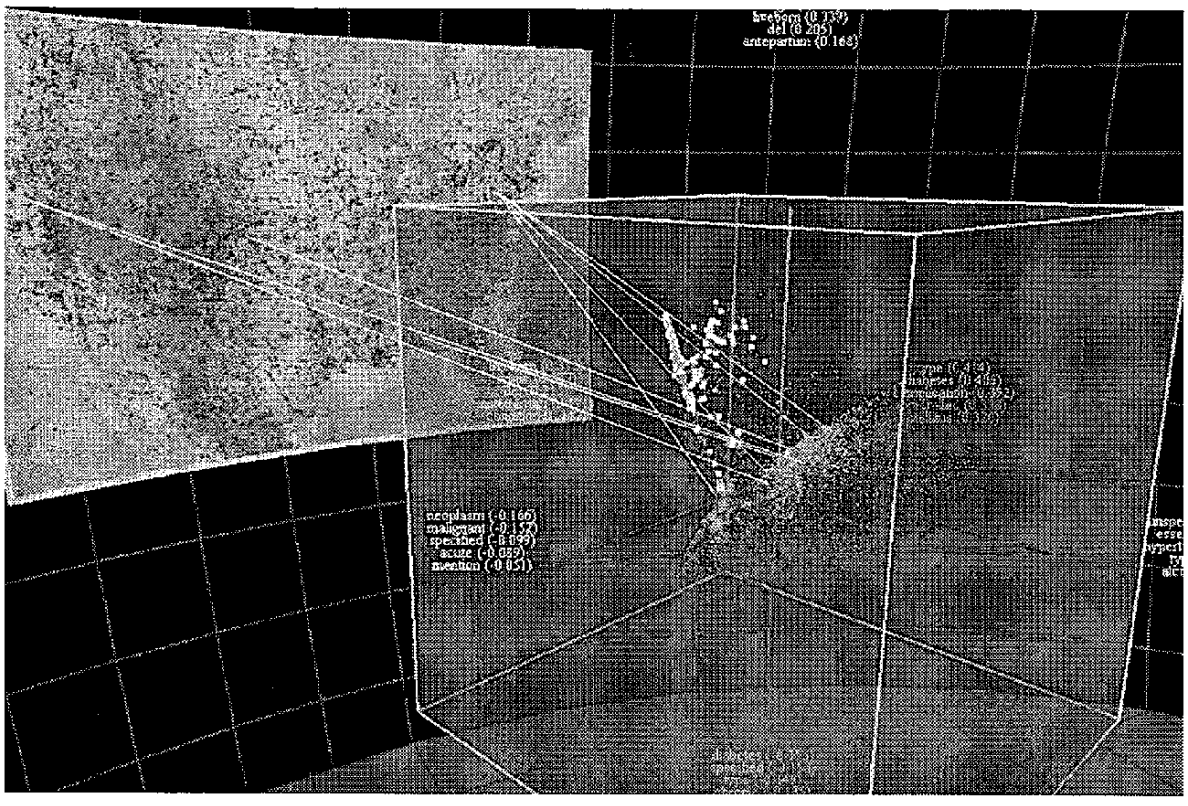

図5-19 StarLight

5.4.6. 評価

ドキュメント可視化の核となるのは、キーワードやテーマといったドキュメントの内容を表現する特徴量と各ドキュメントの表示方法、ドキュメント間の関係の表示方法である。

多くのシステムや技法では、ドキュメントやキーワードを平面や空間上の点とし、ドキュメントやキーワード間の関係は線として表現される場合が多い。ここでの、点 - 線アナロジーはユーザにとって直感的に理解可能なアナロジーであり、ドキュメント情報可視化における最も代表的な技法である。

空間上に各要素を配置するアルゴリズムに関しては、一般的なアルゴリズムはない。ここで、ユーザから見て理解不能なアルゴリズムは、かえってユーザの情報理解を阻害する。

また、機械学習的手法もドキュメントオーサリングによく用いられる手法であるが、学習アルゴリズムが想定したモデルと、ユーザの認識が一致しない場合、往々にしてユーザの情報理解を阻害する。

5.5. インフォメーション・アウトライニング

5.5.1. システム概要

ドキュメント集合を概観するための可視化では、全体像を把握できる程度の抽象度の高い情報を提示する必要がある[27]。しかし、特定の可視化では表現される情報に限りがあり、元々のドキュメント集合が持つ豊富な特性をとらえきれないことが多い。

たとえば、1999年1年間の新聞記事から「2000年問題」に関する記事を検索した場合に、月別の該当記事の件数を見れば、それが年末に向かって次第に増加していることがわかる。すなわち、2000年問題に関する記事は人々の注目度が高まっている、という重要な情報を得ることができる。さらに、2000年になってからは記事数が減少し、話題性が急速に失われているといったことも理解できる。こういった記事数と話題性の関係は、大まかな記事数変化の特徴パターンとして記述することができる。また、月別に記事によく現れるキーワードの分布や、ある月に集中して現れるキーワードを表示すれば、話題変化や2000年問題のみに固有の話題について知ることができる。このようにして関連度順に並べられた記事のタイトル一覧や、単にキーワード別に分類された記事のディレクトリ表示などとは異なる観点からの概観が可能になる。

ドキュメント集合に付与されたメタデータと、検索式を満足するドキュメント集合から動的に計算される上記のような特徴だけを利用して、本文を一切読まないでも望みのドキュメント集合に絞り込めるような可視化が目標となる。このような豊富なビューをもとに、情報のズーム/フィルタ機能と、ナビゲーションを支援する機能とをビュー上の直接操作として実現したものをインフォメーション・アウトライニングと呼ぶ。

直感的にはプログラミング言語smalltalkで採用されたモデル、ビュー、コントローラの3層構造を情報検索向けにアレンジした概念であるということができる。

このように同じデータ集合を異なる観点から可視化するというアイディアはStarlightにも共通するものであるが、インフォメーション・アウトライニングでは次のようなことを重視している。

- 要素間の個々の関連よりも、現在注目している要素集合全体あるいはそれらの部分集合の和や差がどういった特徴を持つかを多面的に可視化する。
- 動的質問と同様にある観点に対応するビューでの直接操作によって

他の観点のビューがどう変化するかを効果的に可視化する。
・互いに関連するビューを関連付け、特定の分野やタスクごとに異なるビューの集合を設計してソリューションを実現する。

このようにして、対象となるデータ集合とそのメタデータに応じて柔軟に可視化表現をビューとして選択することができる。ビューは、それ自体が独立した実態として関連付けられたり、可視化できるデータの一般性や詳細度によって上位・下位関係が定義できる。複雑なタスクも比較的単純なビューの集合を組み合わせることで、ユーザにとって使いやすいシステムを構築することができる。インフォメーション・アウトライニングは実際に行政情報、新聞記事、テレビニュース、Webページなどのさまざまな情報に対して可視化に基づく大規模な情報検索を構築する手法として採用されている。

