

# 「データ宝石箱」による大規模データの視覚化

## Large-Scale Data Visualization Using “Data Jewelry-Box”

伊藤 貴之      山口 裕美

日本アイ・ビー・エム(株) 東京基礎研究所

〒242-8502 神奈川県大和市下鶴間 1623-14

Takayuki ITOH      Yumi YAMAGUCHI

IBM Research, Tokyo Research Laboratory

1623-14 Shimotsuruma, Yamato-shi, Kanagawa, 242-8502 Japan

E-mail: {itot,yyumi}@trl.ibm.com

This paper introduces our hierarchical data visualization technique, “Data Jewelry-Box,” and shows its some applications. Our technique represents hierarchical data as a set of nested rectangles. It applies a novel packing algorithm that places thousands of rectangles in nearly minimized display spaces in only several seconds. The paper also shows the applications of the technique for the visualization of Web sites, processes of distributed computers, and tree structure of keywords in large-scale document data.

**Keywords : Hierarchical data visualization, Rectangle packing, Delaunay triangular mesh, Web site visualization, Distributed system visualization, Text data visualization.**

### 1. はじめに

例えば、計算機のファイルシステム。例えば、大会社の人事組織。例えば、Yahoo に代表されるカテゴリ分類型のウェブサイト。身のまわりには、階層構造で整理されたデータは非常に多く存在する。これらの階層型データを対話的に表現する GUI の大半は、まず上位階層を表示し、ユーザーの選択操作によって徐々に局所的に下位階層を表示させるように構築されている。これらの GUI とは逆に、最初から階層型データ全体を一画面に展開して表示することで、データの分布を一目で理解するような視覚化手法はないものか？ ちょうど宝石店のショーケースのようにデータ全体を見渡せる視覚化手法はないものか？ ... 著者らはこのような発想から、大規模階層型データ視覚化手法「データ宝石箱」を開発した。

「データ宝石箱」は、入れ子状に配置された長方形によって階層型データを表現する。図 1 にその実用イメージの例を示す。画面空間を有効活用するために、本手法では長方形をできるだけ隙間なく配置することで、占有空間を小さく抑えている。

本報告では、まず 2, 3 章にて、「データ宝石箱」[1-3]のアルゴリズムを概観する。続いて 4 章にて、「データ宝石箱」の適用例として、ウェブサイトのアクセス分布[3, 4]、分散計算環境におけるプロセス分布、大規模テキストデータにおけるキーワード群の構成、の各種データに対する視覚化の実行例を示す。5 章にて、関連研究との比較について述べ、6 章にて本報告の結論と課題を述べる。



図 1. 「データ宝石箱」の実用イメージ。(左)46 人の人物をグループ分けした分布図の表示。(右)計算機ファイルシステムのディレクトリ階層を入れ子状に表示したもの。

## 2. 「データ宝石箱」のアルゴリズム概要

まず「データ宝石箱」における階層型データ全体の配置手順について説明する(図 2 参照)。本手法では、まず最下位階層を構成するノードをアイコン(図 2 の場合は正方形)で表現して、それを隙間無く配置する。続いて、これを含む長方形を生成する。続いて、これらの長方形群を隙間無く配置し、同様にこれを含む長方形を生成することで、上位階層を表現する。以上の処理を、最下位階層から最上位階層に向けて反復することで、データ全体の配置を決定する。

ここで「データ宝石箱」では、以下の条件をできるだけ満たすように長方形の配置場所を決定する。

[条件 1] 隣接長方形との干渉がまったくない位置に長方形を配置する。

[条件 2] 長方形の配置領域を囲む占有領域が最小となる位置に長方形を配置する。

上記の 2 条件を満たすように長方形を配置する問題は、占有面積を最小化するように入力形状群を配置する最適化問題として、古くから知られている。この問題は主に、VLSI 回路の基板配置、板金や服飾型紙への部品配置などの用途で実用されている。これらの用途では、遺伝子アルゴリズムなどの最適化手法を用いて部品の配置を実現している例が多い。しかし、数分～数時間の計算時間を要することが多く、対話的操作を要する視覚化の分野には向かない。著者らは、最適解でなくてもいいから、ある程度良好な配置結果を、短時間に算出する手法を導入することにした。詳細については 3 章にて述べる。

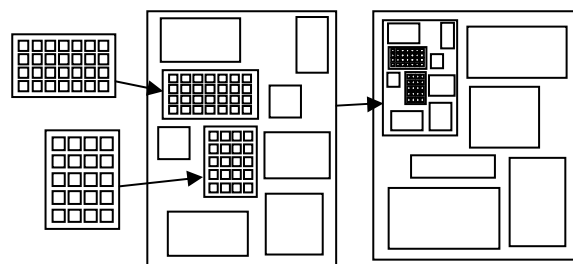


図 2 階層型データの画面配置順。まず最下位階層を配置し、続いて上位階層に向かって配置処理を反復する。

著者らが最近提案した「データ宝石箱 II」では、ノードの位置情報を記録するテンプレートを導入している[2]。この手法では、前述の[条件 1][条件 2]だけでなく、下記の[条件 3]も同時に満たすように長方形を配置する。

[条件 3] テンプレートに記述された位置にできるだけ近い位置に長方形を配置する。

ここで「データ宝石箱 II」にテンプレートを導入する意義を説明する（図 3 参照）。

例えば、「このデータは画面の左上に置きたい」「このデータは真ん中に置きたい」というようにデータの配置をユーザーがデザインしたいことがあるとする。このようなときには、ユーザーのデザイン結果として得られるノードの位置をテンプレートに記述してから、本手法でノードを配置することで、ユーザーのデザイン意図をノード配置に反映することができる。

