

無閉路有向グラフ可視化のための画面配置とカテゴリ描画の一手法

東原 真希¹⁾

伊藤 貴之²⁾

1) 2) お茶の水女子大学 大学院人間文化創成科学研究科

アブストラクト

階層型データは身近に存在するデータ構造であり、その可視化には多くの視点から研究が進められている。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数あり、その可視化にはまだ議論の余地が多い。本報告では DAG 構造を「親ノードを複数持つ子ノードが存在する特殊な木構造」に置き換え、大規模階層型データ可視化の一手法「平安京ビュー」の拡張によって可視化する試みを述べる。

1. はじめに

身の回りには互いに関連を持った要素を持つ情報にあふれており、それらを可視化することは、データ分析の側面において有効であると考えられる。また、組織内の人物の所属図や新聞記事のカテゴリ分類をはじめとし、身の回りには階層型データとみなすことができるものも多く存在する。

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]と言われており、それぞれ様々な可視化手法が考案されている。その中でも我々は大規模・複雑な階層型データの可視化に取り組んでいる。

階層型データの可視化技術の多くは、ノード・リンク型手法(Hyperbolic Tree [2], Cone Tree [3]など)と空間充填法(Treemaps [4]など)に大別される。一方で身の回りには、複数のカテゴリに属する新聞記事など、ノードの親が高々1つである木構造では表現できないデータも数多く存在する。親を複数持つノードが存在するデータは DAG (Directed Acyclic Graph: 無閉路有向グラフ)構造とみなすことができる。この DAG 構造の可視化に関しても既存研究はいくつかあるが、まだ議論の余地は多い。ノード・リンク型手法の DAG 構造への拡張は既に多く発表されており、例えば Cone Tree を拡張した手法が発表されている [5]。これは階層型データの親子関係を知りたい時に非常に有効であるが、逆に葉ノードを一望するといった目的には空間充填法のほうが向いている。以上のことから、空間充填法の DAG 構造への拡張にも一定の意義があると考えられる。

本報告では、空間充填型の大規模階層型データ可視化の一手法「平安京ビュー」 [6] を拡張し、DAG 構造を持つ階層型データの可視化を試みる。以後、DAG 構造を持つ階層型データを 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」 [3] を使用する。図1に平安京ビューの可視化の一例を示す。

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枠の入れ子で表現している。それらすべて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。また、葉ノードに割り当てられた値を高さとして持たせることができるので、二次元でデータの階層構造を表現し、アフィン変換によって回転することで、その値を比較することができる。図2に図1の可視化結果を回転させた一例を示す。回転はユーザのマウス操作によってなされる。

平安京ビューはこの可視化結果においてはノードが格子状に配列される様が、平安京の地図のように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を主眼を置いた手法である。

