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† 伊藤 貴之†

†お茶の水女子大学 〒112-8610 東京都文京区大塚 2-1-1

E-mail: †{ maki , itot }@itolab.is.ocha.ac.jp,

あらまし 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. また, その具体的な画面配置手法と描画手法について検討する.

Visualization of DAG Information with Space-Filling Layout

Maki HIGASHIHARA† Takayuki ITOH†

† Department of Information Sciences, Faculty of Science, Ochanomizu University

2-1-1 Otsuka, Bunkyo-ku, Tokyo, 112-8610 Japan

E-mail: †{maki , itot}@itolab.is.ocha.ac.jp

Abstract

Hierarchical data is familiar to our daily life, and therefore many researchers have visualized. On the other hand, DAG (Directed Acyclic Graph) is also familiar to our daily life; however, visualization of DAG is still an open problem. This paper presents our study on visualization of DAG by extending “HeiankyoView”, a visualization technique for large-scale hierarchical data visualization technique. This study treats DAG as a special tree structure which child-nodes may be connected to multiple parent-nodes. This paper also discusses detailed algorithm and implementation of node layout and drawing techniques.

1. はじめに

身の回りには階層構造を構成するデータは非常に多い. 我々は大規模・複雑な階層型データの情報可視化に長きにわたって取り組んでいる. 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[2]. これは階層型データの親子関係の構造を知りたいときに非常に有効であるが, 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, 空間充填型の大規模階層型データ可視化の一手法「平安京ビュー」[3]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, 本報告の中では DAG 構造を持つ階層型データのことを 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いて葉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．

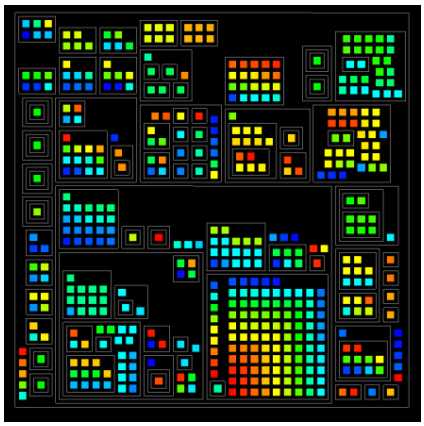

図 1　平安京ビューの可視化例

2.2 FRUITSNet

FRUITSNet (FRamework User Interface Tangled Segments Network)[4] 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．可視化の一例を図 2 に示す．

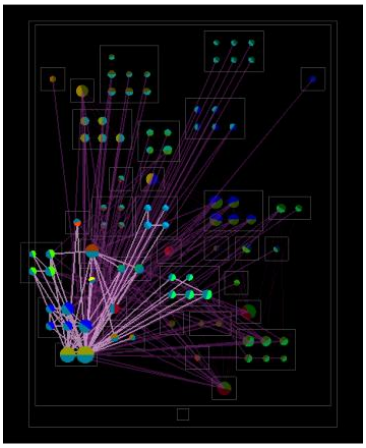

図 2 FRUITSNet の可視化例

FRUITSNet 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，FRUITSNet 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存在する」という情報を失う

簡単な例を挙げると，図 3 のような表示結果が存在する．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可能性がある．この時，図 3 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．またそれぞれの葉ノードを見ても，可視化結果に示されている親以外の親を持つことは読み取ることができない．

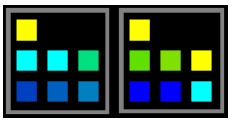

図 3　複数の親に所属する葉ノードの表示例

また，平安京ビューは描画面積を極力小さくするために，長方形の空間充填に重きを置いており，葉ノードや親ノードの関連の強さによって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．図 4 に簡単な DAG 構造を持つデータを示す．