また、「新しいデータから順に左から」「アルファベット順に上から」というように、配置画面の座標軸に何らかの意味を持たせてデータを配列させたいときがあるとする。このようなときには、各ノードの座標値を算出したものをテンプレートに記述してから、本手法でノードを配置することで、何らかの意味に沿って左右または上下に並んだ配置結果を得ることができる。

また、時系列に沿って微量ずつ変化するデータを、できるだけ類似した表示結果で見比べたいときがあるとする。このようなときには、直前の時刻における配置結果をテンプレートに記述してから、本手法でノードを配置することで、シームレスな配置の変化を実現できる。

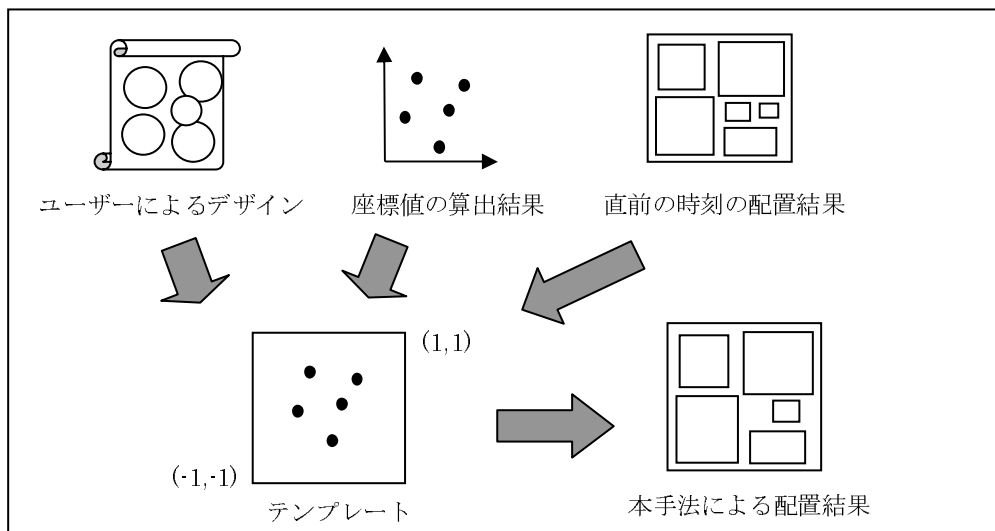


図 3：「データ宝石箱 II」のイメージ。テンプレートに格納した座標値を参照して長方形群を配置する。

### 3. 「データ宝石箱」に用いられる長方形配置アルゴリズム

図 2 からわかるように、「データ宝石箱」では、階層型データ中の 1 階層は長方形の集合で表現される。本章では、[条件 1][条件 2][条件 3]をできるだけ満たし、かつ高速に、長方形の集合を画面空間に配置するアルゴリズムを紹介する。

「データ宝石箱」では、階層型データを表現する長方形群を、1 個ずつ漸進的に画面空間に配置する。この際に、長方形を重ねることなく配置できる隙間を高速に探索して、その位置に長方形を配置する。適切な配置場所を高速に探索するために、すでに配置されている長方形群の中心点、および長方形群の占有領域を囲む長方形領域の 4 頂点、を連結する Delaunay 三角メッシュを用いる（図 4 参照）。

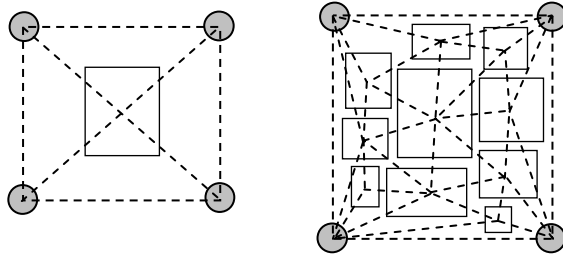


図 4 (左) 三角メッシュ(点線)の初期状態。面積が最大である長方形だけが配置された状態である。(右)すでに配置された長方形の中心点群、占有領域の4頂点の中心点、を連結する三角メッシュ。

本アルゴリズムの手順を図 5 に示す。またアルゴリズム中の各処理について、以下に解説する。詳細については文献 1)2)を参照していただきたい。

1. 長方形の配置順番を決定する。
2. 未配置である長方形を、順番に従って 1 個抽出し、2.1~2.3 の処理を行う。
  - 2.1 三角形の処理順番を決定する。
  - 2.2 未処理である三角形を、順番に従って 1 個抽出し、2.2.1~2.2.2 の処理を行う。
    - 2.2.1 三角形内部に、長方形の候補位置をいくつか算出する。
    - 2.2.2 各々の候補位置について、長方形の配置場所としてのふさわしさを評価する。
  - 2.3 長方形の配置場所を決定し、三角メッシュを更新する。

図 5. 1 階層を構成する長方形群の配置手順。

### 3.1 長方形の配置順番

本アルゴリズムでは、長方形を 1 個ずつ漸進的に画面に配置する。長方形の配置順番について以下に説明する。

「データ宝石箱」では、長方形を面積でソートして、面積の大きい順に長方形を配置する (図 6(左)参照)。[条件 2]を重視する観点から、まず大きな長方形を配置し、続いて小さな長方形を配置できる隙間を探すことで、占有領域の拡大を抑えている。

「データ宝石箱 II」では、まず面積が最大である長方形を配置し、続いてその長方形に距離が近い順に残りの長方形を配置する (図 6(右)参照)。[条件 3]を重視する観点から、隣接順に長方形を配置することで、長方形間の位置関係を保持している。