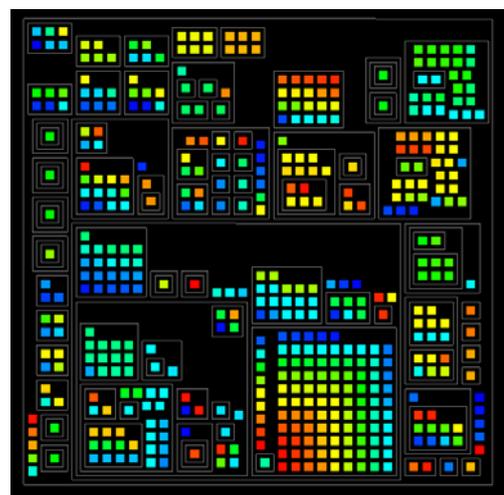


図1 平安京ビューの可視化例

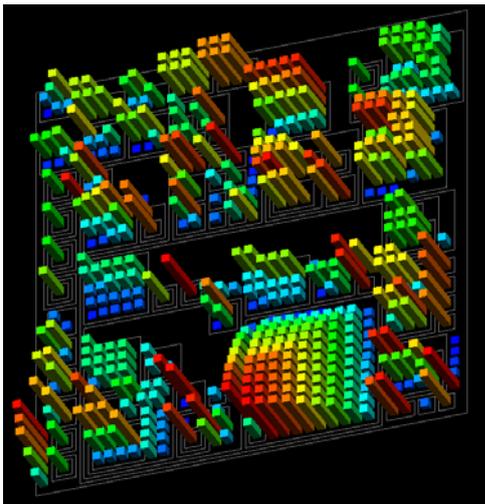


図2 可視化結果を回転させた例

2.2 FRUITNet

FRUITNet (Framework and User Interface for Tangled Segments: Network)[7]とは、アイテム情報が付加されたノードを有する階層型ネットワーク構造の可視化の一手法である。可視化の一例を図3に示す。

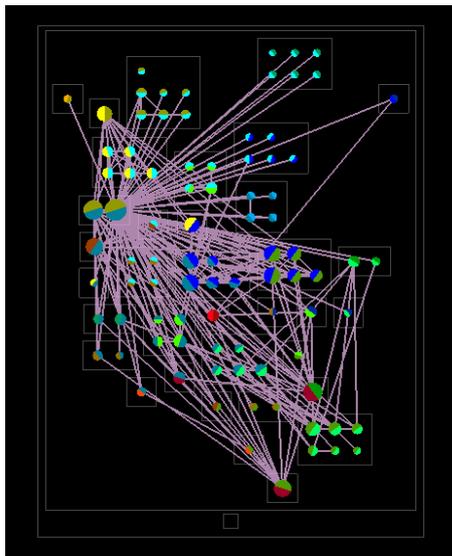


図3 FRUITNetの可視化例

FRUITNet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手法にある。平安京ビューの配置は長方形の空間充填モデルにのみに基づいていたのに対し、FRUITNetではノードの配置を力学モデルで計算した後に空間充填モデルを適用している。それらを併用することによって、リンクで連結されたノードや共通のアイテム情報を有するノードを画面上で近くに配置し、同時に描画面積の低減やノードの重なり回避を実現する。配置アルゴリズムについて4.1節にて詳しく示す。

3. 平安京ビューの拡張による DAG の可視化

3.1 DAG 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の手法であり、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。このため、平安京ビューでDAGを可視化するにはデータ構造に手を加える必要がある。

簡単な例として図4のような可視化例を挙げる。この例で可視化されたデータは2個の親ノードを有し、図4では左右に配列された灰色の枠として表示される。もしこのデータがDAGであり、左右両方の親ノードに接続された葉ノードがあるとしたら、同一の葉ノードを左右両方の枠の中に重複描画する必要がある。しかし、図4からは葉ノードが複数の親ノードに接続されている事実を読み取ることはできない。このことから、平安京ビューでDAGをそのまま表示した際には、

- 本来同一であるノードを複数回描画する必要が生じる
- 「複数の親ノードを持つ葉ノードが存在する」という情報を失う

という問題点が生じることがわかる。

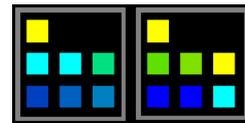


図4 平安京ビューによるDAGの表示に関する問題

また、平安京ビューは長方形の空間充填により描画面積の最小化を優先するため、関連の強い葉ノードや親ノードを自動的に近くに配置できる保証はない。それゆえに、複数の親を持つノードを複数回描画した時、本来同一のデータ要素を表しているノードが複数回描画されるという問題に加えて、その複数回描画された同一のデータ要素が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。図5に簡単な DAG 構造を持つデータを示す。