ビューの多様性はメタデータの豊富さに大きく依存するので、書誌情報などのメタデータを可視化するためには、テキストからの情報抽出やテキストマイニングの技術がきわめて重要である。テキストマイニングでは構文レベルの解析によって「何が＋どうした」という係り受け情報を抽出し、問題分析や意思決定のための情報抽出を行っている。これらの技術によって大量のドキュメント集合に対しても実用的な可視化が実現できる。動画像を含むマルチメディアドキュメントのように個々のドキュメントのサイズが大きな場合には、このようなメタデータを効果的に利用することでスケーラブルかつ多数のクライアントの検索要求に耐える情報検索システムが構築できる。

5.5.2. 評価

インターネット上の情報可視化の節でも述べたが、ドキュメント集合の大規模化に伴い、ドキュメントの状態を概観するための手法の高度化が望まれる。

インフォメーション・アウトライニングは、その解の一つである。ただし、現在、提案・開発されているシステムや理論は初歩的なレベルに留まっている。今後、以下のような理論、あるいは技法を取り入れた新たなインフォメーション・アウトライニング手法の開発が望まれる。

・統計的手法の応用
・言語解析に基づいた手法

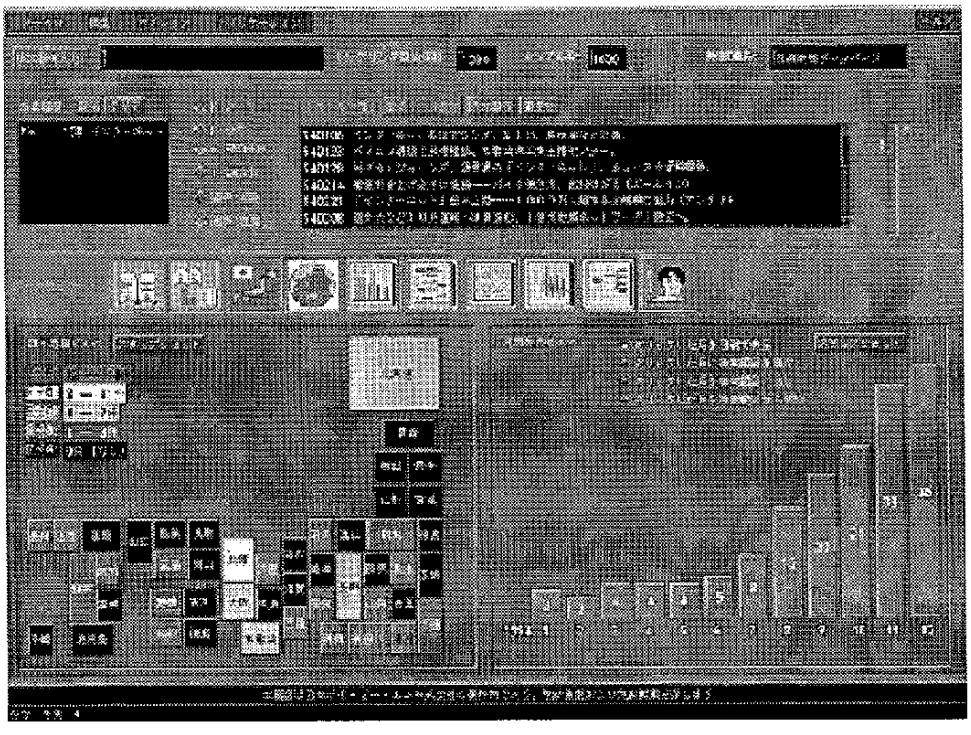

図5-20 インフォメーション・アウトライニング

5.6. ソフトウェア可視化

大規模なコンピュータプログラムは非常に複雑であり、その保守の困難性はよく知られている。大規模システムを理解、変更、修正するには、時間とコストがかかる。大規模なプロジェクトが保守状態になると、プログラマーの生産性は低下し、エラーの増大、プロジェクトの遅延がおこりやすい。設計書の更新は行われずに、ソースコードのみの変更を行うことも行われ、ソースコードが唯一のシステムの動作を記述するドキュメントになることもある。結果、システムの動作を理解するためにソースコードを読むことに膨大な時間を費やすことになる。

複雑なシステムやソースコードを取り扱う上で、プログラマーの生産性を向上させる一つの方法として、ソフトウェア可視化ツールが提案されている。もともとソフトウェアのデータやアルゴリズムは、それ自体が抽象的な事象であり、それらを可視化するには工夫が必要である。たとえば、ソフトウェアの可視化を考えると以下のような課題に突き当たる。

- 小数等のデータ型の可視化
- アラートを挙げるという事象の可視化
- 繰り返しというアルゴリズムの可視化
- 日付を可視化する場合、カレンダのメタファが適当か、あるいは99年09月09日といった文字列による表現が適当であるか、などの判断
- アラートを示すには四角い箱に入ったスピーカのアイコンを用いるより、屋外用スピーカのアイコンの方が適当である、などの判断
- プログラムの構造や振る舞い、制約、状態の可視化

この問題に関して、さまざまな研究、システム開発が行われてきた。ソフトウェア可視化ツールはグラフィック技術を用いてプログラムを可視化する。ここで重要なことは、可視化を行うことにより、ソフトウェアの理解が容易になるということである。ソフトウェアには、可視化すべき3つの特性がある。

- ソフトウェア構造

  ソフトウェア間の関連を表現するには、有向グラフが一般的に用いられている。CASEツールやプログラム分析ツールも有向グラフを基本としている場合が多い。たとえば、グラフの各ノードが関数を表現し、

関数間の呼び出しの関係を有向グラフで表現することがある。

・実行時動作

アルゴリズムアニメーションを用いて、アルゴリズムの動作をグラフィカルに表現する。プログラムの特徴や、バグ修正のトレース、パフォーマンスを表現する。

・プログラムソースコード

プログラムやソースコードをグラフィカルに表現するために、プログラムの分岐や繰り返しを示す制御文を色分けして表現するエディタが一般的に用いられる。

これまでのソフトウェア可視化ツールは、小規模なシステムをターゲットにしており、大規模なプロジェクトに対応することはできない。アルゴリズム可視化ツールは、可視化の前に設計者にコードの理解を求める。この方法では、大規模システムに適応することは不可能で、プログラマーに問題のあるコードの位置を発見することの手助けにはならない。大規模システム開発の一般戦略は、プロジェクトを、階層構造を持つモジュール群に分割することである。これらのモジュールは画面に個々に表示されるので、モジュール分割は、システムの全体を眺めるような可視化を困難にしている。

このように、既存のシステム可視化ツールでは大規模なシステムの全体を把握するというような要求に答えることは困難である。システム可視化ツールで最も研究・開発が行われているのは、ビジュアルプログラミングと呼ばれる分野である。

ビジュアルプログラミングでは、ソースコードを中心としたソフトウェアの可視化を行う。テキストレベルのソースコードは確実にシステムの特徴をすべて備えている。ソースコードから、システムの全体を概観するためのツールを用いれば、システムの全体の特徴を掴むことを容易にできる。

5.6.1. ビジュアルプログラミング

1970年代中頃から今日に至るまで、ビジュアルプログラミングの研究分野では、プログラムのアルゴリズムやデータ構造という抽象的な対象を、いかに少ない概念で整理し、可視化するかが議論されてきた。ビジュアルプログラミングの利点は、開発者が特定のプログラミング言語の仕様に煩わされることなく、複雑なアルゴリズムやデータ構造を解析したり合成することに意識を集中できることにある。

