

長方形の空間充填による無閉路有向グラフの可視化

東原真希 (指導教員: 伊藤貴之)

1. はじめに

情報可視化が対象とするデータ構造は大きく7種類(1次元, 2次元, 3次元, n次元($n>3$), 時系列, 階層型, リンク) [1]とされており, その中でも階層型データの可視化は特に活発に研究発表されている. 階層型データの可視化技術の多くは, 木を描画する手法(Hyperbolic Tree, Cone Tree など)と, 空間充填手法(TreeMaps など)に大別される.

一方で身の回りには, DAG (Directed Acyclic Graph: 無閉路有向グラフ) 構造を構成するデータも多い. そして DAG 構造の多くは, 子ノードが複数の親ノードに接続されるような構造を持つ特殊な階層型データとみなすことができる. DAG 構造の可視化に関しても既存研究はいくつかあるが, まだ議論の余地が多い. 例えば木を描画する手法においては, Cone Tree という手法を DAG 構造に拡張した手法も発表されている [1]が, 例えば葉ノードを一望するといった目的には空間充填手法のほうが向いている. そこで本研究では, 空間充填型の階層型データ可視化手法「平安京ビュー」 [2]を拡張し, DAG 構造の可視化を試みる. 以後, DAG 構造を変換して構築される階層型データを DAG と略称する.

2. 関連研究

2.1 平安京ビュー

本研究では, 大規模階層型データに対する空間充填型の可視化手法「平安京ビュー」 [2]を使用する. 図1に平安京ビューの可視化結果の一例を示す. 平安京ビューでは, 葉ノードを長方形のアイコン・親ノードを長方形の枠で描画し, 階層構造を親ノードの長方形の枠の入れ子構造で表現している. それらの長方形を空間充填モデルに基づいて配置することによって, 大規模階層型データの全体を一画面で表示する. 「平安京ビュー」は, その可視化結果における葉ノードの格子状の配列が, 平安京の地図のように整然と並ぶことから命名された. 図1からもわかるように平安京ビューは, データの葉ノードと親ノードの階層構造よりも, 葉ノード群を一望することに主眼を置いた手法である.

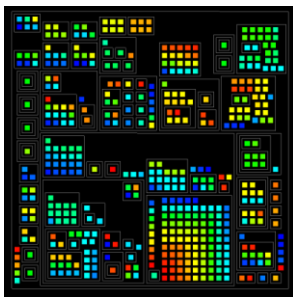


図1 平安京ビューの可視化例

2.2 FRUISNet

FRUISNet (FRamework User Interface Tangled Segments Network) [3]とは, ノードにアイテム情報が付加されたネットワーク構造の可視化の一手法である. FRUISNetは平安京ビューと同様に, 葉ノードをアイコンで, 親ノードを長方形の枠で表現するが, 平安京ビューとの大きな違いはその配置方法にある. 平安京ビューの配置は長方形の空間充填モデルのみに基づいていたのに対し, FRUISNetではノードの配置に力学モデルと空間充填モデルを併用している. それらを併用することによって, 関連するノード同士が近くに配置した上に, 描画面積を小さく抑えて一画面表示に成功している.

3. 平安京ビューの DAG への可視化

3.1 DAG の可視化の問題点

平安京ビューは木構造を持つ階層型データ可視化手法であ

るため, 葉ノードは親ノードを高々一つしか持つことができない. それによって, 平安京ビューで DAG を可視化しようと試みたとき, 起こる問題として次の2点が挙げられる.

- ▶ 本来一つであるノードを複数回描画する必要が生じる
- ▶ 「複数の親を持つ葉ノードがある」という情報を失う

図2に簡単な例を示す. ここで, 右の親ノード(枠)と左の親ノード(枠)それぞれの左上の葉ノードが本来一つのノードで複数の親に所属するノードであるとする. しかしこの図からは, 複数の親を持つ葉ノードの存在は読み取ることができない. また各々の葉ノードが, 可視化結果に示されている親以外の親を持つとしても, それを読み取ることができない.