### 3.2 三角形の探索順番

1 個の長方形を配置する際に、本アルゴリズムでは Delaunay 三角メッシュを構成する三角形を 1 個ずつ探索しながら、長方形の配置にふさわしい位置を探し出す。三角形の探索順番について以下に説明する。

「データ宝石箱」では、三角形を大きさでソートして、大きい順に三角形を処理する（図 7(左)参照）。[条件 2]を重視する観点から、大きな三角形になるほど内部に隙間がある可能性が高い、というヒューリスティックに基づいて処理順を決定している。三角形の大きさの基準には、面積、内接円半径、外接円半径などを用いることができる。

「データ宝石箱 II」では、まずテンプレートに記述された座標値を包括する三角形を抽出し、その三角形を処理する。続いてその三角形を出発点にして、幅優先探索によって隣接三角形を探索し、テンプレートに記述された座標値から一定距離内である三角形を対象として処理を進める（図 7(右)参照）。[条件 3]を重視する観点から、テンプレート座標値に近い三角形から順に処理している。

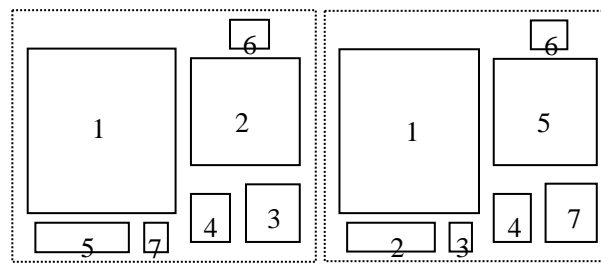


図 6. (左)「データ宝石箱」における長方形の配置順番。(右)「データ宝石箱 II」における長方形の配置順番。

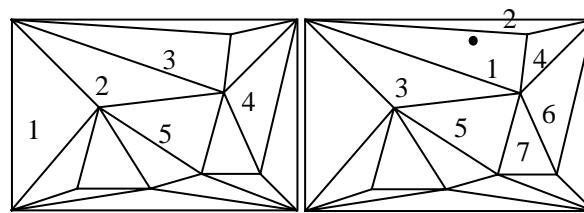


図 7. (左)「データ宝石箱」における三角形の処理順番。(右)「データ宝石箱 II」における三角形の処理順番。

### 3.3 長方形の候補位置の算出

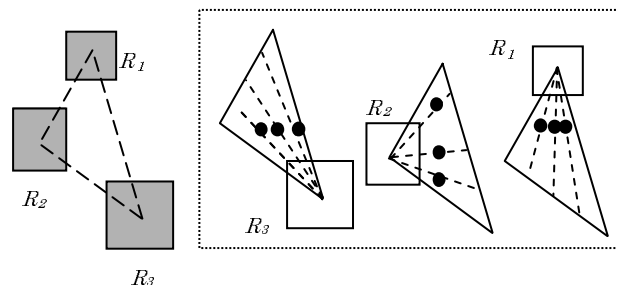


図 8. 長方形の候補位置の決定。左図のような三角形に対して、右図の黒丸が候補位置となる。

1 個の三角形を探索する際に、本アルゴリズムでは三角形内部に複数の候補位置を算出し、その中から長方形の配置にふさわしい位置を選択する。候補位置の算出方法について以下に説明する。

長方形の候補位置を算出するために本手法では、まず三角形の各辺を  $n$  等分に分割する。続いて三

角形の各頂点について、対辺の  $n$  等分点と連結する  $(n-1)$  本の線分を生成する。この各々の線分上で、三角形の頂点上にすでに配置された長方形と接する位置を、これから配置する長方形の候補位置とする（図 8 参照）。

著者らの実験では、「データ宝石箱」では、 $n$  の値を  $n=2$  に固定しても、十分な配置結果が得られている。一方「データ宝石箱 II」では、十分な配置結果を得るためには  $n$  の値を適応的に変更する必要がある。そのため「データ宝石箱 II」のほうが、候補位置が多くなり、そのぶん計算時間も増加する傾向にある。

### 3.4 候補位置の評価式

本アルゴリズムでは各々の候補位置に対して、長方形の配置へのふさわしさを評価値として算出し、評価値が最も良好である位置を長方形の位置として選択する。評価値の算出式について以下に説明する。

「データ宝石箱」では、ある候補位置に長方形を配置したことによる占有領域の拡大量を  $S$  とし、候補位置ごとにこれを算出する。 $S$  の値には、占有領域の面積や外周長などを用いることができる。[条件 1] を満たし、かつ  $S=0$  である候補位置が 1 個見つかったら、図 5 の処理 2.2 を終了し、その位置に長方形を配置する。[条件 1] を満たし、かつ、 $S=0$  である候補位置が存在しない場合には、[条件 1] を満たす候補位置の中で  $S$  の値が最小である位置に長方形を配置する。

「データ宝石箱 II」では、占有領域の拡大量  $S$  の他に、テンプレートに記述された座標値と候補座標値との距離  $D$  を導入して、 $aD+bS$  を候補位置の評価式とする。ここで  $a, b$  はユーザーが指定する正値の定数である。[条件 1] を満たす候補位置の中で  $aD+bS$  が最小である位置を特定し、この位置に長方形を配置する。

### 3.5 三角メッシュの更新

長方形の位置を決定したら、本アルゴリズムでは以下の方法によって三角メッシュを更新する。

$S > 0$  である位置に長方形を配置する場合、いま配置する長方形は三角メッシュの四隅をはみだしてしまう。この場合には、まず長方形の占有領域を囲む長方形の 4 頂点のうちいずれかを移動し、いま配置する長方形も包括するような占有領域を作成する。

