

オブジェクトデータベースのための対話的可視化システム

森田互昭[†] 牧之内顕文[‡]

本論分では、オブジェクトデータベースの可視化を自動的、対話的に行うためのシステムについて提案する。近年、データベースに格納された大量データの分析を行うための可視化システムが多数開発されているが、それらの多くは可視化に専門的な知識や複雑な手間が必要であり、知識のないユーザが効果的な可視化を実現することが困難である。そこで我々は、ユーザが専門的な知識を必要とすることなく可視化を実現するシステムを提案する。当システムでは、ユーザの入力した抽象的な可視化項目から自動的に可視化の手法決定を行い可視化を行う。ユーザは自動的に得られた結果に対して修正や追加を行うことで様々な可視化を対話的に実現する。本論文ではシステムの概要と可視化の流れについて述べ、実際にシステムを用いての可視化例を示す。

An Interactive Visualization System for Object Database

Nobuaki MORITA[†] and Akifumi MAKINOUCI[‡]

This paper presents a novel system that visualize the object databases automatically and interactively. Recently, many systems to analyze the large quantity data stored in the database is developed. However, existing visualization systems require user's special knowledge and complicated establishment to visualization. Therefore, it is difficult for general users to realize visualizations. So, we propose a system that realize various visualizations without user's special knowledge and complicated establishment. System automatically decides the way of visualizations suitable for the abstract visualization goal that are given by the user and realize visualization. We describe the system construction and show the efficiency of our approach by applying the method to an example.

1、はじめに

近年、コンピュータの処理速度の向上やディスク容量の増加に伴い、データベースが大規模化し、データベースに格納された大量のデータ集合から何らかの知識を発見することへの需要が高まってきている。

大規模なデータ集合から知識を得るために、情報可視化をデータベースに適用することでデータ間の関連や分布、構造の直感的な理解を助ける、いわゆるデータベース可視化システムが近年数多く提案されている[1]。しかし、可視化の手法は数値比較のための棒グラフや分布を調べる散布図、関連を調べるツリーなど、目的に応じて数多く存在する。そのため、ユーザはその中から

目的に応じた適切な可視化手法を選択し、さらにその可視化を実現するために様々な設定を行う必要がある。結果としてユーザには可視化を行うための専門知識や複雑な手間が必要となり、予備知識の無いユーザには効果的な可視化を行うことが難しい。そこで我々は、ユーザの入力した項目から自動的に可視化手法の決定と設定を行い、専門知識や手間を必要とすることなく様々な可視化を実現するシステムを提案する。

本システムは対象をオブジェクトデータベースとしている。ユーザのオブジェクトデータベースへの問い合わせ結果に可視化を適用し、データオブジェクトを3次元空間内に図形として描画することでデータベースの可視化を実現する。

ユーザはシステムに対して、可視化の対象と操作からなる可視化項目を入力する。可視化項目を受けてシステムが最適な手法を自動的に決定し、その手法に必要な設定を行い、可視化を実現する。得られた結果に対して、

† 九州大学大学院 システム情報科学府
Department of Intelligent Systems Graduate School of Information
Science and Electrical Engineering Kyushu University
‡ 九州大学大学院 システム情報科学研究院
Department of Intelligent Systems Graduate School of Information
Science and Electrical Engineering Kyushu University

ユーザは直接その内容を変更したり、新たな可視化項目を追加することが可能になっている。自動的に実現された可視化結果を見ながら修正や追加を行うことで、専門知識や手間を必要とすることなく、様々な可視化を対話的に実現する。

本システムで可視化の対象としているオブジェクトデータベースでは、格納されたデータの属性や関連についての情報がスキーマとして明確に記述されている。スキーマ情報を可視化設定に使用することで、ユーザに対するデータタイプや関連情報の設定の手間を軽減している。

次章では関連研究として一般的なデータ集合の可視化を行うシステムについて述べる。3章、4章、5章では提案する可視化システムの概要について述べ、6章で実際にサンプルデータベースを使用しての可視化例を示す。