ビジュアルプログラミングを支援するソフトウェアには、インタラクティブなインターフェースが用いられている場合が多い。

ビジュアルプログラミングでは、“抽象的概念を可視化する時に最も困難なことは、曖昧さの海で溺れるのを未然に防ぐことである”といわれているように、プログラムのアルゴリズムやデータ構造を簡潔な概念だけで表記する研究が進められてきた。現在では、プログラムやシステムを静的、動的、機能的という3種類の視点から分析・設計し、それぞれの視点に適合した仕様化技術を用いて、各々の情報を可視化すべきであるとの合意に至っている。仕様に用いる図式の種類を限定することによって、開発者の学習期間を短縮したり、第三者にとっても理解しやすい仕様を提供できるという利点が得られる。

5.6.2. SeeSoft

AT&Tで開発されたSeeSoft[27] [28]は、プログラムソースコードの可視化を目的として開発されたアプリケーションである。プログラムの各行を細長い線にマッピングすることで可視化を行い、5万行程のコードを同時に分析できる。可視化された図は、ある統計値により色づけされる。たとえば、赤は編集される頻度が高い部分、青は編集される頻度が少ない部分というような色づけを行い、一目でユーザに青い部分がコードとして安定している部分であるということが認識できるようになっている。

SeeSoftでは、プログラムのソースコードの各行を細長い長方形として表現する。一つのファイルのソースコードは、長方形を縦に並べて細長い線のように表示される。あるアプリケーションは、複数のファイルのソースコードから生成されるのが通常であるので、一つのアプリケーションのソースコードは、複数の細長い線の集合として表示される。

図5-21の左の図は、通常のサイズのテキストで、真ん中の図は、フォントを小さくした図である。右の図は、SeeSoftの表示である。

ピクセル表示は、コードの各行を色の付いた小さなピクセルとして表示する表示方法である。ピクセル表示を用いると、前述の線表示よりも高密度な情報表示が可能である。ピクセルは、列の中に左から右へと並べられ、列の集合としてカラムがあり、一つのカラムが一つのファイルに一致する。この表示方法では、数百万行のコードを一つの画面に同時に表示できる。

ピクセル表示は、図5-22の左下のブラウザーウィンドウにおけるスクロールバーに使われる。このブラウザーは、3つのテキストの見え方を含んでいる。左がピクセル表示、中央が、ピクセル表示の選択された部分を表

示する線表示、中に配置された小さなウィンドウは線表示中の選択部分を表示した実際のテキストである。

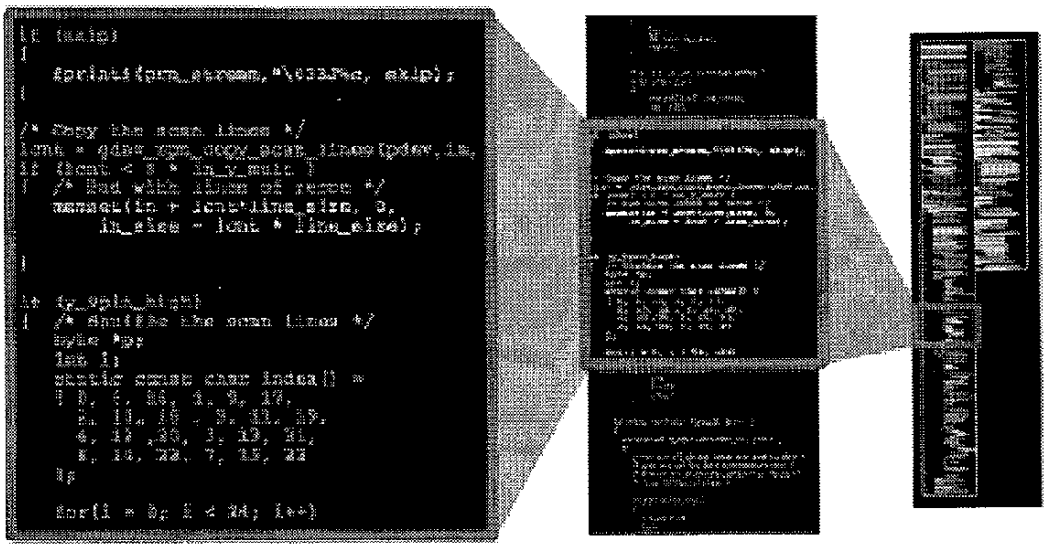

図5-21 SeeSoft 線によるコードの表示

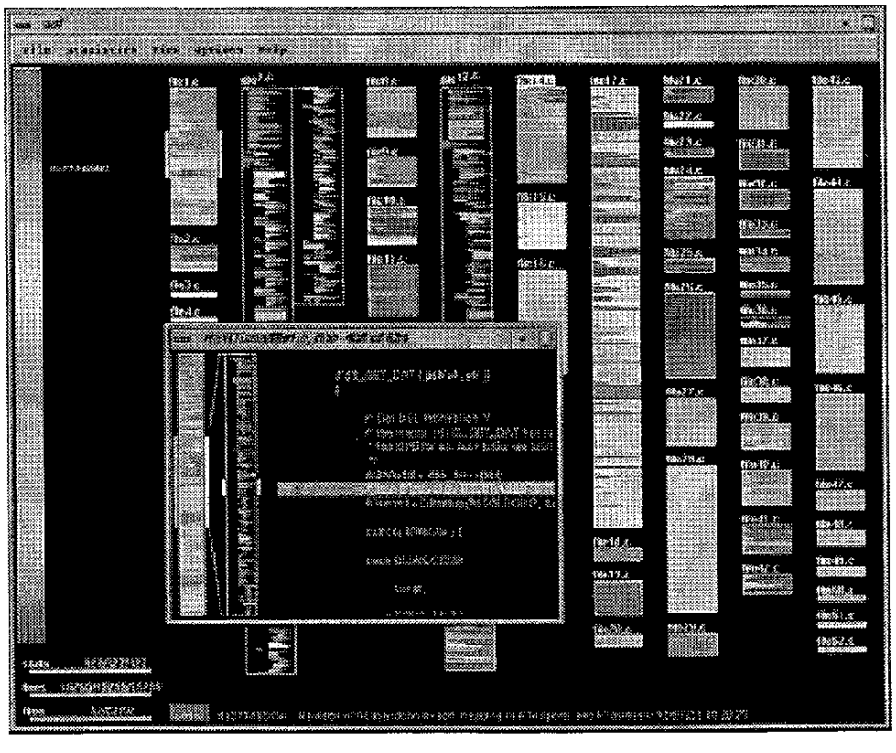

図5-22 SeeSoftにおける3通りの表示

バージョンコントロールシステムは、ソフトウェア開発において広く使われている。バージョンコントロールシステムのデータベースは行レベルの情報を維持している。これら行レベルの情報とは、例えば以下に記す情報である。

・各行がいつ変更されたか
・プログラマーの誰がコードのある部分を記述したか
・どこにバグがありバグフィックスがどうされたか

図5-22はバージョン履歴の図である。バージョンデータを探索するツールを用いることにより3通りの生産性向上が考えられる。

まず、プロジェクトの新メンバーであるプログラマーがソースコードを理解することが考えられる。たとえば、作成したプログラマー毎にコードが色分けされていれば、誰によってコードが変更され、その動き方を認識しているのが誰であるかを確認できる。