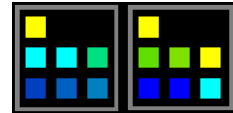


図2 複数の親に所属する葉ノードの表示例

また, 平安京ビューは描画面積を小さくするために, 長方形の空間充填を優先しており, 関連の強い葉ノードや親ノードを自動的に近くに配置することはできない. それゆえに, 複数の親を持つ葉ノードを複数回描画した時, 本来一つの情報であるノードが複数回描画され, かつ画面全体に散らばって配置される可能性がある, という問題もある.

3.2 階層型データとしての DAG の処理

ここで従来手法と提案手法での DAG 構造を持つ階層型データの処理について, 簡単な図を用いて説明する. 簡単な DAG 構造を持つデータを図3に示す.

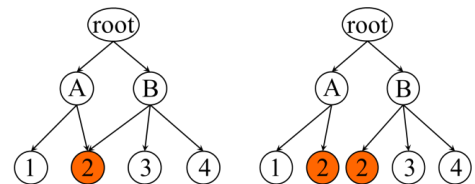


図3(左)本来の DAG 構造. (右)平安京ビューで DAG を可視化する際の一般的な木構造への変換

図3の例では, 色のついた葉ノード2が, 親ノードAと親ノードBを持っている. これを平安京ビューで表示するには, 図3(右)のように, 葉ノード2をノードが所属する親ノードの数だけ複数回描画する必要がある. しかしこの時, 前節でも述べてきたように, 平安京ビューで可視化すると, 本来のデータ構造を読めなくなってしまう. そこで本手法では図4(左)のように, 新たに親ノードを作成し, その下に葉ノードを格納することを考えた.

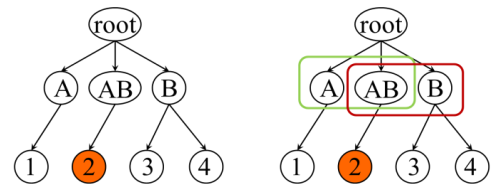


図4(左)親ノード AB の作成 (右)カテゴリの追加

新たに親ノード AB を作成することによって, 葉ノード2を参照した時, このノードが親ノードAと親ノードBの両方に属することを読み取ることができるようになる. しかしここで, 親ノードAに所属する葉ノードが, 葉ノード1と葉ノード2の2個が存在するにも関わらず, 葉ノード1のみが親ノードAに所属していると誤読される恐れがある. 親ノードBに注目した時にも同様のことが言える. そこで図4におけ

る親ノード間の関連を明示するために、図3における元々の親ノードに関する情報を付加する。図4(右)における緑と赤の枠が、図3における親ノードに対応する。本研究ではこれらを「カテゴリ」と称する。

3.3 画面配置アルゴリズム

以上の処理によって、DAG 本来の構造を可視化することを考える。ここまで簡単のために DAG を有向グラフで図示してきたが、続いて平安京ビューによる可視化との対応を考える。本研究では、図4(右)に示した構造を、平安京ビューの拡張によって図5のように可視化することを考える。

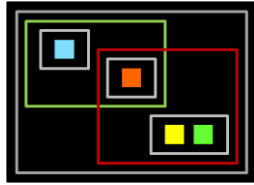


図5 提案手法による描画イメージ

図5のような描画を実現するにあたり、ノードの画面配置は非常に重要である。特に、同一カテゴリに属する親ノードは、画面上で極力離れないことが望ましい。2.1節でも述べたように、平安京ビューは画面配置に空間充填モデルのみを用いている。これは描画領域を小さくすることに有効であるが、関連のある親を近くに配置するという要求には、そのままでは対応しない。この問題を解決するために提案手法では、2.2節で紹介した FRUITSNET の画面配置アルゴリズムを応用する。具体的には、同一カテゴリに属する親ノード間を架空のリンクで接続し、このリンクが適切な長さを保つような力学モデルを適用することで、同一カテゴリに属する親ノードどうしを、画面上で近くに配置させるようにする。

3.4 カテゴリの描画

図5の描画イメージでは長方形でカテゴリを描画したが、長方形では描画面積が必要以上に大きくなり、カテゴリ間の不必要な重なりを生じやすくする。また長方形では全ての辺が平行または垂直になるため、これが読みにくさを生じる可能性もある。これらの問題点を解決するために、カテゴリの描画に対してバリエーションを考える。