2、関連研究

データベース内のデータを分析し、企業の戦略的決定等へ活用するための手段として、多次元データ解析(OLAP)[14]やデータマイニングなどの分析手法が研究されている。しかし、これらの分析手法は、特定形式の大量データの特徴を把握することのみ応用範囲が限定されていた。そこで、この手法に情報可視化の概念を適用することでデータの構造や関連を視覚的に表現し、直感的な分析作業を支援する、データベース可視化システムが近年数多く提案されている。

データベースに格納されたデータのような、一般的なデータ集合を可視化するための可視化システムは、システム固有の可視化を行うシステムと、モジュールベースの可視化システムの2つに分類される。

システム固有の可視化を行うシステムには、Eureka[2]、Xgobi[4]、IVEE[3]、InfoZoom[5]などがある。これらのシステムでは散布図やテーブル形式といった、システムごとに決められた手法を用いて可視化を実現する。使用する手法がすでに決定しているため、ユーザは対象データと軸の対応付けのような設定を行うだけで可視化を実現できる。しかし、これらのシステムでは特定の可視化しか表現できないため、データ間の関連のような情報を表現することが難しいなど、ユーザの目的を完全には実現できないという問題がある。

モジュールベースのシステムの汎用的なものには、Tioga[6]やTioga2[7]、INVOVISOR[8]やAVS[15]がある。これらはデータ変換や図形配置のような処理が記述された様々なモジュールを、ユーザが目的に応じて組み合

わせることで可視化プログラムを作成し、可視化を実現する。既存のモジュールだけでなく、ユーザが新たにプログラムモジュールを追加することも可能で、設定次第で様々な可視化が実現できる。しかし、ユーザがモジュールの組み合わせによる可視化を行うためには、目的を達成するためにどんな可視化を行うべきか、その可視化を行うためにどのモジュールを組み合わせるべきか、と言った十分な予備知識が必要となる。また、可視化の度にユーザが設定を行う必要があるため、様々な可視化を試行錯誤しながら適用することが難しい。

当システムでは Wehrend の分類法[10]を元に、自動的な可視化手法の決定を行う。Wehrend によれば、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、そのデータに対する操作 $A = \{\text{Identify, Locate, Distinguish, Categorize, Cluster, Rank, Compare, Associate, Correlate}\}$ の組が与えられれば、その目的を実現する適切な技法が特定できると仮定されている。この分類をもとに、ユーザの入力した対象データの種類の、適用する操作の組から可視化手法を自動的に決定する。

既存のシステムでは可視化の対象はデータ集合単位で指定されるが、当システムではオブジェクト集合の持つプロパティ単位で可視化対象を指定し、また、可視化手法をモジュールではなく基本的な表現要素(プリミティブ)の組み合わせにより表現する。これにより、より細かい単位での可視化を行い、結果への新たな項目の追加や目的に応じた細かい修正を可能にしている。ユーザは自動的に得られた結果を見ながら修正や追加を行うことで対話的に様々な可視化を実現する。

3、可視化システム概要

提案する可視化システムは、オブジェクトデータベースへの問い合わせ結果として得られたオブジェクト集合を3次元空間に図形として配置することでデータベースの可視化を実現する。オブジェクトデータベースでは、格納された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ている。それぞれのクラスは属性や関連からなるプロパティをもっており、属性の Integer や String のような型情報や、関連の1対1、対多といった多重度に関する情報などがスキーマ内に定義される[11]。例えば、図1のような Professor, Student, Course, Subject, Person の5つのクラスからなるデータベースにおいて、クラス Professor は属性として String 型の rank、Integer 型の salary と teacher_id、上位クラス

である Person から継承した String 型の name, sex と Integer 型の age をもつ。関連としては Course クラスとの多対 1 の関連 course と Student クラスとの 1 対多の関連 students、Subject クラスとの多対多の関連 takes をもっている。これらの情報は全てスキーマに定義されており、ユーザはこれらの情報を取得して利用することが出来る。