第二に、誤りを引き起こしそうな箇所がどこであるかがわかるということである。たくさんの色の入り混じったファイルがあれば、複数のプログラマーによって手が加えられており、エラーを含む頻度が高いと理解できる。このようなファイルはリエンジニアリングする価値があると判断できる。

大規模プロジェクトの同時作業において、サブシステムからシステムの足跡をたどることは困難である。最後は、このような場合、サブシステムの責任者が直近に変更されたコードのみを見ることで、現在のサブシステムのコードが全体の標準から外れていないか検査できる。図5-22では、開発に明らかなパターンが存在している。各ファイル中におけるソースコードの色の違いから、最近変更されたファイルとより安定しているファイルを区別できる。

図5-23はカラム分解モードを用いてコードを線表示で修正を重ねた率を表示している。修正そのものは作業戻りを示し、生産性の低下を表しており、修正を重ねることは、さらに大きな作業戻りを指し示している。図5-23では、各カラムは２つに分割されている。左側は、バグフィックスが加えられた全ての行を示し、右側は、バグフィックスで引き起こされたバグを表している。半直線は、元のバグフィックスが正常に機能したことを示している。実線は、修正に修正が加えられたことを示している。左の半直線の色は、最初のバグフィックスの世代を示し、右の半直線は、最新の

バグフィックスの世代を示している。もし、それぞれの半直線の色合いが異なるのであれば、本来のバグフィックスとその後の二重の修正は時間的に離れて発生したということを表している。

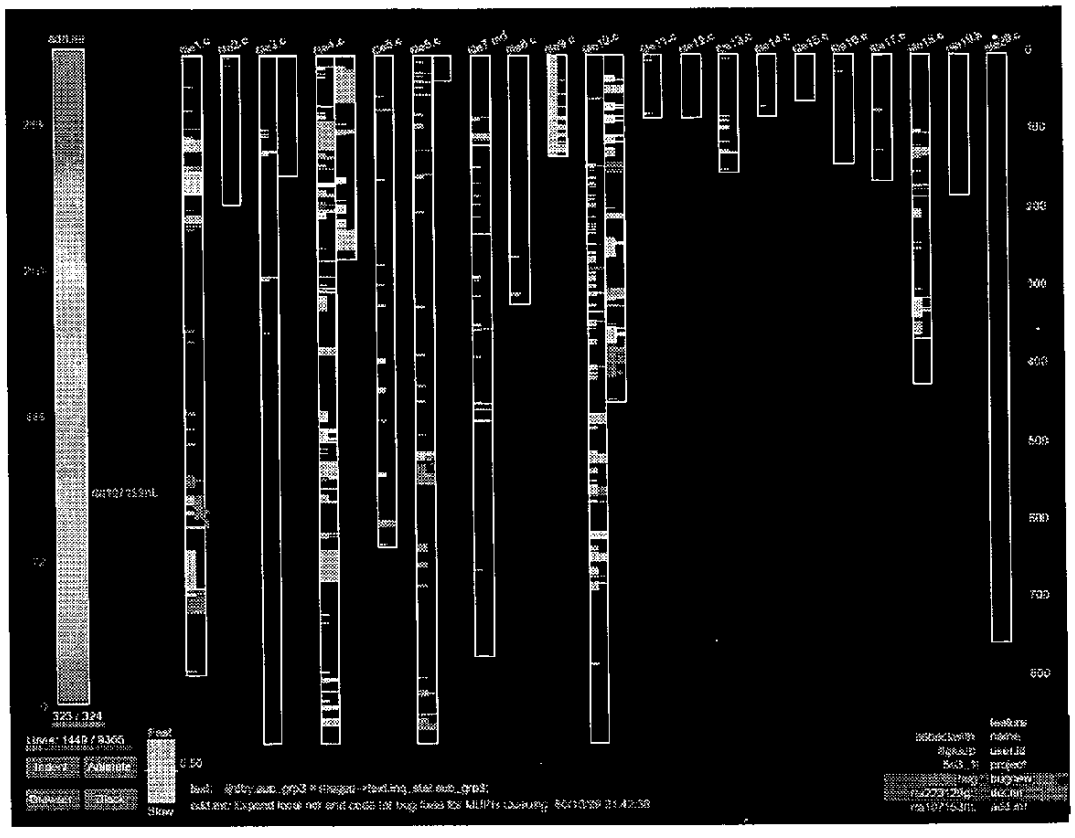

図5-23 バグフィックスの履歴

SeeSoftは、プリプロセッサによる分岐の複雑さを表示することもできる。プラットホームに依存するソースコードは、プリプロセッサを用いて記述することが一般的である。プリプロセッサは、#Ifdef、#endif などであり、条件コンパイルを制御する。これらのプリプロセッサは、ソースコードの理解を妨げる場合がよくある。SeeSoftでは、複数のプラットホームのファイルを比較し、プリプロセッサへの依存度を視覚的に表示できる。

また、ステートメントの分岐の複雑さは、複数の条件文に囲まれた複数の繰り返し文等によって生起される。一般的に、ネスティングが複雑なほど実行した時の条件を決定することはより困難になる。SeeSoftは複数フ

ァイルのコードをピクセルで表示し、色を用いてネスティングのレベルを表示できる。この時、線はレベル4以上のネスティングを表示する。この可視化により、再構成すべき対象の選択が可能になる。

実行時の性能を最適化することは、システム開発で考慮すべき重要な問題である。ソースコードの実行時の性能を計測するために、プログラマーは最もCPU時間を消費する箇所を決定し、その時間を減少させようとする。このプロセスはコードチューニングと呼ばれる。ほとんどのコンパイラは行レベルのプロファイルをサポートする。SeeSoftはユーザのテストの結果から得たソースコードの「HotSpot」を表示できる。カラー・パターンをより見えるようにするためにライン・インデントなしで線表示を用いている。色は、ラインが実行した回数を表している。たとえば、赤は実行回数が多い箇所を示し、青は実行回数の少ない箇所を示している。灰色は実行されなかった行を示している。灰色はコメント等である。

5.6.3. 評価

ソフトウェアのソースコードそのものを、何らかのメタファによって示すことは、それほど困難な作業ではない。しかし、ソフトウェアが持つ概念（たとえば、データ型、改変履歴そして動的特性など）を表示する方法に関しては、一般的な方法がない。もともと、それらの概念は抽象的な概念であり、実体としての表現が難しい。

SeeSoftは、実体として存在しない概念をうまく実体化している。ここで使用されている技法は、先に述べた焦点・文脈化、およびドキュメントオーサリングなどで使用される方法に基づいている。SeeSoftの成功理由は、これら基本的な技法と実体の無い概念のマッピングの仕方、すなわちメタファにある。

良いメタファは、良い可視化の源の一つであるが、ある概念表現をするために、どのようなメタファを取ればよいかという一般則は、今のところ無い。

5.7. オーサリングツール

5.7.1. システム概要

マルチメディア情報のオーサリングでは、情報の作り手が伝えたい意味をどのように表現するかということと同時に、いかに簡単に情報を生成できるかといった課題についても考える必要がある。