続いて、いま配置する長方形の中心点を三角メッシュに追加する。まず中心点を包括する三角形の 3 頂点を、その中心点と連結する。続いて、Delaunay の条件を満たすように三角メッシュの辺を入れ替える（図 9 参照）。

このように Delaunay 条件を満たす三角メッシュを使い続けることで、細長い三角形の生成を低減し、効率よく長方形の候補位置を算出する。

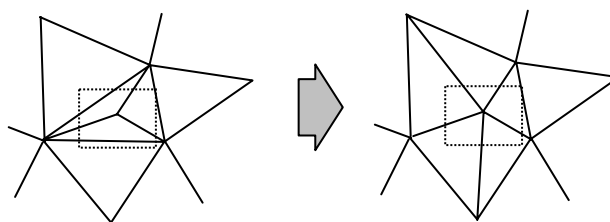


図 9. (左) いま配置した長方形の中心点を三角形の 3 頂点に連結した状態。(右) 三角形の辺

を局所修正して、三角メッシュを更新した状態。

## 4. 適用例

本章では、「データ宝石箱」を用いた情報可視化の事例をいくつか紹介する。

### 4.1 ウェブサイトのアクセス分布の視覚化

著者らはすでに、「データ宝石箱」をウェブサイトのアクセス分布の視覚化に適用した事例を報告している [3] [4]。本節では、その概要と画像表示例について紹介する。また図 10 に、本事例の処理の概要を示す。

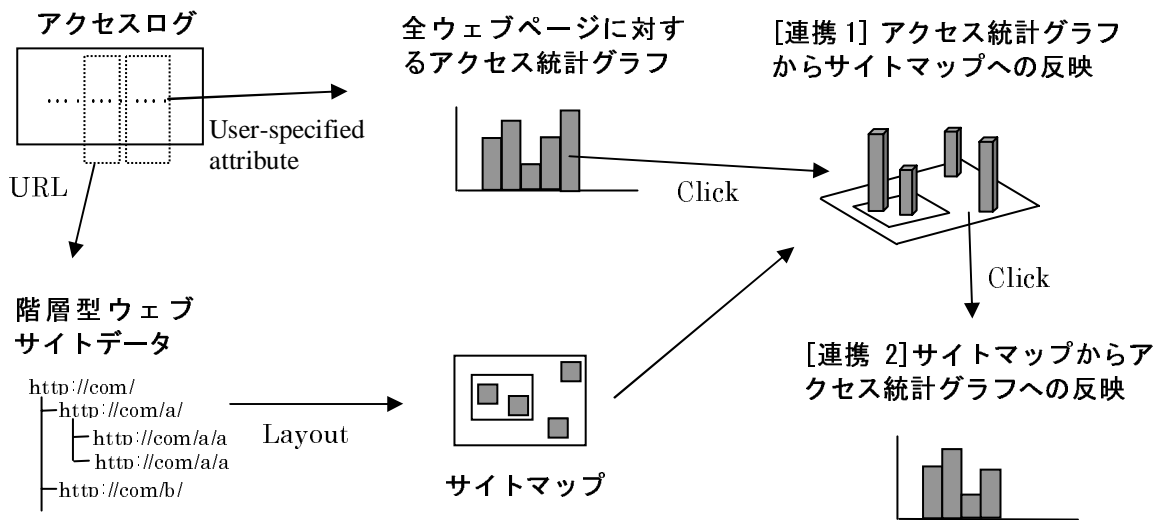


図 10. ウェブサイトのアクセス分布視覚化事例の処理の概要。

### 入力データ

著者らが報告した視覚化事例では、ウェブサーバーに蓄積されるアクセスログファイルを入力データとした。アクセスログには、サイト閲覧者がウェブページにアクセスしたときの、1)サイト閲覧者の IP アドレス、2)認証を行った際のユーザー名、3)アクセス日時、4)リクエスト内容、5)ステータス、6)転送バイト数、7)アクセスされたページの URL、8) 7)のリンク元の URL、が記録されている。

### サイトマップの自動生成

ウェブサイトの構造を表現するサイトマップを自動生成するために、本事例ではアクセスログから URL を抽出して、そのディレクトリ階層を参照してウェブページの階層型データを構築する。

続いて本事例では、「データ宝石箱」を用いて上記の階層型データを画面配置する。本事例では、ウェブページをアイコンで、ディレクトリを入れ子状の箱で表現することで、ウェブサイトのサイトマップを表現する。

### アクセス統計グラフの表示

本事例では、サイト閲覧者のアクセス傾向を視覚化するために、アクセスログからユーザーの指定した属性を基準にしてアクセス数を集計した棒グラフを作成する。本報告ではこの棒グラフを、アクセス統計グラフと呼ぶ。アクセス統計グラフは、属性に基づいた分類を横軸に、アクセス数を縦軸に

とる。アクセスログの属性のうち、1)、7)、8)に関しては、アクセス数の多い順に一定数（著者らの実装では30項目）を、左から並べて表示する。

#### 連携1: アクセス統計グラフからサイトマップへの反映

アクセス統計グラフで、ユーザーが関心のある項目をクリックすると、サイトマップ上で、指定された項目に関するアクセスがあったウェブページのアイコンをハイライトする。このときアイコンに高さを与えることで、個々のページのアクセス数を表示する。このようにして、アクセスログの特定の属性に対するアクセス分布を、サイトマップ上で視覚化できる。