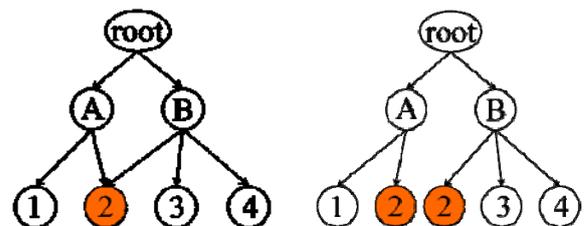


図5 (左)本来の DAG 構造。(右)平安京ビューで DAG を可視化の際の一般的な木構造への変換

この例では、色のついた葉ノード2が、親ノードAと親ノード

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 5(右)のように、葉ノード 2 をノードが所有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み取らなくなってしまう。そこで本手法では図 6(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る。

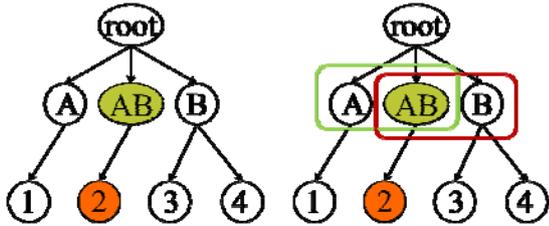


図 6 (左)親ノード AB を作成 (右)カテゴリの追加

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れるようになる。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 つが存在するにも関わらず、葉ノード 1 のみ親ノード A に所属すると誤読される恐れがある。親ノード B に注目した時も同様のことが言える。そこで図 6(左)における親ノード間の関連を明示するために、図 5 における元々の親ノードに関する情報を付加する。図 6(右)の緑色と赤色の枠が、図 5 の親ノードに対応する。本報告ではこれを「カテゴリ」と称する。

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。本研究では図 5(右)に示した構造を、平安京ビューの拡張によって図 7 のように可視化することを考える。

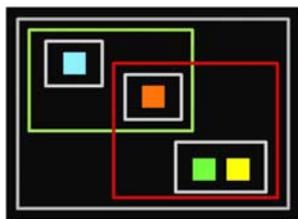


図 7 提案手法による描画イメージ

4. 提案手法における DAG の画面配置

4.1 画面配置アルゴリズム

図 7 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。特に同一のカテゴリに属している親ノードは、画面上で近くに配置されることが強く望まれる。2.1 節でも述べたように、平安京ビューが採用する空間充填モデルは、画面領域を小さくすることに有効であるが、関連のある親ノード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本手法では、2.2 節で紹介した FRUITSNET の画面配置アルゴリズムを適用する。図 8 にて FRUITSNET にお

ける画面配置の処理の手順を示す。

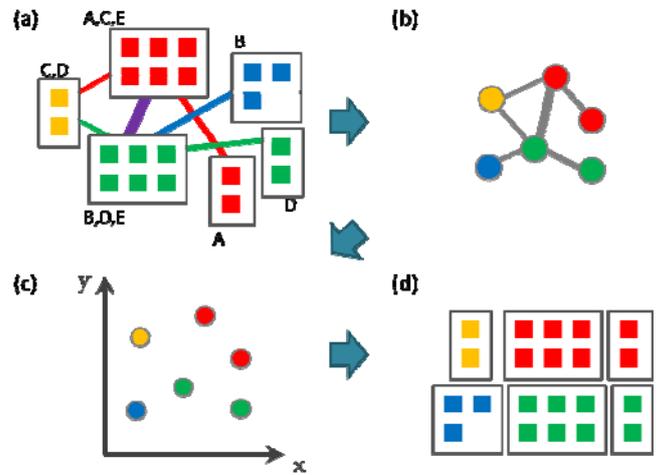


図 8 (a)同一カテゴリに属する親間に重み付き辺を引く。(b)力学モデルで配置を計算 (c)配置結果を理想位置とする (d)空間充填モデルで配置を調整

本手法では、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、それぞれのリンクにはノード数に比例して重み付けする。図 8(a)は親ノードをリンクで接続した概念図であり、各リンクの太さがその重みを表している。続いて図 8(b)に示すように、入力データ構造の親ノードをノードに置き換えたデータ構造を構築し、各リンクが適切な長さを保つような力学モデルを適用して配置を計算する。具体的には、ノード間を接続する重み付きのリンクにバネに仮想し、反復的解法によってバネ間の力に関する運動方程式を解くことで、リンクの長さを適正化する。さらに、図 8(c)に示すように力学モデルでの配置結果を理想位置として、空間充填モデルを適用して配置を調整している。図 8(d)に配置調整結果の例を示す。ここで、空間充填モデルでは以下の三点に重きを置いて配置を計算している。