マルチメディア情報のオーサリングツールが可視化システムに示唆を与える点は主に、その操作性に関する点である。マルチメディアオーサリングツールにおいて、さまざまな操作モデルが採用されている。伝統的なグラフィックツールで採用されている操作から、ユーザの日常生活のメタファに基づいた操作まで、数多くの操作が提案・開発されている。

ここでは、幾何学的な三次元図形を描画するシステム、手書きの描画情報から三次元図形を生成するシステム、そして動画編集システムを例にとり、考察を行う。

（1）幾何学的な三次元図形の生成

インタラクティブなインターフェースには、WYSIWYG（What You See Is What You Get）という原則がある。WYSIWYGは、ディスプレイに表示された書類が、そのまま印刷されるという意味であった。後に、ディスプレイ上でユーザが見ている対象が操作可能な対象であるという意味が生まれ、GUIを語る上での原則の一つとなった。

二次元図形を作成するためのツールは多数ある。三次元図形を二次元モデリングツールのように簡易に描くことを目指して開発されたシステムも、近年のシステム価格の低下により多くのツールが、商品化されようとしている。

また、OpenGLなどのプログラミング言語の普及と、コンピュータ処理速度の向上によって、三次元図形を描画したり、回転させるといった操作を描画対象に対して施すのは、それほど多くの専門的知識を必要としなくなった。

Claymore[29]は、三次元図形生成ツールの一つであり、描画のアイコンとして多面体や四角錐、三角錐、円錐、立方体、角柱などを提供している。ユーザは、これらのアイコンをマウスで選択し、キャンパス上に配置して描画を行う。描画中は、三次元空間で図形を指示したり、移動させたりすることができる三次元マウスカーソルを使用できる。また、三次元ハウンティングボックスによって操作可能な範囲を指定して図形の切り取り、拡大/縮小ができる。

（2） 三次元図形生成システム

三次元図形生成ツールを用いれば、幾何学的図形を作成することは容易となる。しかし、紙と鉛筆を用いて考えをまとめる時に、メモやスケッチでアイディアのアウトラインを表現する場合には、そこで描かれるものが球や立方体といった、きちんと定義された図形でなければならないという制約は受けていない。

Teddy[30]は、ユーザが描いた二次元の自由曲線を三次元の絵のシルエットとしたり、三次元図形への操作としたりして、三次元多面体を生成するモデリングツールである。

たとえば、ユーザが図形の内部に描いた閉曲線は、突起やなでつけを開始する操作と解釈し、図形の外部に始終点を持つ曲線を描くと、図形の切り取り線と解釈する。また、一度生成された図形に対して形状を変化させる操作も、変化させる範囲と変化させる方向と距離を2ストロークの自由曲線で入力するだけで指示できる。

情報可視化システムにおいては、単に表示だけではなく、表示された対象の操作の良し悪しも重要なポイントとなる。Teddyのように、我々の日常的な動作のメタファを導入することも、一つの選択肢であるといえる。しかし、我々の日常動作は定義が明確でないものも多い。安易なメタファの導入は、システムが要求するモデルとユーザがシステムに関して構築するメンタルモデルとの乖離を生むこととともなる。

（3） 動画像編集ソフト

情報は表現の軸を増加させることによって、より多くの意味を表現できる可能性が広がる。動画データや音声データがデジタルデータとなり、編集時の情報劣化に関する問題が解消されただけでなく編集ソフトウェアも安価に入手できるようになった。近年、CCDカメラ内蔵の携帯電話やノート型パソコンが低価格になり、動画像編集ソフトウェアがあらかじめインストールされているコンピュータも多くなっている。また、電子メールに動画を添付して送ることも行われるようになった。

動画とは、時間軸を持った画像や図形である。音声も時間軸を持つデータであるので、これらのデータは時間を軸として合成できる。動画や音声、オーディオデータには、それぞれを再生するために必要な継続時間という属性がある。これらのデータがデジタル化されたことによって、最終的に作成される動画のタイムラインと呼ばれる時間軸に、データの相対的な再生開始時刻を指定して配置するという操作によって、各データを同様に扱

うことができる。継続時間属性を持たないビットマップなどのデータも、表示を開始する時刻と終了する時刻を指定することによって、動画と同様の取り扱いを行える。簡単な動画像編集ソフトウェアでも、各データの特徴を可視化するために様々な工夫がなされている。さらに高度な動画像編集ソフトウェアでは、視覚情報ではない音声情報を、音の波形として可視化しているものもある。音の波形が可視化されることで、音の高低や強弱を目で確認することができ、精緻な編集が可能となる。

動画像編集ソフトウェアでは、マルチメディアデータに新たな属性を付加させる、あるいは属性を変換することによりデータを可視化し、編集や合成を可能にしている。

5.7.2. 評価

オーサリングツールが可視化に与える示唆は、ソフトウェア可視化と同様に、メタファの利用についてである。オーサリングツールは時間やモダリティといった抽象的な概念を、メタファを使って表現している。このメタファにより、対象を編集する際の操作性の向上、対象に新たな意味を付け加える事などが可能になっている。ここで、使われている時間やモダリティに関するメタファは、一般的なメタファであり、いろいろなツールで採用されている。

また、日常的動作のメタファによる操作も、可視化システムの動作を考える上で示唆的である。日常動作のメタファは、ユーザにとってわかりやすい操作を提供できる反面、一人よがりなメタファがユーザを混乱させる結果にもなる可能性がある。

6. インターフェースモデル

6.1. インターフェース設計ガイドライン

これまでの章で行ってきた、情報可視化に関するレビューに基づき、ドキュメントオーサリングを支援する情報可視化システムを主たるターゲットとした時の、インターフェースモデルに関して考察を行う。

SIDE(Structured User Interface and Evaluation Method)プロジェクト[31]によるユーザーインターフェースガイドラインにおけるユーザーインターフェースの設計項目として表 6-1の項目群が挙げられている。

これらの項目のうち、ユーザやシステムに関する項目（タスクの明確化と対象ユーザの明確化）に関しては、以下のように定義可能である。

- タスク：情報検索を伴う、インタラクティブなドキュメントオーサリング。
- ユーザレベル：情報検索を行おうとするユーザのレベル。コンピュータ利用に関し、まったくの初心者ではないユーザを対象とする。

このような、特定化を行った上で、ドキュメントオーサリングの可視化による支援を考える時、関連性の高い項目は表 6-1の網掛けされた項目に限定される。

全体的なこと	プロセス	
	楽しさ	
	達成感	
	信頼性	
	安全性	
	透明性	
	異文化対応	
	ユニバーサルデザイン	
	ヘルプ	
ユーザーインターフェースの構築	構造化	冗長性
		一貫性
	情報入手の効率性	マッピング
		画面要素間の関係のわかりやすさ
		メタファ
		識別性
		可視化
		視覚特性
		メンタルモデル
		強調(グラフィックス、意味)
		一貫性
		簡素性
		アフォーダンス
		手がかり
	理解のしやすさ	デフォルトの提供
		エラーへの対応(戻れる、中断できる、確認してくれる
		寛容性(ユーザによる自由な制御)
		カスタマイズ
		記憶負担の軽減(再生よりも再認)
		ユーザ主体
		システム状態表示
		本質的な情報の表示
		用語メッセージ
		簡素性
		ユーザの位置の表示
		階層構造
	可視化	レイアウトの良さ、美しさ
		一貫性
		簡素性
		コーディング
		フォント/文字、図、アイコン
		色
		動画、音
	操作性	身体的負担軽減
		操作感(反応速度等)
		システム側の状態遷移の適切さ
		学習の進捗状況への対応(習熟度)
		効率(操作の軽減)
		フィードバック
		音
システムの把握	システムの把握	
	タスクの明確化	
ユーザ	対象ユーザの明確化	
	ユーザレベルへの対応	

