

オブジェクト指向データベースの半自動可視化環境

尾下真樹[†] 牧之内 顕文^{††}

本論文では、オブジェクト指向データベースの検索結果に対して、さまざまな可視化を対話的に適用するためのシステムを提案する。既存の汎用的な可視化システムでは、元のデータから可視化結果の図形への変換方法をすべてユーザが設定する必要があった。そのため、自由度の高い可視化が行える反面、さまざまな可視化方法を試行錯誤的に適用することが難しく、また、有効な可視化を行うためには、既存の可視化技法に関する十分な予備知識を持っていることが要求されていた。本論文で提案するシステムでは、可視化技法に関する予備知識の少ないユーザであっても、容易にさまざまな可視化を行えることを目標としている。本システムのユーザは、可視化の方法を具体的に指定する必要はなく、ユーザの入力した抽象的な可視化目的をもとに、適切な可視化方法をシステムが決定する。ユーザは、可視化目的を逐次的に追加することで、対話的な可視化・分析作業を行うことができる。このような半自動可視化環境を実現するため、我々は、主に2つの手法を開発した。まず、既存の可視化技法を構成している基本的な表現要素を抽出し、その基本表現を組み合わせることでさまざまな可視化を実現する手法を開発した。また、ユーザから与えられた可視化目的に応じて、適切な表現の組合せを決定するための手法を開発した。本論文では、我々の開発したこれらの手法について述べ、本手法による可視化の具体例を示す。

A Semi-automatic Visualization Environment for Object-oriented Database Systems

MASAKI OSHITA[†] and AKIFUMI MAKINOCHI^{††}

This paper presents a novel system to apply various visualizations to a query result in an object-oriented database system. The existing visualization systems require users to specify the details of a method for converting data to visualized figures. Therefore, it is difficult to try various visualizations and analyses rapidly. Furthermore, to achieve an efficient visualization, the user should be familiar with the visualization techniques. In this research, we aim to provide a system in which even a beginner of visualization techniques is able to perform various visualizations easily and interactively. The system decides the suitable method based on abstract visualization goals that are given by the user. By adding a visualization goal incrementally, the user achieves interactive visualizations and analysis. To realize this system, we developed two methods. We developed a technique to realize various visualizations by combining the basic representations that were extracted from existing visualization techniques. In addition, we developed a technique to determine efficient combinations automatically in response to a visualization goal that is given by the user. This paper presents these method and shows the efficiency of our approach by applying the methods to an example.

1. はじめに

近年、データベースの役割として、単にデータを管理するだけでなく、データベース中のデータを分析

することによって、企業の戦略決定などへ活用できることが期待されている。企業ではデータウェアハウスの構築が進んでおり、多次元データ解析(OLAP: On-Line Analytical Processing^①)やデータマイニングなどの分析手法への関心が高まっている。しかし、これらの分析手法は、特定形式の大量データからの知識発見に应用対象が限られている。そこで、これらの手法を補う手段として、情報可視化の技術が注目されている¹⁾。データベース内のデータをユーザに対して視覚的に表示することによって、ユーザの持つ背景知識を活用した、ユーザ主体のデータ分析作業を行うことが

[†] 九州大学大学院システム情報科学府知能システム学専攻
Department of Intelligent Systems, Graduate School
of Information Science and Electrical Engineering,
Kyushu University

^{††} 九州大学大学院システム情報科学府知能システム学部門
Department of Intelligent Systems, Graduate School
of Information Science and Electrical Engineering,
Kyushu University

できる。

データベースの可視化に限らず、一般に、情報可視化の技術は計算機科学の分野における非常に大きな課題であり、これまでに多くの研究が行われてきた。とりわけ、近年の計算機の処理能力や描画能力の向上、扱うデータの大規模化などにより、情報可視化の応用領域は広がり、より重要な研究課題となっている。これまでの研究によって、さまざまな可視化手法が開発されている。しかし、有効な手法は対象とするデータや目的によって異なるため、情報可視化に関する十分な予備知識のないユーザにとって、既存の技法を有効に活用し、効果的な可視化を実現することは難しいという問題がある。また、現在の可視化システムでは、1つの可視化を適用するたびに面倒な設定作業が必要であるため、ユーザがさまざまな可視化手法を試行錯誤的に適用することが困難である。

本論文では、対象とするデータベースとしてオブジェクト指向データベースを取り上げ、データベースの検索結果に対して、ユーザがさまざまな可視化を対話的に適用することで、データの分析・把握作業を行えるようなシステムを提案する。本研究では、上記の課題をふまえ、可視化技法に関する予備知識の少ないユーザであっても、容易にさまざまな可視化技法を適用できるシステムを目標としている。そこで、本システムでは、ユーザが可視化の方法を具体的に指定するのではなく、ユーザの入力した抽象的な可視化目的をもとに適切な可視化方法を決定するような、半自動可視化環境を実現する。本システムのユーザは、可視化するオブジェクトのクラスのプロパティに対して、どのような可視化を行いたいのか、という可視化目的を指定する。可視化目的は複数指定可能であり、可視化目的を追加するたびに、その目的を実現するように可視化の方法が変更される。ユーザは、可視化目的を逐次的に追加することで、対話的な可視化・分析作業を行うことができる。

このようなシステムを実現するため、我々は、主に2つの手法を開発した。まず、既存の代表的な可視化技法を構成している基本的な表現要素を抽出し、その基本表現を組み合わせることで、さまざまな可視化を実現する手法を開発した。また、その可視化手法に基づき、ユーザから与えられた可視化目的に応じて、適切な表現の組合せを自動的に決定するための手法を開発した。

本手法では、最低限の表現の組合せによって可視化を実現するため、従来の汎用的な可視化環境に比べて実現可能な可視化表現の自由度は低い。その代わり、

ユーザの与える可視化目的から、自動的に表現の組合せを決定することを可能としている。また、本研究が対象としているオブジェクト指向データベースでは、オブジェクトの属性やオブジェクト間の関連がスキーマとして明確に定義されているという特徴がある。そこで、本手法では、そのスキーマ情報を利用することによって、ユーザによるメタデータの設定作業を必要とすることなく、可視化方法を自動的に決定する。

次章では、データベースの可視化システムに関する関連研究について述べる。3章では、我々の提案する可視化システムの概略を述べる。そして、4章では、基本的な表現要素の組合せによる可視化の実現手法について述べる。5章では、ユーザの与える可視化目的から、表現方法を自動的に決定するための手法について述べる。最後に、6章では、本システムによる可視化の適用例を示す。

2. 関連研究

これまで、データベース内の情報の可視化を目標として、さまざまなシステムが研究・開発されてきた。従来の主なアプローチとしては、散布図による可視化環境と、モジュールベースの可視化環境(MVE: Modular Visualization Environment)の2種類がある。まず、散布図による可視化環境の代表的なものとしては、XGobi⁵⁾やIVEEE³⁾、商用のシステムのSpotfireなどがあげられる。これらのシステムでは、それぞれのデータを2次元や3次元の空間におけるノードとして描画する。その際、データの属性値と、ノードの座標や色などの描画属性を対応づけることによって、データの属性値の分布状況や、複数の属性値どうしの相関関係を把握することができる。また、表示する値の範囲をユーザが動的に変更できるインタフェースを提供したり、複数の散布図を同時に表示したりすることで、さまざまな分析作業を行うことができる⁵⁾。しかし、散布図による可視化方法では、基本的に1つのデータを1つのノードとして描画することしかできないため、オブジェクトどうしの関係のような構造的な情報を表現することが難しい。