- 長方形が重なり合わないこと
- 画面占有面積を最小化すること
- できるだけ理想位置に近い場所に配置すること

以上の配置手法により、同一カテゴリに属する親ノード同士を画面上で近くに配置させ、各長方形の重なりを回避し、かつ描画領域を小さく抑えることに成功している。

4.2 配置結果

図 9 は葉ノード数 1089、親ノード数 185 である DAG データを平安京ビューの空間充填モデルによって配置した例である。同一カテゴリ内に所属する親ノードがピンク色の枠で描画されているが、これらが画面上の広い範囲に分散しており、あまり望ましい配置結果ではないことがわかる。

一方で図 10 は、図 9 と同一のデータを FRUITSNET の配置アルゴリズムを適用し、力学モデルと空間充填モデルを併用した結果である。図 10 の全体図・拡大図の両方から、図 9 では画面全体に分散していた親ノードが、一か所に集まったことがわかる。

このデータにはカテゴリが114あったが、そのうち94は画面を四分分割した範囲内にすべての親ノードが収まった。カテゴリ内の親ノードが離れたところに配置された親ノードに関してはカテゴリ描画で問題を解決する。

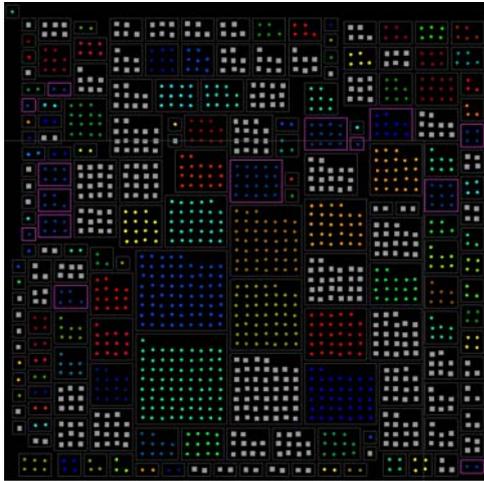


図9 空間充填モデルのみで配置した例

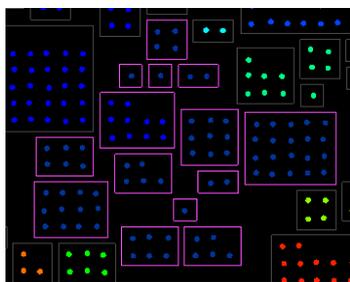
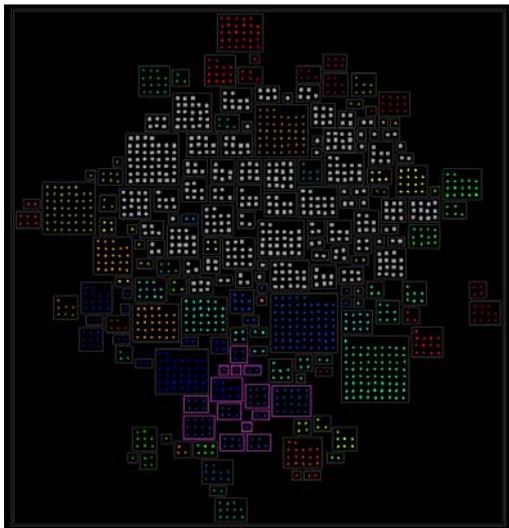


図10 FRUITSNetsの配置アルゴリズムを適用した同データ(上)と拡大図(下)