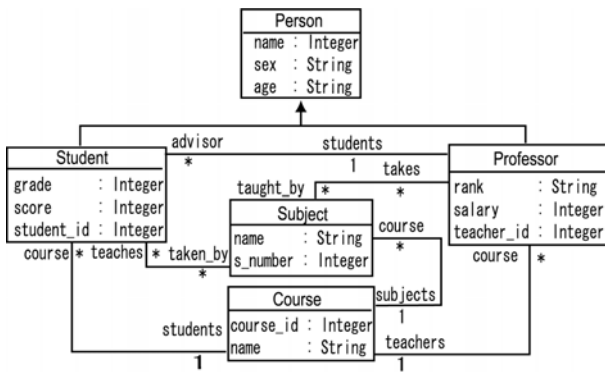


図 1：オブジェクトデータベースの例

当システムはプロパティを可視化対象とすることでデータの属性だけでなくデータ間の関連も可視化することが可能となっている。また、スキーマ情報を自動的に取得し可視化設定に使用することで、ユーザが可視化を行う際の対象データのタイプ設定のような手間を軽減している。

システム全体は、自動化機能と可視化機能の 2 つから構成される。自動化機能がユーザの入力に従い可視化の手法を自動決定し、決定した手法に従って可視化機能が可視化を実現する（図 2）。

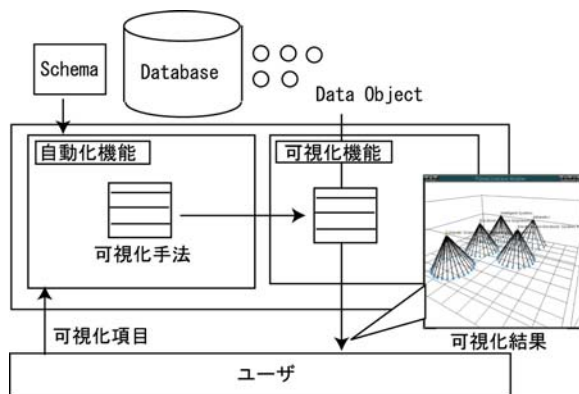


図 2：可視化システム概要

4、 自動化機能

自動化機能はユーザの入力した可視化項目を受け、そ

の内容から棒グラフ[12]や円グラフ[12]などの最も適すると思われる可視化手法を決定する。ユーザが入力する可視化項目は、可視化の対象とそれに適用する操作の組からなる。

4, 1 可視化対象の指定

可視化の対象は、「クラス Professor の age」のようにオブジェクト集合とそのプロパティが指定される。対象オブジェクト集合には、データベース内に存在するエクステントとデータベースへの問い合わせの結果として得られたオブジェクト集合が指定可能である。ユーザにはオブジェクト集合の選択肢とプロパティの選択肢がそれぞれリストボックスで提供され、それぞれ 1 つを選択することで可視化対象を特定する。

図 1 のデータベースを可視化する際、エクステントとして定義されているオブジェクト集合には Person, Student, Professor, Subject, Course があり、この一覧が選択肢として提供される。ユーザが選択肢より Professor を選択すると、さらに Professor のプロパティ (rank, salary, name, sex, age, students, takes, course) を指定するための選択肢が提供され、salary を指定することで可視化対象「Professor の salary」が決定する。問い合わせ結果の可視化を行う際には、ユーザによって発行された問い合わせの変数がオブジェクト集合の選択肢に追加される。例えば、可視化対象を「41 歳以上の Professor」としたい時、ps = Professor from Professor where Professor.age>40 のような問い合わせを発行すると、問い合わせで定義された変数 ps が Person, Student, Professor, Subject, Course からなる選択肢の新たな要素として追加される。ユーザは選択肢から ps を選択し、さらに ps の持つプロパティ、つまり Professor のプロパティを選択することで、「40 歳以上の教官を示す変数 ps の salary」のような問い合わせ結果を可視化対象として指定する。

選択された対象プロパティは、それぞれが Integer や Float, Bool などのスキーマで定義されたタイプを持っているが、このタイプによるデータの分類は、可視化を行う上で適した分類であるとはいえない。そこで、選択された対象プロパティは、その性質に応じて可視化のためのタイプに分類される。

対象プロパティはまず、属性と関連の 2 つに大別される。対象プロパティが属性である場合、プロパティは Wehrend の分類法[10]を参考に、Nominal, Ordinal, Quantitative, Time, Geographical, の 5 つに分類される。Nominal は同値比較のみが可能なデータで、