表 6-1 SIDEによるユーザインターフェース設計項目

6.2. 設計プロセス

可視化システムを実現するにあたり、以下のようなデザインプロセスを経てシステムの設計が行われると仮定する。ここで参照するデザインプロセスは先に述べた、SIDEプロジェクトのデザインプロセスを参照している。

- 情報の把握
  - 目的の明確化
  - システムの把握
  - ユーザの明確化
- 情報の構造化
  - 情報の分類
  - 情報の優先順位
  - 情報の提示順序
- ユーザーインターフェースデザイン
  - ストーリーの構成からフローチャートの完成へ
  - デザインコンセプトに基づく画面間の方針決定
  - 画面操作部の可視化
- ユーザーインターフェースの評価

これらのステップの中で、可視化システムの構築モデルを考える際、最も関連があるのは「情報の構造化」のステップであろう。そして、次は「情報の把握」である。

本節では、「情報の把握」のサブ・タスクに関して議論を行い、「情報の構造化」に関する考察は次節で行う。

「情報の把握」のうち、「目的の明確化」はドキュメントのオーサリングとする。また、「システムの把握」、「ユーザの明確化」に関しては次のように考えられる。

- システムの把握

  ドキュメントのオーサリングをシステムの主たる目的とする時、ここまでの情報可視化システムに関する概観を通して、情報可視化システムの処理対象（表示対象）となるのは、ドキュメント群、あるいはドキュメントを特徴付けるキーワードなどの諸特性、そしてそれら特性に関する統計量である。

これらの表示対象を表示する時に、領域特殊な特性をいかにグラフィック化し、ユーザとのインタラクションを設定するかが問題となる。ドキュメントオーサリングの場合、ドキュメントが表わしている領域にルーツをもつ特殊性よりも、ドキュメント量の多さおよび、収集されたドキュメントが内包する領域の多さの克服が、システム的に支援すべき第一番目の項目となる。

これを行うためには、表示方法の検討と並んで、各ドキュメントを特徴付ける特徴量の検討が必要である。

・ユーザの明確化

ドキュメントオーサリングシステムのユーザは、現在、コンピュータシステムに関して一定の経験があるユーザ、少なくとも対象とする領域の情報検索の経験があるユーザを暗黙裡に仮定しているものが多い。ドキュメントオーサリングや情報検索に関して、初心者ユーザのモデルは、まだ明確ではないことから、情報検索の経験を持つユーザを可視化システムのユーザとして仮定する。

6.3. 可視化システムのモデル

この節では、前節で述べた「情報の構造化」ステップにおいて、先に示した各設計項目における、可視化システムとの関連性と選択可能な方法について考察を行う。

6.3.1. 構造化

(1) 冗長性

可視化される対象を加工して、個々の対象を識別可能な特徴を各対象に付随させる。ドキュメントオーサリングで最も多く用いられる特徴は、キーワードである。他にドキュメントの一次情報やアクセス記録、更新記録なども特徴として捉えることができる。

サルトンによるドキュメントのベクトル化表現は、内容やセマンティックスに基づいた検索を提供するための冗長性の一例である。

(2) 一貫性

ドキュメントの構造は、ドキュメントオーサリングの観点から一貫する必要がある。可視化に特異的な構造がドキュメントの構造に影響を与える事は避ける。

6.3.2. 情報入手の効率性

(1) マッピング

可視化されて表示された対象が、実際の何に該当するかが直感的に理解できるようにする。

アイコンの多用はかえってユーザの混乱を招く可能性がある。意味不明なデザイン、あるいは作成者の一人よがりなアイコンの使用は避ける。文字表示の方が良いインターフェースとなる場合がある事を考慮する。

(2) 画面要素間の関係のわかりやすさ

対象を異なった側面から見る場合、モードの概念を導入する。モードを取り入れる場合、“緊急脱出”コマンドを用意する。

モードの替わりに、他のウィンドウを表示しても良い。また、モードがネストする場合、モードナビゲーションもしくはモードマップなどを同時に表示する。モード名をつける場合、作成者の一人よがりな名前付けは避ける。

(3) メタファ

対象を表示する際、表示したい特徴以外の情報は圧縮されるような表現方法を選択する。

SeeSoftにおける、ソースコードのラインと可視化された情報としての線との対応関係は良いメタファの参考である。また、ThemeScapeにおける、テキストの二次元的風景画メタファのように、良いメタファは他のメタファを生む要因である。

(4) 識別性

可視化された各対象間の識別性は、オリジナルの識別性と同じにする。ただし、大量の情報を同じ粒度で一度に表示することは、可視化の有効性を損なう。

この問題に関しては、魚眼ビュー的技法の導入が効果的である。

(5) 視覚特性

人間が一度に認識できる情報量は限られている。この問題に対して、焦点・文脈化アプローチ、特に魚眼ビューが有効である。

ただし、SemNetのようにユーザの視点が部分空間を移動する時、表示の変化が激しいと、ユーザに負担をかけることもあるので、実装にあたっては注意が必要である。

(6) メンタルモデル

ユーザの対象に対するメンタルモデルと、可視化によって提示される情報からユーザが構築するメンタルモデルの間に、ギャップが生じてはならない。

GalaxyOfNewsにおけるアニメーション効果は、ユーザがシステムの行っているということが認識しやすくなっている意味で、操作に関するシステムとユーザのメンタルモデルのギャップを埋めるのに有効である。

(7) 強調

情報の強調の意味をユーザが容易に理解できなければならない。強調されている対象と、されていない対象との違いに関する“意味付け”が明確である必要がある。

(8) グラフィック、意味

アイコンや操作用のグラフィックの意味が明確である必要がある。可視

化の表示様式の変化によって変化する操作に関しては、操作用のグラフィックに関しても明示的な変更が加わる必要がある。

(9) 一貫性

可視化された情報を通じて、ある対象に対して行う操作は、操作モードが同じであるかぎり一貫している必要がある。対象とする情報の入手や操作が一貫しない場合、ユーザの認知的疲労を招き、可視化がユーザの作業効率を低下させる。

(10) 簡素性

グラフィックによる表現は必要最低限にする。ユーザの操作に対するTAT(Turn Around Time)の良さは、可視化システムの重要な要素である。

小池らのシステムではフラクタルビューによる表示ノード数の制御により、表示されるノード数が削減し、ほぼ同程度の量のノードが表示される。このため、システムの反応速度が同程度に保たれる。