5. 提案手法におけるカテゴリの描画

5.1 長方形での問題点

我々はカテゴリ描画実装の第一段階として、長方形でカテゴリ描画を実装した。図11はその実行例であり、指定した一つの

カテゴリのみを赤い長方形の枠で描画している。また、カテゴリに所属する親ノードをピンク色でハイライトしている。この例からも読み取れるように、長方形ではカテゴリの描画面積が必要以上に大きくなる上、ピンク色でハイライトされていない親ノードがカテゴリ内に収まってしまい不必要な重なりが起きている。図11では赤い長方形内部にて、右上部および左下部に配置したいくつかの親ノードが、当該カテゴリに属さないにも関わらず赤い長方形の枠の中に包括されてしまっている。また、図11には見られないが、カテゴリに所属する親ノードのうち、ごく少数が離れた位置に配置された時には、カテゴリに属さない親ノードの不適切な包括が顕著に生じることが観察されている。

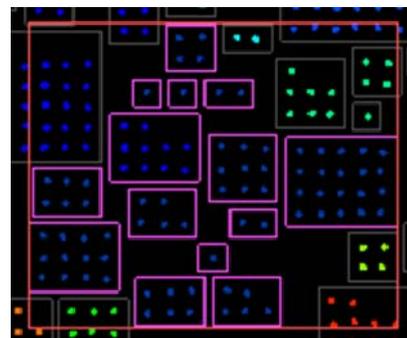


図11 カテゴリを長方形で描画した例

また、関連を持つカテゴリを複数同時に描画した例を図12に示す。この図のみでは、それぞれのカテゴリ内に所属する親ノードを判別することは難しい。中央に位置する赤色でハイライトされたカテゴリと、それに所属するピンク色でハイライトされた親ノード以外は、視覚的な判別が困難である。

このように、複数のカテゴリを描くという点からも、カテゴリに所属する親ノードのハイライト表示することなくカテゴリ内の親を認識する描画が求められる。

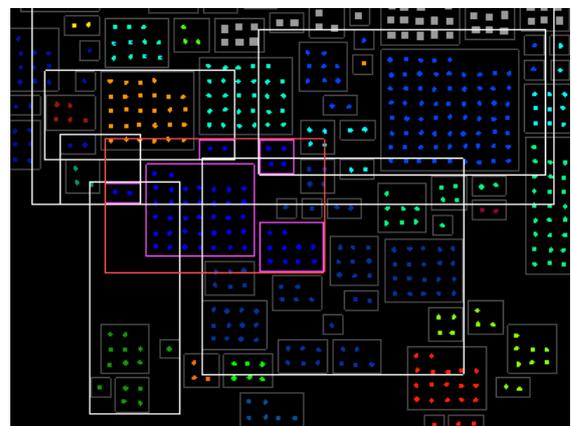


図12 カテゴリを複数描画した例

これらの問題を解決するために、カテゴリ描画に対して図13のようなバリエーションを考えた。

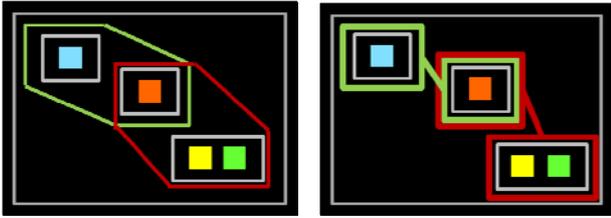


図 13 (左)凸包の描画イメージ (右)枠と接続線の描画イメージ

図 13(左)は凸包でカテゴリを描画したイメージ例である。長方形での描画の直観的な分かりやすさを保ちつつ、長方形よりは描画面積を小さくできると期待できる。しかしそれでも、カテゴリ内の親ノードが隣接しない時、関係のない親ノードへの重なりが大きくなることもある。この問題を軽減する一手段として、図 13(右)に示すように、枠と接続線での描画が考えられる。この方法では他の親ノードへの重なりを少なくすることができるため、親ノードが離れて配置されたときに効果があると考えた。