一方、モジュールベースの可視化環境として、Tioga²¹⁾や、商用システムのAVS、INFOVISER¹⁾などがあり、現在の汎用的な可視化ツールのスタンダードとなっている。これらのシステムでは、データ変換を行う基本的なモジュールを組み合わせることで、データベースのデータから可視化結果の図形への変換方法をプログラミングできる環境を提供している。そのため、ユーザの設定次第でさまざまな可視化技法を

実現することができる。しかし、有効な可視化を実現するためには、ユーザには、既存の可視化技法に関する十分な予備知識を持っていることが要求される。また、現在可視化しようとするデータに対して、どのような可視化技法を適用するのが効果的であるか、という仮定をあらかじめ持っていなければならない。これらのシステムでは、新しい可視化を行うたびに、その方法をユーザが設定する必要があるため、手軽にさまざまな可視化技法を試行錯誤することが難しいという問題がある。

このような従来の可視化環境の問題を解決するため、可視化技法に精通していないユーザでも既存の可視化技法を活用できるように、有効な可視化技法の発見を支援できるシステムへの要求が高まっている。その代表的なシステムとして、Fujishiroらによって開発された、科学技術計算結果の可視化支援のための、GADGET システム¹¹⁾がある。GADGET システムは、科学技術データのための可視化技法を体系化した Wehrend の分類法²⁴⁾を利用している。Wehrend 分類では、可視化の目的として、データタイプ $D = \{\text{Scalar, Nominal, Direction, Shape, Position, Spatially Extended Region or Object, Structure}\}$ と、データに対する操作 $A = \{\text{Identity, Locate, Distinguish, Categorize, Cluster, Rank, Compare, Associate, Correlate}\}$ の組が与えられると、その目的を実現する適切な可視化技法が特定できると仮定している。GADGET システムでは、Wehrend 分類に基づく可視化事例¹⁵⁾を利用することで、MVE のユーザに対して、入力された可視化目的に対応する可視化技法を実現するためのモジュールのテンプレートを提供する。最近、古畑らによって、GADGET システムのアプローチを、科学技術計算結果の可視化技法だけではなく、情報可視化の技法に適用した GADGET/IV システム²⁾が提案されている。古畑らは、最近の情報可視化に関する代表的な文献から可視化技法と可視化目的を収集し、既存の可視化技法を 14 種類に分類し、可視化目的として 14 種類の語彙を定義して、これらに対応付けている。GADGET/IV システムを利用することで、たとえば、ユーザがデータタイプとして Nominal、操作として Compare を与えると、可視化技法 BarChart が提案され、BarChart を実現するためのモジュール・テンプレートが MVE 上にロードされる。

しかし、Wehrend の手法では、ユーザは全体で 1 つの可視化目的のみしか指定できないため、複雑な構造データに即した可視化を行ったり、複数の可視化

手法を組み合わせたりといった、きめ細かい操作ができないという問題点がある²⁴⁾。GADEGET/IV システムでは、それぞれの可視化技法を実現するためのテンプレートをあらかじめ用意しておく必要がある。そのため、複数の可視化目的を指定できるように GADEGET/IV システムを拡張しようとする、すべての可視化目的の組合せに応じた大量のテンプレートを用意する必要があり、そのままのアプローチでは拡張は困難である。また、GADGET/IV システムによって提供されるような洗練された語彙を用いてもなお、一般のユーザにとって、多くの選択肢の中から適切なデータタイプや可視化操作を指定することは難しいという問題もある。

本研究では、GADGET/IV システムと同様に、ユーザの与える抽象的な可視化目的に応じて、システムが可視化の実現方法を自動的に決定するというアプローチをとる。しかし、本手法では、あらかじめ用意されたテンプレートを用いて可視化を行うのではなく、基本的な表現要素をユーザの可視化目的に応じて動的に組み合わせることで可視化を行う。このことにより、本手法は、複数の可視化目的に応じた可視化を実現可能としている。本システムのユーザは、オブジェクトのプロパティごとに可視化目的を指定することができ、新しい可視化目的を追加するたびに、その目的を実現するようにシステムが可視化方法を変更する。

本手法では、MVE のように汎用的な可視化環境を提供するのではなく、実現できる可視化技法をある程度限定し、最低限の表現要素の組合せによって可視化を行う。その結果、ユーザの指定した可視化目的から、可視化表現の組合せを自動的に導出することを可能としている。また、本システムは、スキーマ定義から得られる情報を利用することで、指定されたプロパティに対して適用可能な可視化操作のみを選択肢として提示するといった、可視化技法や語彙に十分精通していないユーザでも容易に利用できるようなユーザ支援を行う。

3. 可視化システムの概要

本研究では、オブジェクト指向データベースの標準仕様である ODMG3.0⁸⁾に準じて考える。ODMG3.0 では、データベースに格納される永続オブジェクトのクラスは、スキーマによって定義される。それぞれのクラスは属性や関連などのプロパティを持ち、属性の型情報や、関連の持つ 1 対 1 や 1 対多などの関係情報が定義される。また、ODMG3.0 ではスキーマインタフェースが定義されており、これらのスキーマ情報を

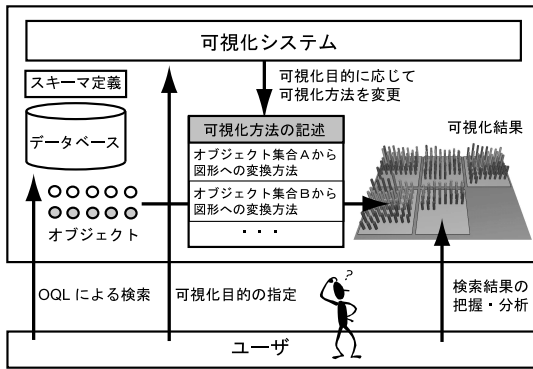


図1 システム構成

Fig. 1 The system diagram.

アプリケーションから取得して利用することが可能となっている。

図1に、本論文で提案するシステムの構成図を示す。本システムでは、OQL⁸⁾による問合せの結果に対して、ユーザが対話的に可視化を行える機能を提供する。本システムは、オブジェクトの集合を、3次元空間における図形として描画することで可視化を実現する。オブジェクトから図形への変換方法は、それぞれのオブジェクト集合ごとに定義される。オブジェクトの属性と図形の描画属性の対応関係は、テーブル形式で記述され、インタフェースとしてユーザに示される(図1)。本システムでは、ユーザは、その可視化の方法を直接指定するのではなく、どのような可視化を行いたいかという抽象的な可視化目的を指定することで、システムがその目的を実現するための適切な可視化方法を自動的に決定する。可視化目的は、あるオブジェクト集合と、そのクラスの1つのプロパティ、そして、そのプロパティに対して適用する操作の組によって与える。ユーザは、可視化されたシーンをブラウジングすることで、検索結果を分析・把握することができる。本システムでは、可視化目的はそれぞれのプロパティごとに指定可能であるため、ユーザは、逐次的に可視化目的を追加することで、対話的な可視化・分析作業を行うことができる。

本システムは、クラスの属性や関連の情報を、基本的に、ODMG3.0のスキーマインタフェースを用いることで、スキーマ定義から自動的に取得する。そのため、既存の可視化ツールが必要としていた、データベースから可視化システムへのデータ変換作業や、メタデータの設定作業を必要としないという利点がある。また、ユーザが可視化目的を入力する際には、属性の型情報をもとに、選択可能な候補のみがユーザに示されるため、可視化手法に関する十分な予備知識のない

ユーザであっても本システムを有効に活用できることが期待できる。

このようなシステムを実現するため、我々は、主に2つの課題に対して取り組み、その解決手法を開発した。1つは、基本的な表現の組合せによって、オブジェクト集合を可視化するための手法である。そして、もう1つは、ユーザの与える可視化目的に応じて、自動的に表現の組合せ方法を決定するための手法である。当然ながら、両者は独立ではなく、密接に関連し合っている。以下、4章と5章では、それぞれの手法について順に述べる。