#### 連携2: サイトマップからアクセス統計グラフへの反映

サイトマップで、ユーザーが関心のあるウェブページのアイコンをクリックすると、そのページへのアクセスを集計した統計グラフが表示される。これによりユーザーは、サイト全体のアクセス傾向だけでなく、特定のページに対するアクセス傾向も知ることができる。

ここで、約2000ウェブページを公開しているウェブサイト为题材として、1週間のアクセスログから得られるアクセス傾向を視覚化した実験例を示す。ウェブページに対するアクセスだけを視覚化の対象にするために、本実験ではまず前処理として、URL が画像ファイルやスタイルファイルなどを示している行を、アクセスログから削除した。

このアクセスログを日付で分類集計して、横軸が日付、縦軸がアクセス数を示すアクセス統計グラフを作成した（図11(左)参照）。さらに、このアクセス統計グラフを構成する7本の棒グラフを、1時間単位で分割して24色に色分けした。

同様に、アクセスログをステータスで分類集計したアクセス統計グラフを作成した(図11(右)参照)。このグラフでも、棒グラフを1時間単位で分割して24色に色分けした。

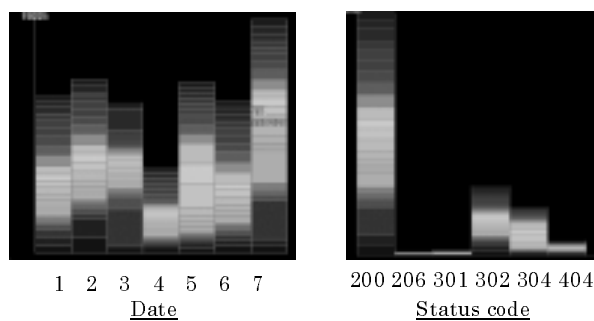


図11 (左)日付で分類集計したアクセス統計グラフ。(右)ステータスで分類集計したアクセス統計グラフ。

図11(左)より、このウェブサイトでは最終日のアクセス数が最も高いことがわかった。ここで[連携1]を用いて、最終日の棒グラフのある1時間をクリックして、サイトマップ上でその1時間のアクセス分布を表示した(図12参照)。このアクセス分布から、なぜ最終日のアクセスが多かったのか、その背景を探索した。

図12の右上部に描かれた円の内部に、ほとんどのページがアクセスされたディレクトリを発見した。[連携2]を用いて、これらのページへのアクセスをサイト閲覧者のIPアドレスで分類して表示すると、すべてのページに同一IPアドレスからのアクセスがあることがわかった。これらの結果から、あるディレクトリのファイルをすべて見ている熱心なサイト閲覧者が存在していたことがわかった。



また、図 12 の右下部に描かれた円の内部に、非常にアクセスの多いページを発見した。[連携 2]を用いて、そのページへのアクセスを対象として、リンク元の URL で分類したアクセス統計グラフを表示した。すると、ある新聞会社の URL からのリンクによって訪れているサイト閲覧者が多いことがわかった。リンク元である新聞会社にアクセスしてみると、新聞会社のオンラインニュースにそのページが紹介されていたことがわかった。

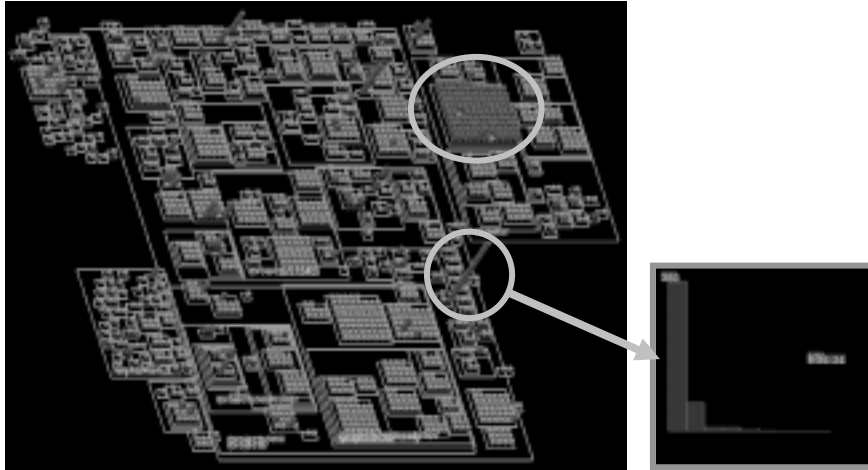


図 12. [連携 1]を用いて、アクセス統計グラフ中のある 1 時間をクリックし、その時間のアクセス分布をサイトマップ上で表現したもの。さらに[連携 2]を用いて、サイトマップ上のウェブページのアイコンをクリックすると、そのウェブページに対するアクセス統計グラフを表示できる。

従来のウェブアクセス分析ツールでは、棒グラフ、折れ線グラフ、ランキング表、などといった単純な表現だけで、ウェブサイト全体の概略的なアクセス傾向を表現することが多い。このような概略的な表現方法だけでは、必ずしもすべてのアクセス傾向を発見できるとは限らない。「データ宝石箱」を用いてウェブページ単位でアクセス数を表現することにより、詳細なアクセス傾向の発見とその背景の分析を支援することができるので、ウェブアクセス分析ツールの改善に寄与できると考えられる。

例えば本事例で[連携 1]を用いて、ウェブサイト全体のアクセス分布を視覚化することで、特徴的なアクセス傾向をもつウェブページを発見することができた。例えば図 12 の右上部のディレクトリのように、個々のウェブページへのアクセスは決して多くないながらも、ディレクトリ内のすべてのウェブページにアクセスされているというような、地味ながら興味深いアクセス傾向を発見することができた。このようなアクセス傾向の発見において、本事例は従来のツールに比べて有利であるといえる。