(11) アフォーダンス

対象に対する操作方法が、わかりやすい可視化情報表示である必要がある。

(12) 手がかり

対象に対する操作や、他の表示方法、他の特徴の表示方法などに関する手がかりに容易にアクセスできることが必要である。ThemeScapeにおいては、テキストの二次元的風景画に、与えられた領域における主題の強さを表すため、高さを表わす次元を導入し手がかりとしている。

6.3.3. 理解のしやすさ

(1) デフォルトの提供

表示方法などに関して、必ずデフォルトを用意する。またユーザがデフォルトの内容を理解できる必要がある。また、ユーザがデフォルトの変更を容易に行える必要もある。

(2) エラーへの対応

エラーが発生した場合に任意の場所に戻し、続行中の作業を中断できるようにする。その際、情報の破棄が発生する場合、システムによる確認を用意する。

(3) 寛容性

表示制御や対象操作にあたって、ユーザによる自由な制御を極力許可する。システムの都合によるユーザの操作制限の導入は、極力制限する。また、制限が存在する場合は、ユーザに提示する。

(4) カスタマイズ

ユーザによるデフォルトの内容のカスタマイズを容易にする。また、デフォルトへ戻すことを可能にする。

(5) 記憶負担の軽減

表計算のマクロのように分析のパスや操作手順などの保存を可能にする。また保存された手順の呼び出しを容易にする。

(6) 簡素性

グラフィックによる表現は必要最低限にする。多色表示はユーザの行う操作に対するTATを低下させ、それにより可視化された情報へのユーザのアクセスを阻害する。

(7) ユーザの位置の表示

ユーザが焦点を当てている対象や、それがおかれている文脈を適切に表示する。これらの情報表示では、文字で表示することの検討や、ヒストリー情報を表示する事なども考慮する。

(8) 階層構造

可視化された情報の表示では、表現できる限り、ネットワーク構造ではなく階層構造で表現する。またネットワーク構造で対象が表現される場合、部分化、記述要因の低減などにより、階層表現を行える操作パスを用意する。

階層構造の階層が深くなる、あるいは下位レベル表示要素が多くなる場合、フラクタルビューのような表示手段を導入する。

6.3.4. 可視化

(1) レイアウトの良さ

レイアウトの良さの評価は、ユーザによって異なる。一般的には必要最低限の簡素さを維持することを意識する。

(2) 一貫性

同一対象（特徴）の表示は、表示・操作モードの中で一定にする。たとえば、サルカーのシステムのように地図情報を扱うシステムでは、操作の対象の表示は、我々の地図認識という点で一貫している。

(3) 簡素性

必要最低限の簡素さを維持する。操作に関しても同様である。Powers of Ten Thousandのように、複雑な表示はかえってユーザの認識を悪化させる場合がある。

(4) コーディング

対象を、可視的エンティティにコーディングする規則は、同一対象（特徴）について、表示・操作モードの中で一定にする。

小池らのシステムでは、自己相似性を利用することにより、木のどのノードに着目していても統計的に同じような視点が得られ、常に着目ノードとその周辺を見ることができるように工夫されている。

(5) フォント/文字、図、アイコン

可視化されて表示された対象が、実際の何に該当するかを直感的に理解しやすくする。

アイコンの多用はかえってユーザの混乱を招く。意味不明、あるいは作成者の一人よがりなアイコンの使用は避ける。アイコンのデザインは、エンドユーザに対する認知的テストも含めて検討する。

また、フォントの強調の仕方は国や地域によって異なることを考慮する。

(6) 色

色彩の変化により対象の特徴を表現することが適切であると考えられる場合以外、なるべく少ない色数を用いる。

また、色彩の変化が有効である場合であっても、段階的変化がユーザの認知的処理に影響を与えない範囲で限定する。

6.3.5. 操作性

(1) 操作感

ユーザが行った操作に対するTATはなるべく短くする。操作の結果はすぐに表示する。

(2) 状態遷移の適切さ

ある操作モードから他の操作モードへの遷移、表示されている情報に対する操作によって引き起こされる状態の遷移は、ユーザが理解可能なようにする。また、このような状態遷移をユーザが容易に予測できるようにする。

(3) 効率

表計算のマクロのように分析のパスや操作手順などを保存し、再利用することにより操作量の低減を計る。また、コマンドに対するショートカットキーの導入を考慮する。

(4) フィードバック

ユーザが行った操作に対して、適切な反応を返す。また、反応が返せない場合、その理由、システムの状態などを表示する。

6.4. ユーザ・タスク・機能

4章で、ナレッジマネジメントという側面から見た場合に、ユーザと可視化に求められる機能について述べた。ナレッジマネジメントにおけるタスクと、ユーザの関係をまとめると表6-2になる。

タスク	可視化の機能	経営者	情報管理者	一般ユーザ	顧客
情報管理	構造化情報抽出機能(インデックスなどの構造化情報の抽出)		○		○
	データ概観機能(収集したドキュメントやデータの傾向を概観する)	○	○	○	○
	視点切替機能	○	○	○	○
仮説生成と検証	インタラクション機能(素早いインタラクションの確立)		○		
	条件変更機能(柔軟な条件変更)	○	○	○	
	モデル指示機能(ユーザが持っているメンタルモデルとシステムモデルとの違いを明示する)	○	○	○	
情報検索	概観機能	○	○	○	○
	ズーム機能	○	○	○	
	フィルタ機能	○	○	○	○
	オンデマンド詳細化機能	○	○		
	関連付け機能	○	○		
	履歴機能		○	○	○
	抽出機能		○		○

表 6-2 ナレッジマネジメントのタスクとユーザ

つぎにユーザの作業支援という観点に限定した可視化による支援と、タスクとの関係は表6-3のようにまとめられる。

		情報管理			仮説生成と検証			情報検索						
		構造化情報抽出機能	データ概観機能	視点切替機能	インタラクション機能	条件変更機能	モデル指示機能	概観機能	ズーム機能	フィルタ機能	オンデマンド詳細化機能	関連付け機能	履歴機能	抽出機能
構造化	冗長性	○	○					○	○	○	○	○		○
	一貫性	◎		◎	◎	◎	◎			◎	◎	◎		◎
情報入手の効率性	マッピング	○	○	○		○	○	○	○	○	○	○		○
	画面要素間の関係のわかりやすさ			◎	◎	◎							◎	
	メタファ	○	○					○	○	○	○	○		○
	識別性	◎	◎	◎		◎	◎	◎	◎	◎	◎	◎		◎
	可視化													
	視覚特性	◎	◎	◎				◎	◎	◎	◎	◎		◎
	メンタルモデル	○				○	○			○	○	○		○
	強調(グラフィックス、意味)	◎	◎				◎	◎	◎	◎	◎	◎		◎
	一貫性	○	○				○	○	○	○	○	○	○	○
	簡素性		◎			◎		◎	◎	◎	◎	◎	◎	◎
	アフォーダンス			○		○	○			○	○	○		○
	手がかり		◎	◎		◎	◎	◎	◎	◎	◎	◎		◎
理解のしやすさ	デフォルトの提供	○	○	○	○	○	○	○	○	○	○	○		○
	エラーへの対応(戻れる、中断できる、確認してくれる)	◎	◎	◎	◎	◎	◎							
	寛容性(ユーザによる自由な制御)	○			○	○	○							
	カスタマイズ	◎		◎	◎	◎	◎			◎	◎	◎	◎	◎
	記憶負担の軽減(再生よりも再認)	○			○	○				○	○		○	○
	ユーザ主体													
	システム状態表示													
	本質的な情報の表示													
	用語メッセージ													
	簡素性	◎	◎	◎	◎	◎	◎	◎	◎					
	ユーザの位置の表示	○	○	○	○	○	○	○	○				○	
	階層構造	◎		◎	◎									
可視化	レイアウトの良さ、美しさ	○	○	○				○						
	一貫性	◎	◎	◎	◎	◎	◎	◎		◎	◎	◎	◎	◎
	簡素性	○	○	○	○		○	○	○	○	○		○	○
	コーディング	◎	◎	◎				◎	◎	◎	◎	◎		◎
	フォント/文字、図、アイコン	○	○	○				○	○	○	○	○		○
	色	◎	◎					◎						
	動画、音													
操作性	身体的負担軽減													
	操作感(反応速度等)	○	○	○	○	○		○	○					
	システム側の状態遷移の適切さ	◎		◎	◎	◎	◎						◎	
	学習の進捗状況への対応(習熟度)													
	効率(操作の軽減)				◎	◎								
	フィードバック	○	○	○	○	○	○	○	○	○	○	○		○
	音													