4. オブジェクト集合の可視化手法

本章では、オブジェクトの集合を3次元空間の図形として可視化するための手法について述べる。本研究の目標とするシステムを実現するため、我々は、基本的な表現の組合せによって、さまざまな可視化技法を実現する。従来のモジュールベースの可視化環境では、自由度の高い可視化が実現できる反面、その組み合わせ方が難しく、システムが自動的に可視化方法を決定することが困難である。我々は、最低限の表現の組合せによって可視化を実現することで、多様な可視化を実現しつつ、システムによって自動的に可視化方法を決定することを可能としている。

本章では、まず、ユーザが可視化目的を指定する際に、データに対して適用する操作の種類を定義する。次に、既存の可視化技法をデータ種別や操作ごとに体系化し、それらの特徴付けている基本的な表現の要素を抽出する。その後、その基本的な表現を組み合わせることによって、さまざまな可視化技法や、その組合せを実現するための手法を提案する。最後に、本手法を用いて可視化を行う際の、オブジェクト集合と図形との対応関係をテーブル形式で記述する方法を提案する。

4.1 可視化操作の定義

一般に、情報可視化において、可視化対象となるデータは、値データ(values)と構造データ(structures)の2種類に大別される^{6),23)}。本研究の対象とするオブジェクト指向データベースでは、値データはオブジェクトの属性に対応し、構造データはオブジェクトの関連に対応すると考えられる。

我々は、値データや構造データに対して適用する操作として、Wehrend分類²⁴⁾やGADGET/IV²⁾を参考に以下の11種類の操作を定義した。

- Identify …… 属性値を表現
- Distribute …… 属性値の分布状況を表現

- Locate 地理的に配置
- Correlate 他の属性値との関連を表現
- Order 属性値で整列
- Compare 属性値を比較
- Categorize その属性ごとに分類
- Summarize その属性で集計
- Connect 関連を表現
- Hierarchy 階層を表現
- Associate 相関関係を表現

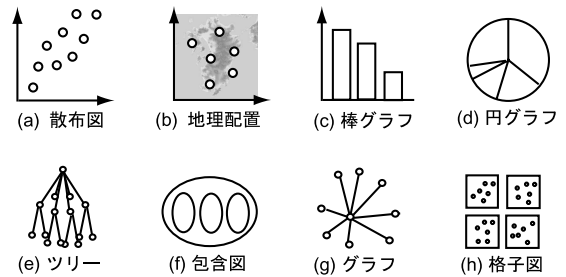


図2 代表的な可視化技法

Fig. 2 Typical visualization methods.

上記の操作のうち、{Identify, Distribute, Correlate, Locate, Compare, Order, Categorize, Summarize} が値データ（属性）に対して適用可能な操作であり、{Connect, Hierarchy, Categorize, Associate} が構造データ（関連）に対して適用可能な操作である。我々は、基本的には Wehrend 分類²⁴⁾や GADGET/IV²⁾で定義されているすべての操作をカバーするように上記の操作を定めた。ただし、既存の手法では構造データをすべて Structure として一括して扱っているのに対して、本手法では構造データを 1 対 1, 1 対多, 多対 1, 多対多という関連の種類によって分類して扱う。そこで、それぞれの構造の表現に適した 4 種類の操作を別に定義した。また、既存の手法では配列データを可視化の対象としているのに対して、我々の手法ではオブジェクトの集合を対象としている。そのため、本手法では、GADGET/IV で定義されている Reveal change of (データの変化を可視化) のような操作は定義されない。同様の操作は、本手法では、時間属性と値属性に対して、別々に Order や Identify などの操作を適用することによって実現される。

4.2 既存の可視化技法の体系化

本節では、可視化技法に関する文献^{2),7),10),11),23)}を参考に、代表的な既存の可視化技法を取り上げ、データの種類と可視化の目的という観点から、これらの可視化技法を分類し、それらの特徴付けている基本的な表現の要素を抽出する。以下、値データを可視化するための技法と構造データを可視化するための技法について順番に述べる。

まず、値データを可視化するための技法について述べる。属性値の分布状態を把握したい (Distribute)、また、他の属性との相関関係を把握したい (Correlate)、といった目的に対しては、2 章で述べたような散布図^{3),5)} (図 2 (a)) による手法が有効である。散布図では、オブジェクトの属性値とオブジェクトに対応するノードの座標を対応付けて描画することで、属性値の分布状態や他の属性値との相関関係を把握することができる。また、図形の色や形などの属性も使

用することにより、さらに多次元の属性の相関関係を表現することが可能である。散布図において、特にオブジェクトが地理的な位置属性を持つ場合は、地理座標に基づいて地図上にマッピング (Locate) するのが効果的である (図 2 (b))。一方、値の大きさを比較したい場合 (Compare) には、BarChart¹⁰⁾ (図 2 (c)) や PieChart¹⁰⁾ (図 2 (d)) などの図形の高さや面積によって値を表現する手法が有効である。

次に、オブジェクト間の構造を可視化するための技法について述べる。構造データの階層的に可視化する (Hierarchy) ための技法としては、ConeTree¹⁸⁾ に代表されるツリー表現 (図 2 (e)) が一般的である。また、データ間の包含関係を表現したり、属性値に応じて分類したり (Categorize) ための手法としては、TreeMap¹³⁾ に代表される、包含図 (図 2 (f)) が利用できる。また、ネットワーク情報などの接続関係を可視化したい場合 (Connect) は、グラフ表現⁴⁾ (図 2 (g)) がよく用いられる。グラフ表現は、また、データ間の相関度を表現する (Associate) ためにも利用できる。

また、上記のように、値データや構造データに対する可視化手法を直接適用するだけではなく、値データを構造として表現したり、逆に、構造データを値として表現したりすることも可能である²³⁾。たとえば、値データをその値に応じて分類することで、その関係を包含図形 (図 2 (f)) によって表現することができる¹⁰⁾。また、格子図 (図 2 (h)) によって、複数の可視化結果を同時に表示する手法²²⁾ も用いられる。一方、構造データに対しては、関連の多重度やツリーの深さなどを値データとして見ることで、値データに対する技法を適用することができる。

4.3 基本的な表現要素の組合せによる可視化の実現

本手法では、オブジェクトやオブジェクト集合を 3 次元空間の 1 つの図形として描画することで、可視化を実現する。オブジェクトから図形の変換方法は、基本的に、あるクラスに属する、オブジェクト集合ごと

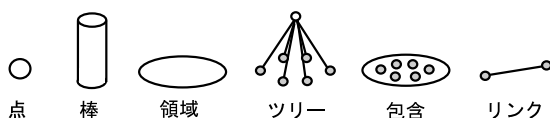


図3 図形の種類
Fig. 3 Types of figures.