Enumerate や Integer、String 等が相当する。Ordinal は同値比較と大小比較が可能なデータ、Quantitative は同値、大小比較に様々な算術計算が可能なもので、Integer、Float などの数値全般の属性がこの 2 つに相当する。Time 型は時間を表す Ordinal 型の特別な属性であり、Geographical 属性は地理情報をあらわす Quantitative 型の特別な属性である。例えば、Professor の Integer 型の属性 salary は Quantitative として定義され、String 型属性 rank は Nominal として定義される。ユーザの選択したプロパティが Integer などの数値型の属性である場合、その可視化のためのタイプはスキーマ情報から一意に決定することができない。そこで、ユーザが対象を選択する際に、可視化分類指定用のウィンドウを提供し、必要に応じてユーザが手動で可視化分類を特定するようにしている。対象プロパティが関連である場合は、関連はその多重度にしたがって 1 対 1、1 対多、多対 1、多対多の 4 つに分類される。

4, 2 可視化操作の指定

選択した可視化対象に適用する操作は、Wehrend の分類法[10]をもとに以下の 11 種類が定義されており、対象プロパティに対して適用可能なものが選択肢として提供される。

- 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... 他の属性との関連を表現
- Order... 属性値で整列
- Compare... 属性値を比較
- Categorize... 属性値ごとに分類
- Summarize... 属性で集計
- Connect... 属性間の関連を表現
- Hierarchy... 階層関係を表現
- Associate... 相関関係を表現

対象プロパティに適用可能な操作は、図 3 の表より決定する。例えば、選択プロパティが Nominal 型の属性である場合は Identify, Distribute, Correlate, Summarize, Categorize が選択肢として提供され、1 対多の関連属性である場合は Connect と Hierarchy が選択肢として提供される。ユーザは選択肢から操作を 1 つ選択し、可視化項目を決定する。

対象と操作からなる可視化項目を受け、自動化機能が図 4 に示すような変換テーブルを自動的に作成することで可視化手法を定義する。

操作	プロパティが属性の時					プロパティが関連の時			
	Nominal	Ordinal	Quantitative	Time	Geographical	1対1	1対多	多対1	多対多
Identify	○	○	○	○	○				
Distribute	○	○	○	○	○				
Locate					○				
Correlate	○	○	○	○	○				
Order		○	○	○					
Compare			○						
Summarize	○			○					
Categorize	○			○				○	
Connect						○	○	○	○
Hierarchy						○	○		
Associate									○

図 3： 属性の種類ごとに適用可能な操作の一覧

No	Class, Var	Filter	Agr	Type	Rel	Child	Layout	X, NX, A	Y, NY, R	H, Z	Size	Ch	Cs	Cv	Label
0															
1															
...															

図 4： 変換テーブル

4、3 変換テーブル

可視化の手法は変換テーブルによって定義される。変換テーブルは一つの行が一つのオブジェクト集合の可視化手法を定義しており、各行はそれぞれ可視化詳細決定用のセルから構成されている。変換テーブルの各行では、システム側で定義された可視化のための基本的な表現要素（プリミティブ）を組み合わせることで可視化手法を定義する。プリミティブは図形の種類、水平方向の配置法、図形の描画属性値から構成される。

図形の種類には、Point, Bar, Area, Enclosure, Tree, Link の6種類がある。Point, Bar, Area はデータの属性や分布を表現し、Enclosure, Tree, Link はデータ間の関連を表現する（図5）。



図5：図形の種類

水平方向の配置法には2次元空間内に散布する Scatter、格子状に順番に配置する Matrix、円状に配置する Circle の3つが用意されている（図6）。当システムでは図形を3次元空間内に配置することで可視化を実現するが、3次元空間を利用して効果的に可視化を行うために、3つの軸を全て同等に扱うのではなく水平方向と垂直方向を区別して扱うことにしている。垂直方向は図形のサイズや階層構造、異なるオブジェクト集合を区別するために与えられ、水平方向はデータの分布状況や平面的な構造を表現する。