表 6-3タスクと可視化

7. 結語

本報告書では、インターネット上の情報収集、特にドキュメントオーサリングを可視化によって支援するためには、いかなる機能を可視化技術が提供できるかという視点で、既存技術の調査と分析を行った。その結果として、インターフェースモデルの提案を行い、さらにそのインターフェースモデルに関して、ユーザ - タスク - 機能といった視点で対応付けを行った。

本事業成果の今後の進展であるが、可視化システムをツールド・アーキテクチャとして捉え、その基盤となる可視化用ライブラリとして、インターフェースモデルで述べた機能を実現したツール群を整備して行くことが考えられる。

本事業では、可視化適用の範囲を限定して作業を行った。その限定された範囲であっても、可視化システムのレパートリーは多く、今後、さらに多様化するものと予想される。現時点においては、可視化で求められる基本機能や既知機能を整理してツールとして適用し、その上にさらに高度な機能の構築を試みるといったアプローチが、妥当であろうと考える。

関連図書

[1] R.Hackathorn., Data Delivery When You Want It, BYTE, June 1997, pp. 51. http://www.opengl.org/

[2] http://www.nk-exa.co.jp/~andoh/vrml/

[3] http://chasen.aist-nara.ac.jp/

[4] http://kakasi.namazu.org/

[5] http://www.spss.com/

[6] http://www.sas.com/

[7] ニューロンネットワークグループ, 桐谷茂. 入門と実習 ニューロコンピュータ, 技術評論社

[8] T. Kohonen, Self-Organization Maps, Springer-Verlag Berlin Heidelberg New York, 1995.

[9] http://www.vogue.is.uec.ac.jp/~koike/papers/bit/bit.html

[10] http://websom.hut.fi/websom/

[11] S.Chakrabarti, M.Dom, S. Kummar, P. R. Ravi,S. Rajagopalan, , A. Tomkins,J. M. Kleinberg, and D.Gibson, ハイパーリンクを賢く使う 日経サイエンス, PP.28-35, 1999.

[12] K. tajima,K. Hatano,T. Matsukur, R. Sano, and K. Tanaka,K, Discovery and Retrieval of Logical Information Units in Web, 1999 ACM Digital Library Workshop on Organizing Web, pp.13-23, 1999

[13] G. G. Robertson and J. D. Mackinlay. The document lens. in Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'93), pp. 101-108. 1993.

[14] Stuart K. Card , Jock D. Machinlay and Ben Shineiderman. Information Visualization. Morgan Kaufmann Publishers

[15] M. Sarkar and M. H. Brown. Graphical fisheye views of graphs. in Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'92), pp. 83-91. 1992.

[16] 三末和男, 杉山公造. 図的思考支援を目的とした図の多視点遠近画法について.情報処理学会論文誌, Vol. 32, No. 8, pp. 997-1005, 1991.

[17] E.G.Noik. Layout-independent fisheye views of nested graphs. in Proceedings of the 1993 IEEE Symposium on Visual Languages, pp. 336-341. 1993.

[18] G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone Trees: Animated 3D visualizations of hierarchical information. in Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'91), pp. 189-194. 1991.

[19] H. Koike and H. Yoshihara. Fractal approaches for visualizing huge hierarchies. in Proceedings of 1993 IEEE/CS Symposium on Visual Languages (VL'93), pp. 55-60. 1993.

[20] K. Perlin and D. Fox. An alternative approach to the computer interface. in Proceedings of 1993 ACM SIGGRAPH Conference, pp. 57-64, 1993.
[21] H. Lieberman. Powers of ten thousand: navigating in large information spaces. in Proceedings of the ACM, 1 Symposium on User Interface Software and Technology (UIST'94), pp. 15-16994.
[22] G. Salton, J. Allan, C. Buckley and A. Singhal. Automatical Analysis, Theme Generation, and Summarization of Machine Readable Text, Science, Vol.264, No.3, pp.1421-1426, 1995.
[23] X. Lin, Visualization for the Document Space, IEEE Visualization'92 Conference, Boston, pp.274-281, 1992.
[24] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur and V. Crow, Visualizing the Non-Visual: Spatial Analysis and Interaction with Information form Text Documents, IEEE Symposium on Information Visualization, New York, pp.51-58, 1995.
[25] E. Rennison, Galaxy of News: An Approach to Visualizing and Understanding Expansive News Landscapes, ACM Symposium on User Interface Software and Technology, New York, pp.3-12, 1994.
[26]http://multimedia.pnl.gov:2080/infoviz/technologies.html#starlight
[27] S. G. Eick, J. L. Steffen and E. E. Sumner, Sessoft ? A Tool forVisualizing Line Oriented Software Statics, IEEE Transactions on Software
Engineering, Vol.18, No.11, pp.957-968, 1992.
[28] T. A. Ball and S. G. Eick, Software Visualization in the Large, IEEE Computer, Vol.29, No.4, pp.33-43, 1996.
[29] 大柴崇, 田中次郎, 3次元モデリングツール Claymore-負荷情報によって強化された直接操作, 日本ソフトウェア科学会第15回論文集, pp.161-164, 1998.
[30] J. Rubin, Handbook of Usability Testing, Wiley, 1994.
[31] 山岡 俊樹,鈴木 一重,藤原 義久. 構造化ユーザインターフェースの設計と評価. 共立出版株式会社

— 禁無断転載 —

平成13年3月発行

発　行　　財団法人　データベース振興センター
東京都港区新橋2丁目13番8号
新橋東和ビル5階
TEL　０３－３５０８－２４３０

委託先　　株式会社日本総合研究所
東京都千代田区一番町16番
TEL　０３－３２８８－４７６１

印刷所　　株式会社カントー
東京都千代田区九段北１－１１－２
TEL　０３－３２３８－６０１１