に定義される。我々は、前節で述べた既存の可視化技法を特徴付けている要素として、図形の種類、図形間の関連表現、図形の水平方向の配置方法、そして、図形の描画属性に注目し、それらを基本的な表現の要素として利用することとした。以下にそれぞれについて順に述べる。

まず、図形の種類として、点・バー・領域・ツリー・包含・リンクの6種類を用いる(図3)。散布図による可視化環境では、図形として点のみが使用される。しかし、値の分布状況ではなく、値の大きさを表すためには、図形の高さや面積による表現が有効であるため、棒グラフや領域の図形を利用する。また、ツリー・包含・リンクの3種類の図形は、他の複数の図形との関連を表すための特別な図形である。これらの図形は、他の図形を子図形として持つ。ツリーは、点図形の特殊な図形で、自分自身の下に子図形を配置する。包含は、同じく、領域図形の特殊な図形で、内側に子図形を配置する。リンクは、2つの子図形をラインによって接続する。ツリーや包含の子図形は、親図形に対して相対座標系で配置される。

次に、図形の水平方向の配置方法について述べる。本手法では、3次元空間に対して可視化を行う。しかし、効果的な可視化を実現するため、すべての軸を同様に扱うのではなく、垂直方向と水平方向を区別して扱うこととした。水平方向の配置によって値の分布状況や平面的な構造を表現し、垂直方向の配置や高さによってオブジェクトの値の大きさやツリーなどの階層構造を表現する。水平配置の方法として、散布配置・格子配置・円状配置の3種類を用いる(図4)。散布配置では、先述の散布図の手法で述べたように、オブジェクトの属性値に応じて図形を配置する。さらに、図形を格子状に配置する格子配置、極座標系で図形を配置する円状配置を使用する。

図形の種類と図形の水平配置方法に加え、実際に図形を描画するための描画属性として、位置・サイズ・色・ラベルの4種類の属性を使用する。これらの描画属性のそれぞれに対して、オブジェクトの属性を割り当てることで、オブジェクトを図形として描画する。ただし、属性値のとりうる範囲は、属性の種類によって異なるため、その範囲に応じて正規化し、描画属性



図4 配置方法の種類

Fig. 4 Horizontal arrangement methods of figures.

に割り当てる。また、座標やサイズなどの一部の項目は省略可能であり、これらの指定が省略されたときには、自然な配置になるように適当な位置やサイズが決定されるものとする。具体的には、円状配置の角度・半径や包含図形のサイズが省略された場合は、適当な最小サイズが計算されて使用される。高さについては、図10の例のように、複数のクラスが重ならないように、それぞれのクラスに対して適当な範囲が割り当てられる。また、色相に関しては、複数のクラスを可視化する場合、それぞれのクラスに対して色相の範囲が割り当られ、クラスごとに色分けして表示される。

これらの図形の種類と配置方法の組合せにより、前節で述べたようなさまざまな既存の表現方法を実現したり、それらを組み合わせたりすることが可能となる。たとえば、ConeTreeは、ツリーと円状配置を組み合わせることによって実現される。また、PieChartは、領域と円状配置の組合せにより実現される。しかし、このような組合せの方法をユーザが直接指定することは困難であるため、本システムでは、ユーザが指定した可視化目的から、システムが自動的に適切な表現の組合せを決定する。

4.4 可視化方法の記述形式

本節では、前節で述べた可視化手法によってオブジェクト集合を可視化するとき、オブジェクトと描画図形との対応関係を表すための記述手法を提案する。本記述法を用いることで、オブジェクトから図形への変換を明確に定義できる。また、オブジェクトと図形の対応関係をユーザに示すことで、可視化結果に対するユーザの理解を深めるために利用できる²⁾。

可視化の表現方法を形式的に記述するための試みとしては、Cardらによって提案された方法⁷⁾がある。Cardらは、可視化技法で用いられるデータ配列と描画図形との関係をテーブル形式で記述するための手法を提案している。しかし、Cardらの手法は、既存の可視化技法でどのような描画属性が利用されているかを分析し、既存の可視化技法を体系化することを主な目的としている。そのため、Cardらの記述方法には表現にあいまい性があり、データ配列から図形への変換方法は明確には定義されない。したがって、それを可視化システムで活用することはできなかった。我々は、

表 1 可視化方法の記述形式

Table 1 Description form for the representation of a visualization method.

No	Class Var.	Filter	GroupBy	Type	Rel.	Child	Layout	X, NX, A	Y, NY, R	Z, H	Size	Shape	Ch	Cs	Cv	Label
0	クラス変数	条件	属性*	種類	関連	No*	配置法	属性	属性	属性	属性	属性	属性	属性	属性	属性
1	クラス変数

データと描画属性の対応関係をテーブル形式によって記述するという Card らの基本的なアイデアのみを採用し、オブジェクトから描画図形への変換を明確に記述できる新たな記述方法を定めた。本手法では、一般的な可視化技法を対象とするのではなく、記述の対象を 4.3 節で提案した可視化手法に絞ることによって、あいまい性のない記述方法を実現した。また、Card らの記述方法が配列データを可視化の対象としているのに対し、我々の記述方法はオブジェクト集合を対象としている。

我々の提案する可視化表現の記述方法の形式を表 1 に示す。また、具体的な記述例として、6 章で示す可視化の適用例の、図 8~11 と表 4~7 を適宜参照している。表 1 において、表の各行は 1 つのオブジェクト集合に対応し、その属性と描画図形の属性との関係を表している。以下、各列の項目について順番に説明する。

まず、No は、表の行番号を示す。Class + Variable は、クラス名と変数名の組により、対象とするオブジェクト集合を示す。ここには、OQL で使用された変数や、データベース内の任意のオブジェクト集合が指定可能である。指定可能なオブジェクト集合の詳細については、5.1.1 項で述べる。

次に、Filter や GroupBy の項目は、一部のオブジェクトのみを図形に変換したり、複数のオブジェクトをグループ化して 1 つの図形として描画したりしたい場合に使用する。選択条件 Filter に条件を記述することにより、条件を満たすオブジェクトのみが図形に変換される。選択条件 Filter は、たとえば、表 7 のツリー表現による可視化例のように、同一クラスのオブジェクトであっても、ルートであるか非ルートであるかといった条件によって可視化方法を変更するために使用する。また、表 6 の具体例のように、集約属性 GroupBy に属性名を指定することで、指定された属性値が同一となる複数のオブジェクトがグループ化され、全体で 1 つの図形として可視化される。Filter と GroupBy は同時に指定することも可能であり、同時に指定された場合には、選択条件を満たすオブジェクトのみが集約される。これらの機能は、上記の例のように、特定の可視化表現を実現するために利用する。

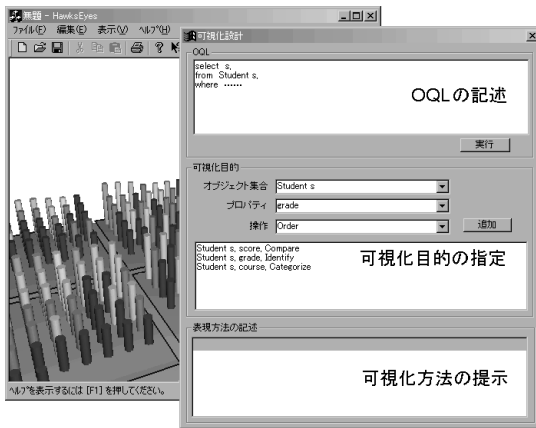
本システムでは、これらの条件はシステムによって自動的に設定されるため、基本的にユーザが直接指定することはない。ユーザが、集約や選択を用いた複雑な問合せを行いたい場合には、OQL の機能を使用する。

次に、Type と Layout の項目には、図形の種類と水平配置の方法を指定する。Type には、図形の種類として、{点, パー, 領域, ツリー, 包含, リンク} のどれか 1 つを指定する。また、Layout には、水平配置の方法として、{散布配置, 格子配置, 円状配置} のどれか 1 つを指定する。図形の種類がツリー, 包含, または、リンクの場合は、子図形となる他のオブジェクトとの関係を Relation, Child に記述する。Relation には関連を表す関連属性名を記述し、Child には対応する行の番号を指定する(表 4, 表 7 の具体例を参照)。なお、親図形に対して選択条件 Filter が設定されている場合は、子図形として可視化されるオブジェクトのうち、対応する親図形が選択されるもののみが可視化の対象となる。

Layout より後の各項目には、図形の描画属性とオブジェクトの属性の対応関係を指定する。水平座標は、水平配置の方法に応じて、X, Y, または、NX, NY, または、A, R の組で指定する。X, Y は散布配置の座標、NX, NY は格子配置の配置順序、A, R は円状配置の角度と半径である。Z は垂直方向の配置位置、H は図形がパーの場合の高さを表す。また、Size は図形の大きさ・面積、Shape は図形の形状、Ch, Cs, Cv は色相・彩度・明度、Label は図形の横に表示する文字列を表す。それぞれのセルには、基本的にクラスの属性名を記述する。オブジェクトを図形に変換するとき、各セルに記述されたオブジェクトの属性値が、図形の描画属性として使用される。ただし、オブジェクトが集約条件 GroupBy によってグループ化されている場合、対応する属性が複数存在するため、描画属性として使用する 1 つの属性値を特定することができない。そこで、表 6 の具体例のように、ODMG3.0 で定義されている集約演算子 {count, min, max, sum, avg} を使用することで、属性値を指定する。

5. 半自動可視化環境の実現

本章では、ユーザから可視化目的が与えられたとき、



可視化結果の表示画面 (左) 可視化方法の設定画面 (右)

図5 インタフェース設計

Fig. 5 Design of user interface.