さらに[連携 2]を用いて、そのアクセス傾向の背景を知ることができた。例えば図 12 の右下部のウェブページのように、ユーザーが関心のある任意のウェブページのアクセス傾向を表示することができた。本事例はすべてのウェブページのアクセス数表示を一画面で実現しているため、任意のウェブページのアクセス傾向を、1 クリックで直接探索することが可能である。この点において本事例は、従来のツールに比べて有利であるといえる。

## 4.2 分散計算環境におけるプロセス分布の視覚化

著者らは現在、「データ宝石箱」を分散計算環境のモニター画面として用いる実験をしている。その一例として、分散計算環境におけるプロセスの推移をリアルタイム表示する視覚化ツールの事例を示す。

本事例では、Grid Computing のフレームワークである OGSA (Open Grid Service Architecture) の API を用いてプロセスを分散した。図 13(上)に示す通り、本事例ではサーバー群の窓口にあたる親サービスを用意し、クライアントからのジョブを受け付ける。親サービスはそのジョブを複数のプロセスに分割し、複数の子サービスに割り当てる。このとき本事例では、図13(下)に示すように、プロセスを未処理プロセスと処理済プロセスに分類し、さらに処理済プロセスをサービスごとに分類することで、プロセスの推移を階層型データとして表現する。この階層型データの時間変化を逐次的に画面表示することで、プロセスの推移を視覚化する。

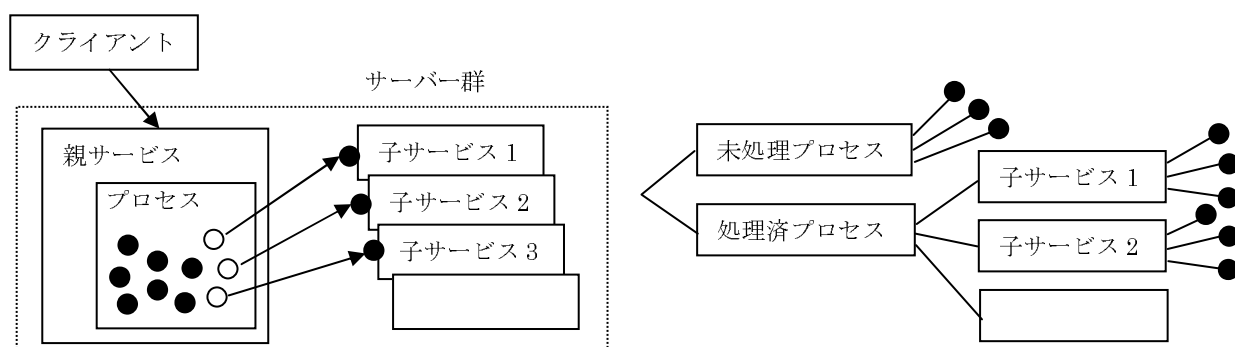


図 13. 本事例における分散計算環境のプロセス分布。(上)システム構成。(下)プロセス推移を階層型データとして表現したもの。

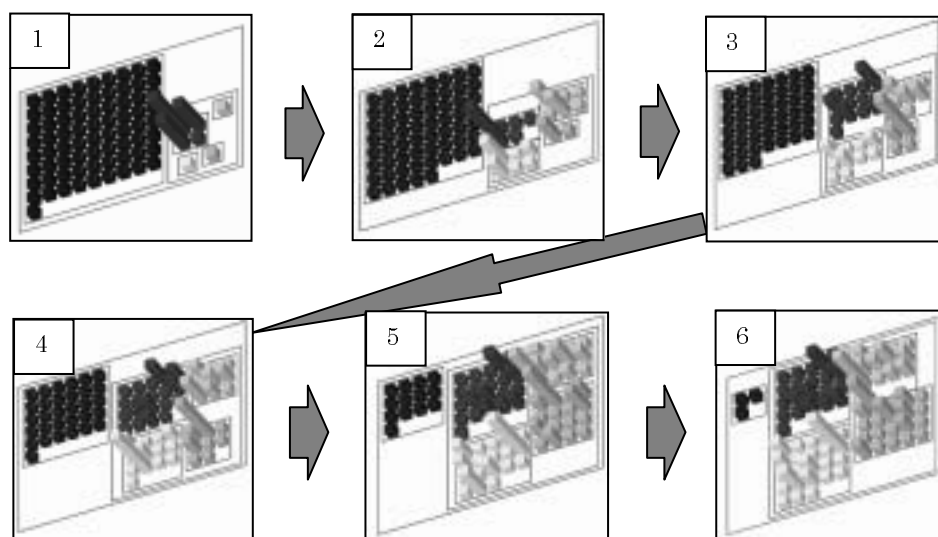


図 14. 本事例によるプロセス分布の視覚化結果。

本事例によるプロセス分布の視覚化結果を、図 14 に 6 コマの画像で示す。図中の左側にある青い棒が未処理プロセスを表しており、右側にある 4 色の棒が 4 個のサービスに割り振られたプロセスを表

している。各々の棒グラフの高さは、各プロセスの所要時間を示している。本事例では「データ宝石箱 II」を用いることにより、逐次的に推移するプロセス分布をシームレスに画面表示している。また、階層型データの画面配置 1 回の計算時間は 1 秒以下であり、リアルタイムなプロセス監視に十分利用できることがわかる。

#### 4.3 大規模テキストデータにおけるキーワードの視覚化

私たちの周囲には、多様な大規模テキストデータが存在する。例えばニュース記事、ウェブサイト、図書情報、論文データベース、実験データ、医療データ、などがこれに相当する。これら大規模テキストデータの有力な分析手段として、キーワードの抽出と構造化があげられる。