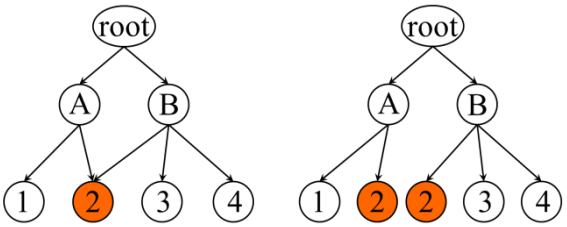

図 4 (左)本来の DAG 構造．(右)平安京ビューで DAG を可視化する際の一般的な木構造への変換

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 B を持っている．これを平安京ビューで表示するには，図 4(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 5(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．

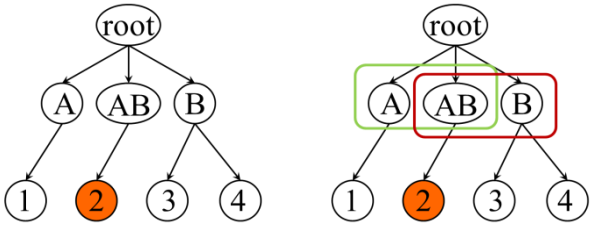

図 5(左)親ノード AB の作成 (右)カテゴリの追加

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 5 における親ノード間の関連を明示するために，図 4 における元々の親ノードに関する情報を付加する．図 5(右)における緑と赤の枠が，図 4 における親ノードに対応する．本報告ではこれらを「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．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 5(右)に示した構造を，平安京ビューの拡張によって図 6 のように可視化することを考える．

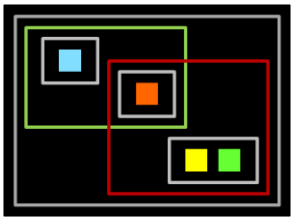

図 6 提案手法による描画イメージ

3.3 画面配置アルゴリズム

図 6 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 6 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．

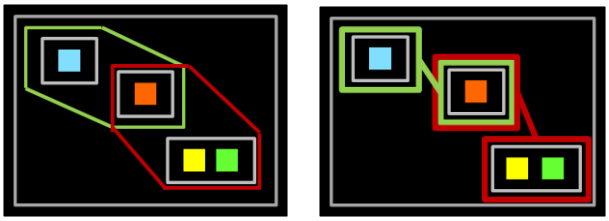

図 7 (左)凸包の描画イメージ
(右)枠と辺の描画イメージ

図 7(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がある．これを軽減する一手段として，図 7(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. 実行例

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 8 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．

図 8　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

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．これは図 9 の長方形の空間充填モデルのみを使った配置結果と比べると，明らかにカテゴリ内の親が近くに配置されたことがわかる．

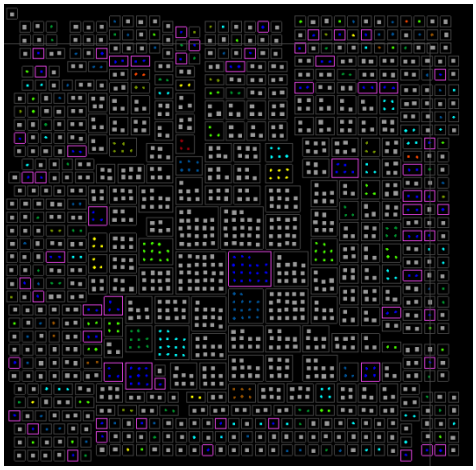

図 9 長方形の空間充填モデルのみでの配置

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，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．

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[5]や Santamaria ら[6]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[7]の手法もあげられるが，この手法は提案手法と違って子ノードを単位とした力学モデルを前提としており，その可視化結果は本報告の提案手法とは一長一短の関係にある．

6. まとめと今後の課題

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，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ください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「動向情報のようやく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] B. Shneiderman, The Eyes Have It; A Task by Data Type Taxonomy for Information Visualization, IEEE Symposium on Visual Language, 336-343 (1996).

[2] 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).

[3] 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).

[4] 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).

[5] P. Simonetto, D. Auber, An Heuristic for the Construction of Intersection Graphs, 13th International Conference on Information Visualization, 673-678 (2009).

[6] R. Santamaria, R. Theron, Visualization of Intersecting Groups Based on Hypergraphs, IEICE Transactions on Information and Systems, E93-D(7), 1957-1964 (2010).

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