その目的を実現するように表現方法の組合せを自動的に決定するための手法について述べる。まず、我々の行った可視化目的の指定インタフェースの設計と、可視化目的としてユーザが選択可能なオブジェクト集合や属性について説明する。次に、可視化目的から可視化方法を導出するための手法として、属性・関連の種類を分類し、それぞれの属性・関連に対して適用可能な操作と、各操作が選択されたときの可視化の実現方法について述べる。

5.1 ユーザインタフェース設計

我々は、現在開発中のプロトタイプシステムのユーザインタフェースとして、図5に示すような2つのウィンドウを設計した。一方のウィンドウは可視化結果の表示画面であり、もう一方のウィンドウは、可視化目的の指定や表現方法の記述の提示などを行う可視化方法の設定画面である。設定画面は、3つのエリアからなる。まず、1番上のエリアはOQL記述フォームであり、ユーザはここにOQLを記述する。中央のエリアは、可視化目的を指定するためのリストボックスである。可視化目的の追加方法については次節で述べる。そして、一番下のエリアには、可視化方法が4.4節で述べた記述法を用いて表示される。このエリアにはユーザの入力は行われず、オブジェクトと図形の対応関係をユーザに示すために使用される。

5.1.1 可視化目的の指定

可視化の目的を指定するため、ユーザは、図6のようなリストボックスを用いて、オブジェクト集合、プロパティ、操作の3項目を順番に選択する。以下に、それぞれの項目で選択可能な要素について、順に説明する。

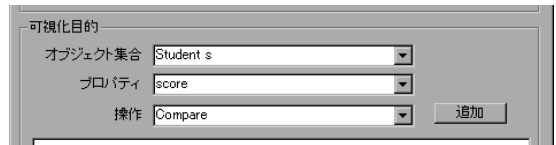


図6 可視化目的の指定インタフェース

Fig. 6 User interface for indicating a visualization goal.

オブジェクト集合の選択 ユーザは、まず、可視化目的を追加するオブジェクト集合を選択する。ここで、選択可能なオブジェクト集合は、以下の3種類である。

- OQLの出力結果のオブジェクト集合
OQLの出力結果は、データベースにすでに存在するオブジェクトの集合である場合もあるし、OQL内で定義された構造体のオブジェクト集合である場合もありうる。どちらの場合であっても、OQLの出力結果を可視化対象のオブジェクトとして指定できる。
- OQL (From 節) で定義されたオブジェクト集合
OQLの出力結果のオブジェクトだけではなく、OQLの内部 (From 節) で定義されたオブジェクトも、クラス名 + 変数名の組の形式で指定できる。Where 節の条件によってオブジェクトの条件が指定されている場合は、その条件を満たすオブジェクトのみが可視化の対象となる。なお、OQLの内部で生成された複合オブジェクトは、名前を持たず、指定することが難しいので、可視化の対象とはしていない。
- データベースに存在するオブジェクト集合
OQLで使用されたオブジェクトのほかに、データベース内に格納されているエクステントや、任意のオブジェクト集合も可視化の対象とすることが可能である。クラス名 + 集合変数名の組で指定する。

すでに図形として可視化されているオブジェクト集合に対して別の可視化目的を追加することも可能であり、また、新たなオブジェクト集合に対して最初の可視化目的を追加することも可能である。1つのオブジェクト集合に対しては、対応する図形の自由度の許す範囲で、複数の目的を指定することが可能である。新しい可視化目的と以前の可視化目的を同時に実現することができない場合は、以前の可視化目的はキャンセルされる。

プロパティの選択 可視化目的を追加するオブジェクト集合を選択したら、次に、そのクラスの1つのプロパティを指定する。オブジェクト集合を選択したとき、必ずそのオブジェクト集合のクラス名も決定されるた

め、そのクラスに応じて指定可能な属性の候補が決定される。そのオブジェクト集合が実際にはサブクラスのオブジェクトを含む場合でも、サブクラスのプロパティは指定することはできない。ただし、下記に示すように、仮想関数を指定することは可能である。OQLの出力結果として、既存のあるクラスのオブジェクト集合ではなく、OQLで定義されたクラスのオブジェクト集合が出力された場合であっても、その属性情報はOQLで定義されるため、その任意の属性を指定することができる。選択されたクラスのメンバのうち、以下に示すものが選択可能であるとする。

- 属性、関連
- 集合の属性 + 集約演算子
属性が集合 (bag, set, array, list, etc) である場合、その属性名を指定しただけでは、複数の属性値のうち描画に使用されるものを特定することができない。そこで、集約演算子 (count, min, max, sum, ave) を使用することで、集合から1つの値を計算する。
- 構造体の属性 + その構造体の属性
指定した属性が構造体である場合は、さらに、その構造体の1つの属性を指定する。
- 引数なしのメソッド
メソッドが read-only で引数を必要としない場合、そのメソッドもプロパティと同様に指定可能とする。メソッドの戻り値によって、上記の属性や関連として扱う。

また、指定したオブジェクト集合が、すでに集約条件付きで可視化されている場合は、集合属性が指定されたときと同様に、属性名 + 集約演算子の組によって属性を指定する。

操作の選択 最後に、これまでに選択したオブジェクト集合とそのプロパティに対して、適用する操作を指定する。我々は、操作の種類として、4.1節で述べた {Identify, Distribute, Locate, Correlate, Order, Compare, Categorize, Summarize, Connect, Hierarchy, Associate} の11種類の語彙を定義した。これらの操作のうち、ユーザの指定したプロパティの型情報をもとに、適用可能な操作のみが選択肢としてユーザに提示される。各プロパティに対して適用可能な操作の条件については、5.2節で述べる。

5.1.2 可視化結果のブラウジング

ユーザは、可視化画面内に表示された空間を観察することによって、検索結果を分析することができる。我々が開発中のプロトタイプシステムでは、現在の、視点移動とズームの機能のみを提供している。ユー

ザがより効率的な分析作業を行うためには、インタフェースに関する既存の研究²⁰⁾でもあげられているように、ユーザの作業履歴を保存・再利用するための機能や、あるオブジェクトに対応する図形を選択したとき、そのオブジェクトの詳細な情報や関連オブジェクトを表示するような機能が必要であると考えられる。

5.2 可視化目的から可視化方法の導出手法

本節では、ユーザによって可視化目的が指定されたとき、その可視化目的を実現するための可視化方法の導出方法について述べる。我々は、属性・関連を可視化に適した型で分類し、それぞれの型に対して選択可能な操作とその条件、そして、可視化目的が追加されたときの可視化表現の変更方法を定めた。ユーザの指定したプロパティが属性である場合と関連である場合によって、型の分類や操作の適用方法が大きく異なるため、それぞれについて順に述べる。