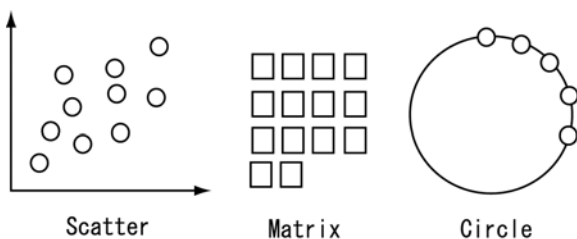


図6：図形の配置法

図形の描画属性値は、実際に描画される図形の詳細を決定し、位置、水平方向のサイズ、色（色相、明度、彩度）、ラベルの4つからなる。描画属性値により対象データの持つ値の大きさや分布状況を視覚的に表現する。

図形の種類、配置法、描画属性値からなるプリミティ

ブの組み合わせを変換テーブルの各行で指定することにより、行ごとに可視化手法を定義する。

図形の種類は変換テーブルの Type のセルに、水平方向の配置法は Layout のセルに記述することで、その行の可視化に使用する図形の種類とその配置法を指定する。描画属性値の設定は変換テーブルの右半分、各描画属性値と対象プロパティを対応づけることで行われる。例えば、図形の種類が Bar の時に対象プロパティ Salary を図形の高さを表す描画属性値 H のセルと対応付けることで、各データオブジェクトが持つ Salary の値を Bar の高さで表現する。X,NX,A と Y,NY,R のセルは図形の水平方向での位置を表す。位置を表す描画属性値は配置法によって異なり、配置法が Scatter の時は X,Y 座標、Matrix のときは NX,NY で配置順序、Circle の時は A,R で角度と半径を表す。H,Z はそれぞれ図形の高さと z 座標の位置を表す。Size は図形の水平方向のサイズ、Ch,Cs,Cv はそれぞれ図形の色相、彩度、明度を表す。Label はデータ文字列をそのまま図形の隣に表示する。

変換行を構成する残りのセルのうち、No は行を識別するための番号、Class,Var には可視化対象のオブジェクト集合名が記述される。Filter,Agr は対象にたいして行われるフィルタリングや集約について記述される。関連を可視化する際には、関連の種類が Rel のセルに記述され、また関連のある行を Child で番号指定することで関連設定を行う。全てのセルの記述内容をもとに、その行で使用する可視化手法が定義される。

図形の種類や配置法、描画属性値は、既存の可視化技法に関する文献[12][9][13][18]を体系化することで決定されている。格子図[19]は包含と格子配置、円グラフ[12]は領域と円状配置、ConeTree[20]などの Tree 表現は点とリンクのように、これらのプリミティブの組み合わせによって既存の可視化の大部分が実現できる。また、円グラフと棒グラフを組み合わせると高さのある円グラフを作成したり、ツリーと棒グラフを組み合わせたりと、複数個の可視化を試行錯誤的に組み合わせることでもさまざまな可視化を実現することが可能である。

4、4 自動的な変換テーブルの作成

変換テーブルはユーザの入力をもとに、自動化機能によって自動的に作成される。変換テーブルの作成には図7のような変換対応表を用いる。変換対応表の列と行は可視化対象データのタイプと操作からなり、各組み合わせに適用可能な描画属性値が優先度の高い順に並んで記述されている。対応表は Macknlly による研究成果[16]を基準に作成している。Macknlly は Nominal 型、

Ordinal 型、Quantitative 型のそれぞれの属性を表現するために、どのような表現方法が適しているかという研究を行っている。我々はこの研究内容を利用することで対象データのタイプと操作の組み合わせに適用可能な描画属性を設定し、それぞれに優先順位を設定した。

ユーザから可視化項目を受けると、自動化機能は対応表を用いて可視化に使用する描画属性値の候補を決定し、その中で最も優先度の高い描画属性値を可視化に使用する。図形の種類と配置法については、NX なら Matrix、H なら Bar のように描画属性値から特定できるときはその図形、配置法を使用する。Ch やラベルのように図形の種類、配置法が決定できない描画属性値が選択された場合は、図形が Point、配置法が Matrix として設定が行われる。例えば、可視化項目を「Ordinal 型の属性を Identify」と指定した場合、図 7 の表にしたがって描画属性値 NX が選択される。Matrix の配置順序と対象プロパティの値が対応付けられ、デフォルトの図形である Point 対象の値の順に格子状に配置される。Compare の場合は対象プロパティと図形の高さ H だけでなく、NX にも対応付けることで図形を値ごとにソートし、データの比較を実現する。Label 属性については、属性値をそのまま文字列として描画するが、オブジェクト数が多いと逆に認識が困難になると考えられる、そこで、Label 属性は対象のオブジェクト数に応じて使用するか否かを決定する。