大規模テキストデータから得られるキーワード群を、階層型データとして構造化するケースは多い。例えば、大分類を表すキーワードを上位階層とし、その下に中分類を表すキーワード、さらにその下に小分類を表すキーワード...というようにして階層構造を構築することができる。あるいは、あるキーワードを出発点にして、それと相関性の高いキーワード群を連結し、さらにそれらと相関性の高いキーワード群を連結し...というようにして有向グラフを構築することができる。

構造化されたキーワード群を視覚化する試みは、例えば文献[7]などで報告されている。文献[7]に報告された拡張 ConeTrees に代表されるような、キーワード群を線で連結する表現形式による視覚化手法は、キーワード群の相関性や親子関係の視覚化に適しているといえる。

「データ宝石箱」も同様に、構造化されたキーワード群の視覚化に用いることができる。「データ宝石箱」では、2 次元的なデータ配置により高さ方向を属性値（例えばキーワードの出現頻度、最新出現日、各種スコアなど）の表現に用いることができるので、構造と属性値の相関性の視覚化などに向いているといえる。

図 15 に、階層型データとして構造化された大規模キーワード群の視覚化の例を示す。今後の課題として、この視覚化結果からどのような知見を得ることができるか検討したい。

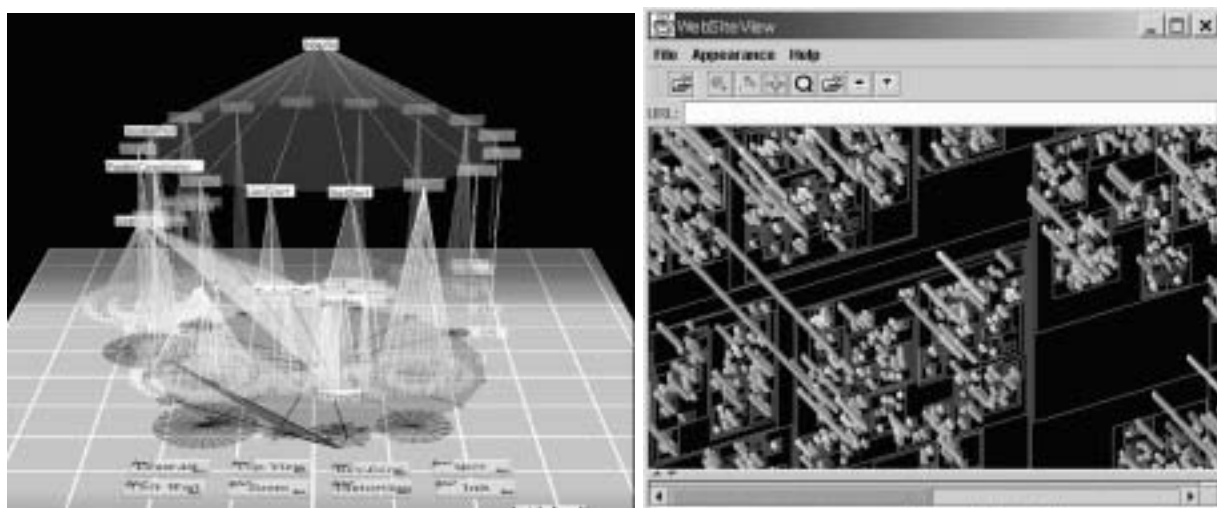


図 15. 大規模キーワード群の視覚化の例。(左)拡張 ConeTrees による視覚化例。(右)「データ宝石箱」による視覚化例。

#### 5. 関連研究

本報告に関連する研究分野として、階層型データの視覚化手法をいくつか紹介する。

階層型データの視覚化手法の多くは、階層型データを木構造として図示している。大規模データへの対応や、ナビゲーションなどの操作性などの観点から、Hyperbolic Tree[5], Cone Tree[6, 7], Fractal Tree[8], などの多くの研究が報告されている。

帯グラフや円グラフを入れ子構造にすることで、階層型データの統計的特徴を表現する手法もいくつか報告されている[9, 10]。その代表的な手法である Treemap[9]の拡張手法として、単純な帯グラフの入れ子構造の代わりに、2次元空間の長方形分割によって階層型データを表現する手法が報告されている[11, 12]。また、近年報告された Quantum Treemap[13]は、著者らが試みているような階層型データのアイコン表示にも適している。

「データ宝石箱」は2次元の入れ子構造で階層型データを表現しているが、3次元の入れ子構造を半透明表示することで階層型データを表現する手法もいくつか提案されている[14, 15]。

本報告に関連する別の研究分野として、ウェブサイトの視覚化手法をいくつか紹介する。

ウェブサイトを視覚化する手法の多くは、ウェブページ間のリンクをアークとするグラフ構造の視覚化手法を適用している[16-18]。グラフデータの視覚化手法自体の技術的な難易度が高いこともあって、これらの手法は実用化が進んでいないように思われる。

ウェブサイトを階層型データとして視覚化する手法もいくつか報告されており、すでに実用化も進められている[19, 20]。これらの手法は階層型データを木構造として図示していることから、「データ宝石箱」によるウェブサイト表現とは指向が異なる。

ウェブへのアクセス結果の視覚化手法は、個人のアクセス経路の視覚化という観点では多くの研究が報告されている[21-23]。しかし、本報告で示したようなウェブサイト全体のアクセス傾向という観点からの視覚化手法は見当たらない。