5.2.1 属性の分類

先述のように、クラスの属性情報は、ODMG3.0で定義されるスキーマインタフェースにより取得できる。しかし、ODMG3.0で定義されている整数型や浮動小数点型の属性は、言語仕様上の都合によって定義されており、その分類は可視化を考えるうえで適しているとはいえない。そこで、属性をその性質によって、Nominal, Ordinal, Quantitative, Time, Geographical の5種類に分類する。以下に、それぞれの型について説明する。

まず、一般的な属性を、Nominal型、Ordinal型、Quantitative型の3種類に分類する。この分類方法は、その属性に対して適用可能な演算の性質に基づくものであり、既存の可視化技法の分類でも広く利用されているものである^{7),16)}。Nominal型は同値比較(=)のみが可能な型、Ordinal型は大小比較(<, >)が可能な型で、Quantitative型は大小比較に加えて足し算や掛け算などの演算(+, -, ×, ÷)が適用可能な型である。以下に、これらの型と、ODMG3.0で定義される型の対応関係を述べる。まず、Nominal型には、列挙型、文字列型、または、整数型が相当する。そして、Ordinal型とQuantitative型には、整数型や浮動小数点型の数値型全般が対応する。なお、文字列型も、辞書式順序を用いることでOrdinal型として扱うことが可能である。さらに、ユーザによっては、数値型を列挙型の用途に使用している場合なども考えられるため、スキーマから得られる情報だけでは、ある属性がNominal型、Ordinal型、Quantitative型のどの型として扱われているかを完全に判定することはできない。そこで、今のところは、基本的には上記の

表 2 属性に対して適用可能な操作と、使用される描画属性の優先順位

Table 2 Available actions for each type of attribute and order of applied rational attributes.

操作	Nominal	Ordinal	Quantitative	Time	Geographical
Identify	Label, Ch,Sh,NX,NY,A	NX,X,Y,A,Z,Ch,Cs,Si,Cv	H,R,Si,X,Y,Ch,Cs,Cv	X,Y,A,Z	X,Y
Distribute	NX,NY,A	X,Y,A,Z,R	X,Y,A,Z,R	X,Y,A,Z	X,Y
Locate					X,Y
Correlate	Ch,Sh,NX,NY,A	X,Y,A,Z,Ch,Cs,Si,Cv	X,Y,Z,R,Si,Ch,Cs,Sv	X,Y,A,Z	X,Y
Order		NZ,X,Y,A,Z	NX,X,Y,A,Z	NX,X,Y,A,Z	
Compare			H,R,Si		
Categorize	○			○	
Summarize	○			○	

分類を使用し、必要に応じてユーザに型を明示的に指定してもらうか、あるいは、クラスを実装しているプログラムコードがユーザにより提供されている場合には、その中で適用されている演算を調べ、その属性が Nominal 型か Ordinal 型か Quantitative 型を判定する方針としている。

次に、Time 型と Geographical 型は、それぞれ、時間属性と地理属性を表す特殊な型である。時間属性や地理属性は表現方法を決定する上で非常に重要であるため^{2),7),10)}、これらは特別な型として扱うこととした。Time 型は、時間属性を表す Ordinal 型の特殊な型であり、ODMG3.0 の Time 型、Date 型、Timestamp 型が対応する。Geographical 型は、地理属性(緯度・経度)を表す Quantitative 型の特殊な型である。しかし、ODMG3.0 では地理属性を表す標準型は定義されていない。そこで、Geographical 型に対応する型は、アプリケーションで定義されているものとする。

以上は、1つの値を持つ属性の分類について述べた。しかし、オブジェクトの属性として、値の集合(bag, set, array, list, etc)が定義されていることもありうる。この場合、4.4節で述べたように、集合に対して集約演算子を適用することで、Quantitative 型の属性として扱う。このとき、集合の属性に応じて、適用可能な集約演算子が異なる。Quantitative 型の集合には、ODMG3.0 で定義されている集約演算子のすべて {count, min, max, avg, sum} が適用可能である。また、Ordinal 型(Time 型を含む)の集合には、集約演算子 {count, min, max} のみが適用可能である。その他の型の集合には、集約演算子 count のみが適用可能である。また、集約演算子 count は、関連に対しても適用可能であり、関連の多重度を値データとして利用できる。

5.2.2 属性に対する操作の適用

属性に対して指定可能な操作は、{Identify,

Distribute, Correlate, Locate, Compare, Order, Categorize, Summarize} の 8 種類である。適用可能な操作は属性の種類によって異なる。表 2 に、各操作が適用可能な型を示す。表 2 の各セルは、ある属性と操作が指定されたときに、その属性に割り当てる図形の描画属性と、その優先順位を表している。各セル内に記載されている描画属性は表 1 と対応しており、また、左側の描画属性ほど優先度が高いことを示している。このとき、どの描画属性を割り当てるかどうかの基準として、Mackinlay による研究成果¹⁶⁾を参考とした。Mackinlay は、Nominal 型、Ordinal 型、Quantitative 型のそれぞれの属性値を表現するために、どの描画属性を用いるのが適しているか、という課題に関して研究を行った。しかし、Mackinlay の研究では、可視化の目的までは考慮されていなかった。そこで、我々は、Mackinlay の研究成果を参考に、可視化目的の概念を導入し、表 2 に示すように、それぞれの型 + 操作の組合せに対して、利用する描画属性の順序付けを行った。

可視化目的として、属性と操作が指定されると、表 2 に記載されている優先順位に従って、未使用の描画属性に対して指定された属性を割り当てる。割当て可能な描画属性がすべて使用されている場合、以前に指定した可視化目的の属性がキャンセルされ、新しく指定した属性が割り当てられる。ただし、図形の種類や配置方法を変更することによって、以前の可視化目的を保ちつつ別の描画属性に割り当てることが可能な場合は、図形の種類や配置方法を自動的に変更する。なお、Compare が指定された場合には、H, R, Si にその属性を割り当てるだけでなく、Order も同時に適用する。また、Categorize や Summarize は、構造的な可視化方法を適用するための操作である。Categorize が指定された場合は、図 8 の可視化例のように、後述する関連に対する Categorize と同様に、その属性値

表 3 操作の適用可能な関連属性
Table 3 Available operations for each type of relationship.

操作	1対1	1対多	多対1	多対多
Connect	○	○	○	○
Hierarchy	○	○		
Categorize			○	
Associate				○

に基づく包含関係を表示する。また、Summarize が指定された場合は、図 10 の可視化例のように、その属性に基づき集計を行う。今のところ、Categorize や Summarize の操作は、Nominal 型と Time 型に対して適用可能としている。ただし、Time 型に関しては、たとえば、月ごとに分類・集計を行うか、週ごとに分類・集計を行うかといった、単位を指定するための追加インターフェースが必要になると考えている。

5.2.3 関連の分類

関連の種類を、関係の多重度によって、1対1、1対多、多対1、多対多の4通りに分類する。ODMG3.0では、多の関連を表現するために、集合かリストのどちらかを利用できる。しかし、今のところ、本手法では、関連がリストであるか集合であるかは区別していない。

5.2.4 関連に対する操作の適用

関連には、その種類に応じて、{Connect, Hierarchy, Categorize, Associate} の4種類の操作が適用可能である。それぞれの関連に対して、適用可能な操作の関係を表 3 にまとめる。Connect が指定された場合には、2つのオブジェクトを表す図形の間をリンクで接続することにより、オブジェクト間の関連を表現する。Hierarchy は、自分自身をツリーで表し、関連するオブジェクトに対応する図形を子図形として配置することで、階層関係を表現する。また、Categorize は、同様に、包含図形を用いて多対1の包含関係を表現する。Associate が指定された場合は、Connect と同様に関連を持つ図形どうしをリンクによって接続し、複数の関連を集約して、オブジェクトどうしの相関度をリンクの太さによって表現する。