5.2 カテゴリ描画結果

図 14 は凸包でカテゴリ描画を実装した例と、同一のカテゴリを長方形で描画した例である。凸包の計算には包装アルゴリズム(Gift-wrapping Algorithm)を適用した。図 14 から長方形よりも他ノードへの重なりが減少しているが、いまだにカテゴリに属さない親ノードを包括してしまっていることがわかる。

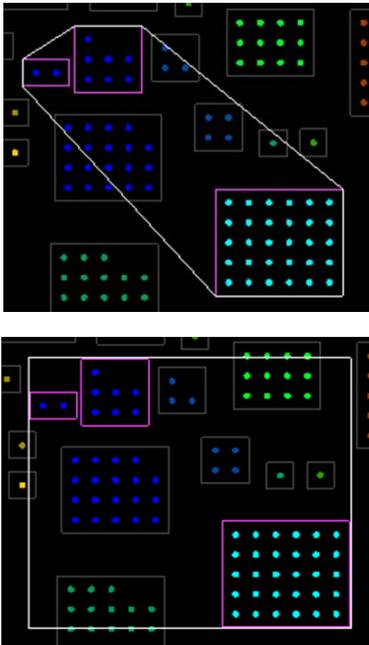


図 14 (上) 凸包でのカテゴリ描画例
(下) 同一カテゴリを長方形で描画した例

図 15 はカテゴリ内に所属する親ノードを接続線で連結した結果である。図 14 と同一のカテゴリを表示しているが、カテゴリに所属しない親ノードへの重なりがなくなった。しかし、カ

テゴリ内の 3 つの親ノードのうち上部にある二つは配置が近すぎるために、接続線の存在に気がつかないおそれがある。また、カテゴリ内すべてのノードが隣り合わせて配置された時、接続線が煩雑になりカテゴリの読み取りが困難になることが考えられる。

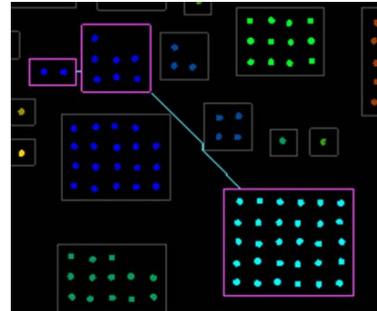


図 15 枠と接続線でカテゴリ描画した例

そこで、近くに配置された親ノードに対してカテゴリを凸包で描画し、遠くに配置された親ノードに対しては枠と接続線を使って描画する、という二つの方法の併用を試みた。結果を図 16 に示す。

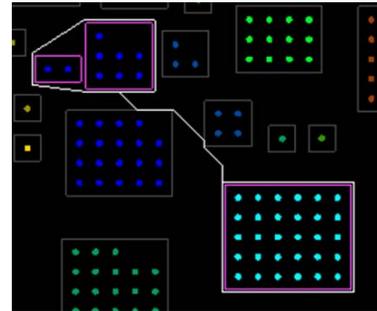


図 16 凸包と接続線を組み合わせた描画例

この表現ではユーザに指定された距離以下の場合、ノード群を凸包で囲み、それらの凸包を接続線で連結している。このとき本手法では接続線を、親ノードが配置されていない空いた部分を通る折れ線として描画することで、親ノードとの干渉を避ける。これにより、カテゴリ外の親ノードへの重なりなど、上記二つのカテゴリ描画での問題点の解消を目指す。

6. 他の DAG 可視化手法との比較

6.1 ノード配置に関する先行手法

オイラー図に似たスタイルで DAG を描画する空間充填型の可視化手法は、本報告の提案手法以外にも、最近になっていくつか発表されている。例として Simonetto らの手法[8]や Santamaria らの手法[9]は、適切なノード配置を得るために親ノード間を架空のリンクで連結する、という点で提案手法と共通している。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、画面占有率を低減できるとは限らない。

また Riche らは, FRUITSNet に類似した画面配置手法を採用した Untangling Euler Diagram[10]を提案している. 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いるため, 配置結果自体は提案手法と一長一短の関係にあるが, 提案手法と比べて計算時間がかなり大きくなると考えられる.