変換テーブルの行は可視化項目が追加されるたびに作成される。1 つの行は 1 つのオブジェクト集合の可視

化を定義するので、複数行で変換テーブルを構成することにより複数個のオブジェクト集合の可視化を同時に行うことが出来る。ただし、既に行の存在するオブジェクト集合、つまり一度可視化が行われたオブジェクト集合について同じオブジェクト集合の異なるプロパティの可視化を追加する時には、新たな行は追加せず同じ行に別の設定を追加する。例えば一度「Nominal 型のプロパティの Identify」で可視化した結果に、同じオブジェクト集合の「Quantitative 型のプロパティを Compare」を追加すると、ラベルは表示されたまま、点で描画されていたオブジェクトがバーに変わり、配置順とバーの高さでその属性が表現され、1 つの可視化で 2 つのプロパティの情報を表現することが可能になる。

同じオブジェクト集合の可視化を追加する際は、既存の可視化と新たに追加された項目で同じ描画属性値を使用すると、既存の可視化結果が上書きされてしまう。そこで、同じオブジェクト集合の可視化を追加するとき使用可能な描画属性値は、既存の可視化で使用されていない描画属性値のみを優先度に従って使用する。「Ordinal 型の属性を Identify」という可視化を行った後、同じオブジェクトの「Nominal 型の属性を Distribute」という可視化を追加すると、使用できるもっとも優先度の高い描画属性値 NX は既に以前の可視化で使用されているので、次に優先度の高い NY が可視化に使用される。

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

図 7：変換テーブル作成の対応表

5、可視化機能

自動化機能によって構成された変換テーブルに従い、可視化機能が実際に可視化を行う。

自動化機能より変換テーブルを受け取ると、その各行で対象として指定されているオブジェクト集合をデータベースより取得する。取得したオブジェクトは対象プロパティの値ごとにソートされ、最大値と最小値の範囲

にしたがって 0 から 1 に正規化される。対象データが文字列の場合は、辞書順にソートして 1 から N に正規化される。正規化した値から行ごとに設定されたサイズや色などの描画属性値を決定し、その描画属性値を持つ図形を生成する。

生成された図形を 3 次元空間に配置する際、空間のサイズやデータ間の関連を考慮した上で図形間の間隔を

一定以上に保つように座標が決定される。各行には、可視化を行う際に、関連を考慮した上で高さの範囲、色相の範囲が割り当てられる、これは、複数行で同じ色、高さを共有して混乱が生じるのを防ぐための処理で、それぞれ行ごとに与えられた高さの範囲内に、与えられた範囲の色相を持つ図形が配置される。

ユーザは可視化結果を見た上で、修正や可視化項目を行い、対話的に可視化を実現する。項目の追加は新規での可視化と同じ手順で行い、結果の修正は変換テーブルを構成するセルを書き換えることで実現する。可視化の履歴はユーザが項目を追加するたびに保存され、必要に応じて履歴中の任意の状況に戻ることが出来る。この「戻る」機能により、対話的な可視化をより快適に行うことが可能となる。

ユーザは、データベースに対してだけでなく、可視化結果に対しても問い合わせを発行することができる。可視化結果として表示された図形集合の一部分を3次元矩形により選択することで対象を指定し、平均や合計などの演算を行ったり、可視化を適用することが出来る。

6、可視化例

オブジェクトデータベースシステムに富士通の Jasmine データベースシステム[17]、プログラミング言語に C++、図形の描画およびユーザインターフェースに OpenGL を使用して Solaris 上で実装を行った。

システムを使用してサンプルデータベースの可視化を行い、対話的な可視化によって実現できる様々な可視化結果を示すことで、システムの有用性について検証する。

6, 1 サンプルデータベース

システムを用いて可視化を行うために、まずは Jasmine 内にサンプルデータベースを作成した。サンプルデータベースは大学の学部を想定したデータベースで、構成は図1で示すようなものになっている。各オブジェクト集合は、Professor が 116 個、Student が 216 個、Subject が 51 個、Course が 5 個のオブジェクトからなり、それぞれが複数個の属性と関連をもつ。