分散計算環境の視覚化、テキストデータ群のキーワードの視覚化、に関するサーベイは今後の課題としたい。

## 6. まとめ

本報告では、著者らがすでに提案している階層型データの視覚化手法「データ宝石箱」の概要を紹介し、その適用例を示した。また本報告により、「データ宝石箱」について以下のような特徴を示すことができた。

- 数百、数千、あるいはそれ以上の規模のデータを一画面に表現できるので、データのもつ局所的な傾向や詳細な傾向を発見しやすい。
- 画面配置に要する計算時間が小さく、リアルタイムな処理に向いている。
- データは2次元的に配置されるので、高さ方向の座標軸を用いてデータの属性を表現するのに向いている。また3次元CGのスキルや装置が必要ない。

第4章で示した適用例以外にも、「データ宝石箱」は多様なデータの視覚化に適用できる。例えば

- 計算機のファイルシステムや、会社の人事組織など、典型的な階層型データ。
- 顧客情報、経理情報、自治体情報、その他いわゆるビジネス系のデータ。
- 遺伝子などの大規模情報のクラスタリング結果。

などの視覚化に有用であると考えられる。

## 謝辞

「拡張 ConeTrees 技法」の画像例をご提供くださった、お茶の水女子大学大学院藤代一成教授および山下由美氏に感謝します。また有益な助言を下さった、日本アイ・ビー・エム（株）東京基礎研究所研究員諸氏、および東京大学堀井秀之教授に感謝します。

## 参考文献

- [1] 伊藤, 梶永, 池端, データ宝石箱: 大規模階層型データのグラフィックスショーケース, 情報処理学会グラフィクスと CAD 研究報告, 2001-CG-104, 2001.
- [2] 山口, 伊藤, データ宝石箱 II ~ 位置情報テンプレートを用いた大規模階層型データのグラフィックスショーケース, 情報処理学会グラフィクスと CAD 研究報告, 2002-CG-108, 2002.
- [3] 山口, 伊藤, 池端, 梶永, サイトマップ表示とアクセス統計表示の連携によるウェブサイト視覚化ツール, 情報処理学会論文誌投稿中.
- [4] 山口, 池端, 伊藤, 梶永, データ宝石箱を用いたウェブアクセスログの視覚化, 第 30 回可視化情報シンポジウム, 2002.
- [5] Lamping J., et al., The Hyperbolic Browser: A Focus+context Technique for Visualizing Large Hierarchies, *Journal of Visual Languages and Computing*, 7, 1, pp. 33-55, 1996.
- [6] Carriere J., et al., Research Paper: Interacting with Huge Hierarchies beyond Cone Trees, *IEEE Information Visualization '95*, pp. 74-81, 1995.
- [7] 山下, 藤代, 高橋, 堀井, 拡張 ConeTrees 技法による DAG 情報の可視化, *Visual Computing グラフィクスと CAD 合同シンポジウム 2002*, pp. 1-6, 2002.
- [8] Koike H., Fractal Views: A Fractal-Based Method for Controlling Information Display, *ACM Transactions on Information Systems*, 13, 3, pp. 305-323, 1995.
- [9] Johnson B., et al., Tree-Maps: A Space Filling Approach to the Visualization of Hierarchical Information Space, *IEEE Visualization '91*, pp. 275-282, 1991.
- [10] Chuah M., Dynamic Aggregation with Circular Visual Designs, *IEEE Information Visualization '98*, pp. 35-43, 1998.
- [11] Bruls D.M., et al., Squarified Treemaps, *Data Visualization 2000 (joint Eurographics and IEEE TCVG Symposium on Visualization)*, pp. 33-42, 2000.
- [12] Shneiderman B., et al., Ordered treemap layouts, *IEEE Information Visualization Symposium 2001*, pp. 73-78, 2001.
- [13] Bederson B., PhotoMesa: a zoomable image browser using quantum treemaps and bubblemaps, *UIST 2001*, pp. 71-80, 2001.
- [14] Rekimoto J., The Information Cube: Using Transparency in 3D Information Visualization, *Third Annual Workshop on Information Technologies & Systems*, pp. 125-132, 1993.
- [15] Sprenger T. C., et al., H-BLOB: A Hierarchical Visual Clustering Method Using Implicit Surfaces, *IEEE Visualization 2000*, pp. 61-68, 2000.
- [16] Shiozawa H., et al., WWW Visualization Giving Meanings to Interactive Manipulations, *HCI International '97*, pp. 791-794, 1997.
- [17] Doemel P., WebMap - A Graphical Hypertext Navigation Tool, *2nd Intl. Conf. WWW*, 785-789, 1994.
- [18] Hendley R. J., Drew N. S., Wood A., Beale R., Narcissus: Visualizing Information, *Symp. Information Visualization*, 90-96, 1995.
- [19] Inxight Star Tree (TM) SDKs, [http://www.inxight.com/products\\_sp/ht\\_sdk/index.html](http://www.inxight.com/products_sp/ht_sdk/index.html)
- [20] Durand D., et al., MAPA: A System for Inducing and Visualizing Hierarchy in Web sites, *9<sup>th</sup> ACM Conference on Hypertext and Hypermedia*, pp. 66-76, 1998.
- [21] Ayers E., Stasko J., Using Graphic History in Browsing the World Wide Web, *4th Intl. Conf. WWW*, 1996.
- [22] Frecon E., Webpath- A Three Dimensional Web History, *IEEE Information Visualization '98*, pp. 3-10, 1998.
- [23] Snowdon D., Fahlen L., Stenius M., A 3D Collaborative Virtual Environment for Web Browsing, *Virtual Reality World Wide '97*, 1997.