6.2 カテゴリ描画に関する先行手法

6.1 節で紹介した Untangling Euler Diagram は, カテゴリ描画処理においても提案手法に類似する点がある. 可視化例を図 17 に挙げる. この手法では二つのアプローチがある. 一つは図 17(左)で示した子ノードをそれぞれ一度だけ描画し, カテゴリを変形し入れ子状にした描画である. 図 17(右)では同一のデータをカテゴリの描画は長方形で統一し, 子ノードを複数回描画し, 複数回描画した子ノードを繋いだ描画である. 図 17(左)の単語を本報告でいう親ノードに置きかえると目的の似た手法であると言える. カテゴリ描画に関しては鮮やかな色分けと直観的なわかりやすさを持った手法であるが, カテゴリの数が増えれば増えるほどカテゴリの重なり, 色の区別が難しくなることが考えられる.

また, 複数のカテゴリへの所属を許す点群を描画する手法として, Bubble Sets[11]や Line Sets[12]といった手法が発表されている. これらの手法は実空間や時系列空間などで既に位置を決定された点に対してカテゴリを描画する手法であり, ノード配置も課題の一部と位置づけている提案手法とは主旨が異なる.

Bubble Sets[11]は, 本報告でいうカテゴリに相当する枠を, 一定の太さを有する帯領域で描く手法である. カテゴリの外の要素への重なりは完全にはなくならないが低減することができる.

Line Sets[12]では Bubble Sets 同様, あらかじめ配置された点群間の関連提示を目的とした手法である. この手法では, カテゴリに相当する情報を, ノードを接続する曲線で, それもできるだけ人が直感的に描く曲線に近い形状で表現する.

以上の関連手法に対する提案手法の差異は, カテゴリ描画に「凸包+接続線」という方法を採用しているという点である. 今後の課題として, 次節で紹介するユーザテストを実施して, 提案手法の有効性を検証したい.

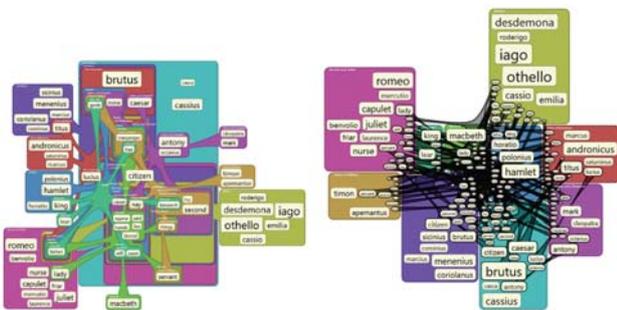


図 17 Untangling Euler Diagram[10]による「シェイクスピアの 10 の戯曲」の頻出する単語群の可視化
(左)単語は一つだけ描画し作品を入れ子にしている例(ComED)
(右)単語を複数回描画し作品を長方形で示した例(DupED)



図 18 Bubble Sets の可視化例
(左)地図上に示された建物を種類ごとに枠で囲んでいる例
(右)タイムライン上に並んだ図をカテゴリ分けしている例

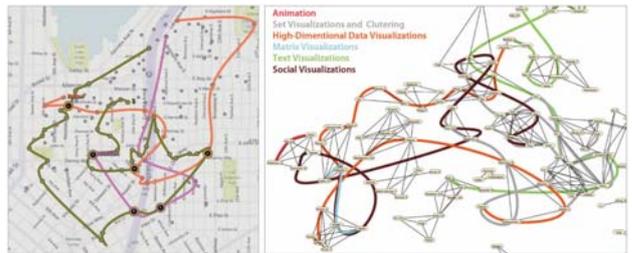


図 19 Line Sets の可視化例
(左)地図上のレストランをカテゴリ別に表示した例
(右)ソーシャルネットワークのコミュニティを示した例

6.3 今後の課題：ユーザテスト

今後の課題として, 同一のデータに対して以下の 6 種類のカテゴリ描画方法を適用し, ユーザテストによってその有効性を比較したい.