6、2 対話的な可視化

可視化の対象オブジェクト集合を Professor とし、ユーザの様々な目的を想定した上で以下の可視化目的を対話的に追加し可視化を行った。

- Professor の age を Distribute
- Professor の salary を Distribute

- Professor の rank を Identify
- Professor の students を Hierarchy

Quantitative 型の age を Distribute という項目をうけて、対応表より X が選択され、X 軸方向に点が分布した結果が表示される(図 8(a))これに salary の Distribute を追加することで、今度は Y が salary と対応付けられ X 軸で age、Y 軸で salary の分布を表した 2 次元散布図が得られる(図 8(b))。この図より、年齢と給料には比例関係があることが見て取れる、また rank を Identify することで、Ch が選択され、rank 毎に点を色分けしたり(図 8(c))、また、students との Hierarchy を見ることでこれまで結果として得られた Professor と、生徒の階層関係を得ることが出来る(図 8(d))。また、戻る機能を用いて図 8-3 の状態に戻り、そこから Professor の Course を Categorize と指定することで、別の関係を含む図で表示したり(図 9(a))、8-3 での可視化を定義する変換テーブルを直接修正することで、点で表示していた散布図を、高さで salary を表す bar の散布図に変えるなど(図 9(b))、ユーザの目的に応じて対話的に様々な可視化を実現できる。

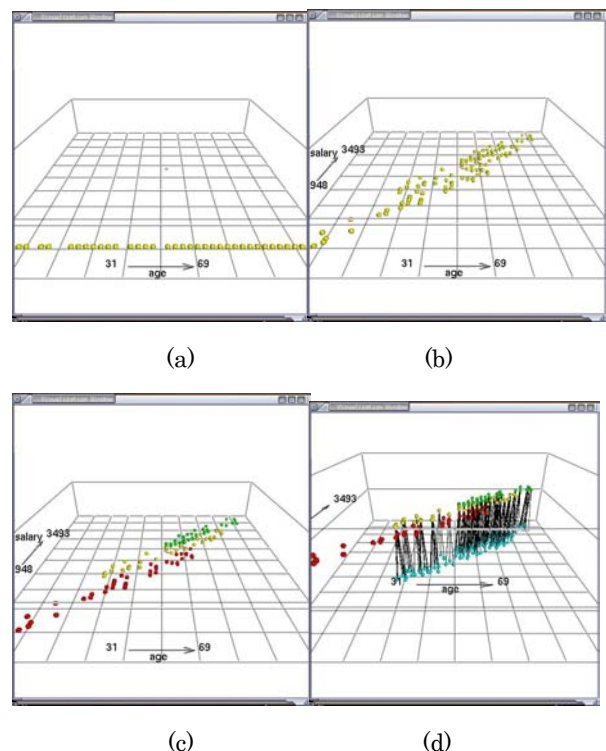


図 8 : 対話的な可視化の例

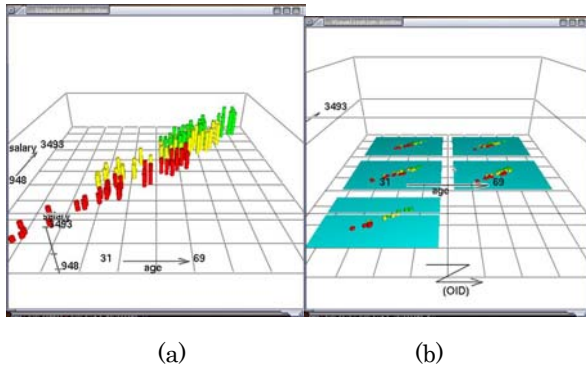


図9：実現可能な可視化の例

7、まとめ

本論文では、オブジェクトデータベースのための自動的、対話的な可視化システムを提案した。可視化対象をオブジェクトデータベースに格納されたオブジェクト集合のプロパティとし、自動的に可視化手法を決定することで可視化に関する予備知識のないユーザにも容易に可視化が行える環境を実現した。また、結果を得た上でのユーザの手動での修正や追加作業を可能とすることで、対話的、試行錯誤的に様々な可視化が実現できると考えられる。今後はシステムについて操作性、実現可能な可視化、知識発見の観点から評価を行い、更なる改良を行う予定である。