6. 適用例

本章では、我々が開発中のプロトタイプシステムによって作成した、本手法の具体的な適用例を示す。現在、本プロトタイプシステムは、ODMG3.0のC++ Bindingの利用を前提に、Windows上でC++を用いて実装を行っている。ただし、現状では、まだOODBMS

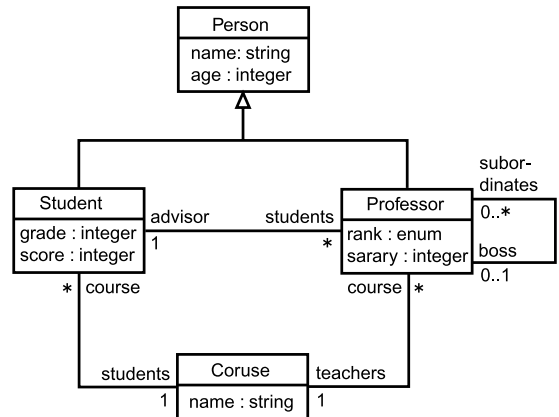


図 7 適用例のオブジェクトモデル図
Fig. 7 The object model diagram of the example.

は使用しておらず、非永続オブジェクトを可視化している。現在までに、オブジェクトから図形への変換処理と、可視化画面の表示が実現されている。図形の描画には OpenGL、ユーザインタフェースは MFC を使用して実装している。

我々は、今回の可視化例を作成するため、ODMG3.0の仕様書⁸⁾に記載されているスキーマ定義例をもとにデータベースを設計し、適当なサンプルデータを生成して可視化を適用した。このデータベースは、大学の教官・学生・課程・講義などをモデル化したものである。そのスキーマ定義のうち、本論文で示す可視化例で使用されている部分のみを取り出したオブジェクトモデル図を図 7 に示す。本章では、まず、このデータベースの検索結果に対して対話的な可視化を行った適用例を示す。次に、その他の可視化の実現例として、いくつかの可視化結果画面と、それらに対応する、4.4 節で述べた形式での可視化方法の記述を示す。

図 8 は、“Select s, From Student s, Where ...” といった OQL の検索結果として Student オブジェクトの集合が得られたとき、可視化目的を逐次的に追加し、対話的な可視化を行った例である。以下に、追加した可視化目的と、実現された可視化結果を、順番に示す。

- Student s の score を Compare
Student オブジェクトを表すバーが、格子配置によって表示される。score がバーの高さと、格子配置の配置順序に割り当てられている。
- Student s の grade を Identify
grade が色相 H に割り当てられ、バーが grade ごとに色分けされて表示される。
- Student s を course で Categorize

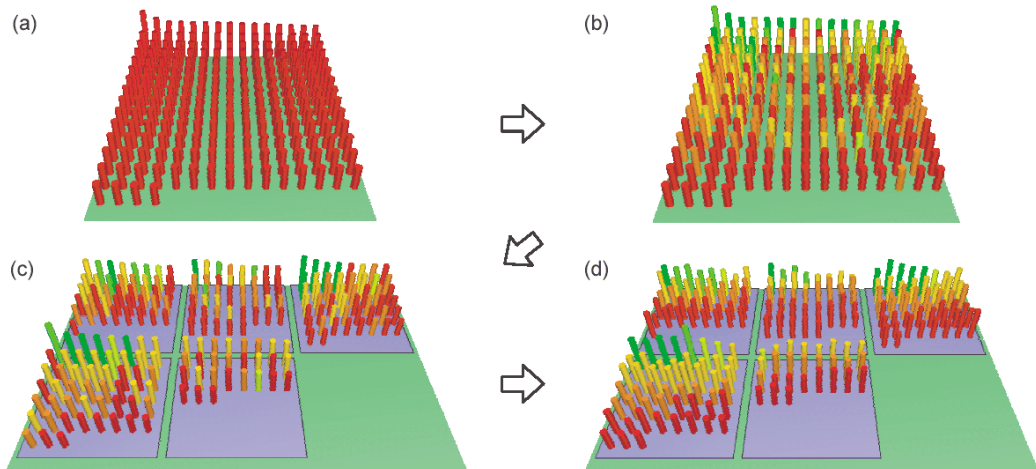


図 8 対話的な可視化の例
Fig. 8 An Example of interactive visualization.

表 4 図 8 の対話的な可視化例 (d) に対応する可視化方法の記述
Table 4 Description of the visualization of the example of Fig. 8 (d).

No	Class Var.	Filter	GroupBy	Type	Rel.	Child	Layout	X,NX,A	Y,NY,R	Z,H	Size	Shape	Ch	Cs	Cv	Label
0	Student s			バー	—	—	格子	Grade		score			grade			
1	Course			包含	students	0	格子									

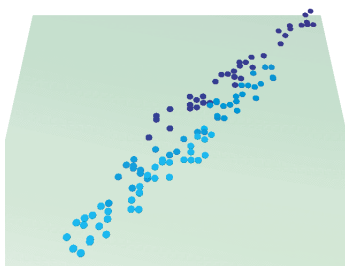


図 9 可視化例 1
Fig. 9 Visualization Example 1.

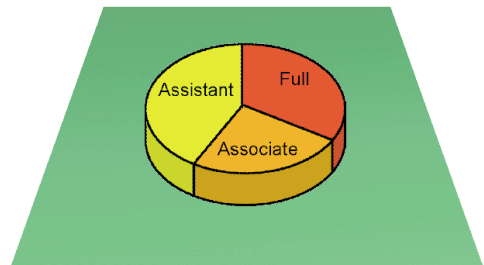


図 10 可視化例 2
Fig. 10 Visualization Example 2.

Course が包含図形として可視化され, Student は Course の子図形として, Course の内部に配置される. Course は, デフォルトの格子配置で配置される.

- (d) Student s の grade を Order 格子配置の順番に割り当てられている score が grade に上書きされ, grade の順番で配置される. 表 4 は, この図 8(d) の可視化方法を, 4.4 節で述べた記述形式を用いて表現したものである.

また, その他の可視化結果の例を, 図 9 ~ 11 に示す. これらの可視化結果は, それぞれ, 表 5 ~ 7 の可視化表現に対応している. 図 9 は, 主に散布図的な可視化の例である. 可視化目的として, Professor の各属性に対して Correlate の操作を指定することで, 属

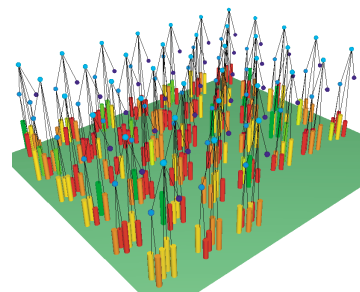


図 11 可視化例 3
Fig. 11 Visualization Example 3.

性間の相関関係が図 8 のような散布図によって可視化される. この可視化結果より, たとえば, ポストの高い教官は, 年齢の高い範囲に分布しており, また, 年

表5 図9の可視化例1に対応する可視化方法の記述

Table 5 Description of the visualization of the example of Fig. 9.

No	Class Var.	Filter	GroupBy	Type	Rel.	Child	Layout	X,NX,A	Y,NY,R	Z,H	Size	Shape	Ch	Cs	Cv	Label
0	Professor p			点	-	-	散布	age	salary				rank			

表6 図10の可視化例2に対応する可視化方法の記述

Table 6 Description of the visualization of the example of Fig. 10.