1. カテゴリ内の親ノードすべて辺でつなぐ
2. 凸包で描く
3. 凸包と接続線で描く
4. Untangling Euler Diagram(ComED)
5. Bubble Sets
6. Line Sets

ユーザテストの具体的な設問としては, 例えば以下のようなものを考えている.

- 要素数の把握しやすさ
- カテゴリに所属する親ノードの判別しやすさ
- 親ノードの所属するカテゴリの判別しやすさ

図 20 は, 同一の親ノード群を囲むカテゴリを, 凸包と接続線で描いた図と, Bubble Sets で描いた図である. このような同一のデータからの可視化結果を, ユーザに提示して設問に回答させることで, 各描画手法の長所や短所を検証したい.

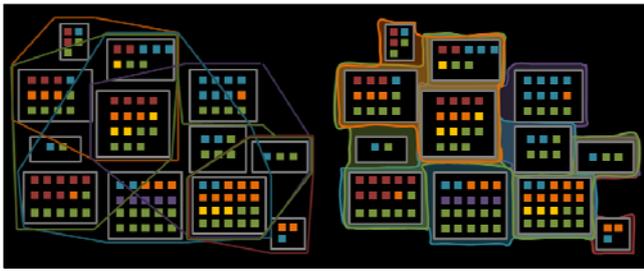


図 20 イメージ比較例
(左)凸包と接続線 (右)Bubble Sets

7. まとめ

本報告では、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 DAG 構造を可視化する手法を提案し、その実行例を示した。ノードの配置には平安京ビューの特性を活かしかつ、関連のあるノードを近くに配置するために FRUITSNet の配置アルゴリズムを適用した。また、カテゴリの描画においては複数の描画を実装して検討した。今後の課題として前章で述べた通り、ユーザテストを実施して提案手法の有効性を検証したい。

参考文献

- [1] B. Shneiderman, The Eyes Have It; A Task by Data Type Taxonomy for Information Visualization, IEEE Symposium on Visual Language, 336-343, 1996.
- [2] J. Lamping, R. Ra, P. Pirolli, The Hyperbolic Browser: A Focus+context Technique for Visualizing Large Hierarchies, Journal of Visual Languages and Computing, 7(1), 33-55, 1996.
- [3] J. Carrier, R. Kazman, Research Paper: Interacting with Huge Hierarchies beyond Cone Trees, IEEE Information Visualization '95, 74-81, 1995.
- [4] B. Johnson, B. Shneiderman, Tree-Maps: A Space Filling Approach to the Visualization of Hierarchical Information Space, IEEE Visualization '91, 275-282, 1991.
- [5] 山下, 藤代, 高橋, 堀井, 拡張 ConeTree 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, 1-6, 2002.
- [6] 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.
- [7] 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.
- [8] P. Shimonetto, D. Auber., An Heuristic for the Construction of Intersection Graphs, 13th International Conference on Information Visualization, 637-678, 2009.
- [9] R. Santamaria, R. Theron, Visualization of Interesting Groups Based on HyperGraphs, IEICE Transactions on Information and

Systems, E93-D(7), 1957-1964, 2010.

[10] N. H. Riche, T. Dwyer, Untangling Euler Diagrams, IEEE Transactions on Visualization and Computer Graphics, 16(6), 1090-1099, 2010.

[11] C. Collins, G Penn, S. Carpendale., Bubble Sets: Revealing Set Relations with Isocontours over Existing Visualizations, IEEE Transactions on Visualization and Computer Graphics, 15(6), 1009-1016, 2009.

[12] B. Alper, N. H. Riche, G Ramos, M. Czerwinski, Design Study of LineSets, A Novel Set Visualization Technique, IEEE Transactions on Visualization and Computer Graphics, 17(12), 2259-2267, 2011.