8、参考文献

- [1] 中嶋正之, 藤代一成, “コンピュータビジュアライゼーション”, November 2000.
- [2] R.Rao and S.K.Card, “The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information”, Proceedings of CHI’94, New York, 1994, pp. 318-322
- [3] C.Ahlberg, E.Wistrand, “IVEE: An Information Visualization and Exploration Environment”, Proceedings International Symposium on Information Visualization, pp.6-73, 1995.
- [4] R. A. Becker, W.S.Cleveland, M.J.Shyu, “The Visual Design and Control of Trellis Display”, Journal of Computational and Graphical Statistics, Vol.5, No.2, pp.123-155, 1996.
- [5] M.Spenke and C.Beilken, “Discovery Challenge: Visual, Interactive Data Mining with InfoZoom – the Financial Data Set”, Workshop Notes on “Discovery Challenge”, 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD ’99 1999, pp.33-38.
- [6] M. Stonebraker, J. Chen, N. Nathan, C.Paxson, A.Su, J.Wu, “Tioga: A database-oriented visualization tool”, Proceedings of the IEEE’93 Visualization Conference, pp.86-93, October 1993.
- [7] A. Aiken, J. Chen, M. Stonebraker and A. Woodruff, “Tioga-2: A Direct Manipulation Database Visualization Environment”, International Conference on Data Engineering, pp.208-17, February 1996.
- [8] Seiji Isobe, Kiyoshi Kurokawa, Hisako, Shiohara, Tetsuya Iizuka, “A Visual and Multi-dimensional Data Analysis System: INFOVISER” IPSJ Journal Vol. 40, No.05-50, 1999.
- [9] I. Fujishiro, R. Furuhashi, Y. Ichikawa and Y. Takeshima, “GADGET/IV: A Taxonomic Approach to Semi-Automatic Design of Information Visualization Applications Using Modular Visualization Environment”, Proc. IEEE Symposium on Information Visualization 2000, October 2000, pp. 77-83.
- [10] S. Wehrend and C. Lewis, “A Problem-Oriented Classification of Visualization Techniques”, Proc. IEEE Visualization ’90, pp.139-143, 1990.
- [11] R. G. G. Cattell, Douglas K. Barry, et al., “The Object Database Standard: ODMG3.0”, Morgan Kaufmann Publishers, 2000.
- [12] C.Chen, Information Visualization and Virtual Environments. Springer-Verlag, 1999.
- [13] S. K. Card, J. Mackinlay, “The Structure of the Information Visualization Design Space”, Proc. IEEE Information Visualization ’97 Symposium, pp.92-99, October 1997.
- [14] S.Chaudhuri and U. Dayal, “An Overview of Data Warehousing and OLAP Technology”, SIGMOD Record, Vol.26, No.1, March 1997.
- [15] Advanced Visual Systems, Inc. AVS Developer’s Guide, 1992
- [16] J.Mackinlay, “Automating the Design of Graphical Presentations of Relational Information”, ACM Transactions on Graphics, vol.5, no.2, pp.110-141, 1986.
- [17] Hiroshi Ishikawa, Yasuo Yamane, Yoshio Izumida, Nobuaki Kawato, “An Object Oriented Database System Jasmine: Implementation, Application, and Extension”, IEEE Transactions on Knowledge and Data Engineering, April 1996, Vol.8, No.2, pp 285-304.
- [18] L.Tweedie, “Characterizing Interactive Externalizations”, Proc. CHI ’97, pp.375-382, 1997
- [19] L.Tweedie, R. Spence, H Dawkes and H. Su, “Externalising Abstract Mathematical Models” Proc. CHI ’96, pp.406-412, 1996.
- [20] G. Robertson, J.Mackinlay and S.K.Card, “Cone Trees: Animated 3D Visualizations of Hierarchical Information”, Proc. CHI ’91, pp.189-194, 1991