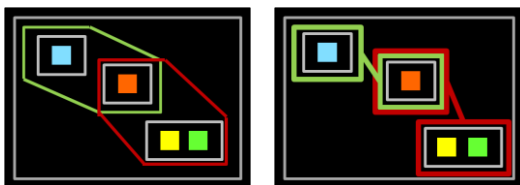


図6 (左)凸包の描画イメージ (右)枠と辺の描画イメージ

図6(左)は凸包でカテゴリを描画した例である。長方形での描画の直観的なわかりやすさを保ちつつ、長方形よりは描画面積は小さくできる。しかしそれでも、カテゴリ内の親が隣接しない時、関係のない親への重なりが大きくなることもある。これを軽減する一手段として、図6(右)に示すように、辺と枠での描画が考えられる。この方法では他の親への重なりを少なくすることができるため、親が離れて配置された時に効果がある。今後の課題として、近くに配置された親に対してはカテゴリを凸包で描画し、遠くに配置された親に対しては枠と辺を使って描画する、といったように二つの方法を併用した描画を試みたい。

4. 実行結果

4.1 使用しているデータ

我々は、「動向情報の要約と可視化に関するワークショップ (MuST)」が提供する毎日新聞全文記事データベース(1999年)から、ビジネス情報の記事を抽出し、各記事を葉ノードに、各記事から抽出されるキーワード群の組み合わせを親ノードに、各キーワードをカテゴリに対応させて DAG を構築した。以下に用いたデータにおいて、カテゴリ数 114、親ノード数

185、葉ノード数 1089 である。

4.2 画面配置結果

図7は、FRUITSNET の配置手法を利用して前述のデータを配置した例である。カテゴリ「パソコン」に所属する親ノードが画面上で集中的に配置されていることから、本研究の意図通りの配置結果が得られていることがわかる。

また、カテゴリ内の親が離れた場合にも全体を上下左右に4分割した時、同一の範囲内に収まる程度の広がりであった。



図7 カテゴリ「パソコン」に所属する親ノード (左)全体図 (右)拡大図

5. 他の DAG 可視化手法との比較

オイラー図に似たスタイルで DAG を描画する可視化手法は、提案手法以外にも、最近になっていくつか発表されている。親ノード間を架空のリンクで連結する手法は、Simonettoらの手法[4]や Santamaria ら[5]との手法とも共通している。しかし、これらの手法では画面配置において空間充填モデルを採用しておらず、画面占有面積を低減できるとは限らない。また、FRUITSNET に類似した画面配置手法を採用した Riche ら[6]の手法もあげられるが、この手法は提案手法と違って葉ノードを単位とした力学モデルを前提としており、その可視化結果は提案手法とは一長一短の関係にある。

6. まとめ

本研究では、大規模階層型データ可視化の一手法「平安京ビュー」を拡張し、FRUITSNET の配置モデルを活用することで、DAG 構造を可視化する手法を提案し、その実装の初期段階としての実行例を示した。

今後、カテゴリの描画に関する諸手法を実装するとともに、GUI 機能を追加開発する予定である。

謝辞

DAG をはじめとするグラフ理論について多数のご助言をいただきましたお茶の水女子大学の萩田真理子准教授に感謝の意を表します。本研究で用いた毎日新聞全文記事データベース(1999年)は「動向情報の要約と可視化に関するワークショップ (MuST)」によって提供されました。

参考文献

- [1] 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, 1-6 (2002).
- [2] 伊藤, 山口, 小山田, 長方形の入れ子構造による階層型データ可視化手法の計算時間および画面占有面積の改善, 可視化情報学会論文集, 26(2), 51-61 (2006).
- [3] T. Itoh, C. Muelder, K.-L. Ma, J. Sese, A Hybrid Space-Filling and Force-Directed Layout Method for Visualizing Multiple-Category Graphs, IEEE Pacific Visualization Symposium, 121-128 (2009).
- [4] P. Simonetto, D. Auber, An Heuristic for the Construction of Intersection Graphs, 13th International Conference on Information Visualization, 673-678 (2009).
- [5] R. Santamaria, R. Theron, Visualization of Intersecting Groups Based on Hypergraphs, IEICE Transactions on Information and Systems, E93-D(7), 1957-1964 (2010).
- [6] N. H. Riche, T. Dwyer, Untangling Euler Diagrams, IEEE Transactions on Visualization and Computer Graphics, 16(6), 1090-1099 (2010).