No	Class Var.	Filter	GroupBy	Type	Rel.	Child	Layout	X,NX,A	Y,NY,R	Z,H	Size	Shape	Ch	Cs	Cv	Label
0	Professor p		rank	領域	-	-	円状				sum (salary)		rank			

表7 図11の可視化例3に対応する可視化方法の記述

Table 7 Description of the visualization of the example of Fig. 11.

No	Class Var.	Filter	GroupBy	Type	Rel.	Child	Layout	X,NX,A	Y,NY,R	Z,H	Size	Shape	Ch	Cs	Cv	Label
0	Professor p	Boss = NULL		ツリー	students	1,2	格子						rank			
1	Professor p	Boss != NULL		ツリー	students	1,2	円状						rank			
2	Student			バー	-	-	円状			score			grade			

齢の高い教官ほど給料も高い、というような理解を得ることができる。また、図10は、ProfessorをrankでCategorizeした例である。その結果、Professorがrankごとに集計されて、それぞれのsalaryの合計が円グラフで可視化されている。図11は、Professorのstudentsに対してHierarchyを適用した結果、ツリーによって、ProfessorとStudentのオブジェクト間の関連が表現されている。

7. まとめと今後の課題

本論文では、オブジェクト指向データベースのための対話的な汎用的な可視化環境を提案した。半自動的な可視化を実現するためのユーザ支援環境は、近年の情報可視化の分野において非常に重要な課題である^{1),2)}。我々の提案するシステムを利用することにより、可視化技法に関する十分な予備知識を持たないユーザであっても、さまざまな可視化方法を対話的に行うことができる。本システムは、データベースのメタ情報をスキーマ定義から自動的に取得するため、ユーザの設定作業を必要とせず、また、汎用的なオブジェクト指向データベースに適用できるという点で、非常に有用であると考えられる。

今回は、表現が複雑になるのを避けるため、Parallel Coordinates法¹²⁾やStick Figure法¹⁷⁾などのより多次元データのための表現技法は導入しなかった。また、今回は表示される図形の数が比較的少ない適用例を示したが、大量のデータを効率的にブラウジングするた

めには、Fish-Eye¹⁹⁾などのFocus+Context手法や、階層構造をDrill-Downして表示するインタフェースが必要となるであろう。今後の課題としては、これらの手法を導入することで本手法の表現力の拡張を図ると同時に、プロトタイプシステムを実装し、より実際のデータベースに対して本手法を適用することで、その実用性を検証していくことが必要であると考えられる。

謝辞 本研究の一部は、文部省科学研究費補助金(課題番号10308012)の援助を受けている。

参考文献

- 磯部成二, 黒川 清, 塩原寿子, 飯塚哲也: 視覚化による多次元データ分析システム: INFOVISER, 情報処理学会論文誌, Vol.40, No.5, pp.2417-2428 (1999).
- 古畑理香, 藤代一成, 市川哲彦, 竹島由里子: GADGET/IV: 情報可視化の半自動設計支援環境, 第11回データ工学ワークショップ (DEWS 2000) (Mar. 2000).
- Ahlberg, C. and Wistrand, E.: IVEE: An Information Visualization and Exploration Environment, *Proc. International Symposium on Information Visualization*, pp.66-73 (1995).
- Becker, R.A., Eick, S.G. and Wilks, A.R.: Visualizing Network Data, *IEEE Trans. Visualization and Computer Graphics*, Vol.1, No.1, pp.16-28 (1995).
- Becker, R.A., Cleveland, W.S. and Shyu,

- M.J.: The Visual Design and Control of Trellis Display, *Journal of Computational and Graphical Statistics*, Vol.5, No.2, pp.123-155 (1996).
- 6) Bertin, J.: *Graphics and Graphic Information Processing*, deGruyter Press, Berlin (1977).
 - 7) Card, S.K. and Mackinlay, J.: The Structure of the Information Visualization Design Space, *Proc. IEEE Information Visualization '97 Symposium*, pp.92-99 (Oct. 1997).
 - 8) Cattell, R.G.G., Barry, D.K., et al.: *The Object Database Standard: ODMG 3.0*, Morgan Kaufmann Publishers (2000).
 - 9) Chaudhuri, S. and Dayal, U.: An Overview of Data Warehousing and OLAP Technology, *SIGMOD Record*, Vol.26, No.1 (1997).
 - 10) Chen, C.: *Information Visualization and Virtual Environments*, Springer-Verlag (1999).
 - 11) Fujishiro, I., Takeshima, Y., Ichikawa, Y. and Nakamura, K.: GADGET: Goal-Oriented Application Design Guidance for Modular Visualization Environments, *Proc. IEEE Visualization '97*, pp.245-252 (Oct. 1997).
 - 12) Inselberg, A. and Dimsdale, B.: Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry, *Proc. IEEE Visualization '90*, pp.361-375 (1990).
 - 13) Johnson, B. and Shneiderman, B.: Tree-Map: A Space-Filling approach to the visualization of hierarchical information structures, *Proc. IEEE Visualization '91*, pp.284-194 (1991).
 - 14) Keim, D.A.: Databases and Visualization, *Tutorial Notes of ACM SIGMOD International Conference on Management Data*, pp.1-81 (1996).
 - 15) Keller, P.R. and Keller, M.M. (Eds.): *Visual Cues - Practical Data Visualization*, IEEE Computer Society Press (1992).
 - 16) Mackinlay, J.: Automating the Design of Graphical Presentations of Relational Information, *ACM Trans. Graphics*, Vol.5, No.2, pp.110-141 (1986).
 - 17) Pickett, R.M. and Grinstein, G.G.: Iconographic Displays for Visualizing Multidimensional Data, *Proc. IEEE Systems, Man and Cybernetics*, pp.514-519 (1988).
 - 18) Robertson, G., Mackinlay, J. and Card, S.K.: Cone Trees: Animated 3D Visualizations of Hierarchical Information, *Proc. CHI '91*, pp.189-194 (1991).
 - 19) Sarkar, M. and Brown, M.: Graphical Fisheye Views, *Comm. ACM*, Vol.37, No.12, pp.73-84 (1994).
 - 20) Shneiderman, B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualization, *Proc. IEEE Symposium on Visual Languages '96*, pp.336-343 (Sep. 1996).
 - 21) Stonebraker, M., Chen, J., Nathan, N., Paxson, C., Su, A. and Wu, J.: Tioga: A database-oriented visualization tool, *Proc. IEEE '93 Visualization Conference*, pp.86-93 (Oct. 1993).
 - 22) Tweedie, L., Spence, R., Dawkes, H. and Su, H.: Externalising Abstract Mathematical Models, *Proc. CHI '96*, pp.406-412 (1996).
 - 23) Tweedie, L.: Characterizing Interactive Externalizations, *Proc. CHI '97*, pp.375-382 (1997).
 - 24) Wehrend, S. and Lewis, C.: A Problem-Oriented Classification of Visualization Techniques, *Proc. IEEE Visualization '90*, pp.139-143 (1990).

(平成 12 年 6 月 20 日受付)

(平成 12 年 9 月 27 日採録)

(担当編集委員 掛下 哲郎)



尾下 真樹 (学生会員)

1998 年九州大学工学部電気情報工学科卒業。2000 年同大学大学院システム情報科学研究科修士課程修了。現在、同博士課程在学中。コンピュータアニメーション、情報可視

化の研究に従事。



牧之内顕文 (正会員)

1967 年京都大学工学部電子工学科卒業。1970 年グルノーブル大学理学応用数学科 Docteur-Ingenieur 取得 (株)富士通 (株)富士通研究所、九州大学工学部教授を経て、1996 年九州大学大学院システム情報科学研究科教授。1996 ~ 1998 本会理事。電子情報通信学会、ACM、IEEE Computer Society、人工知能学会各会員。