

C²IMA/JS: 検索可能な可視化パーツ事例集とカスタム化支援機能を用いた Web 情報環境向けアジャイル可視化ツールキット

瓦谷 望† 市川 哲彦‡

† 山口大学大学院理工学研究科 〒755-8611 山口県宇部市常盤台 2-16-1

‡ 山口大学メディア基盤センター 〒755-8505 山口県宇部市南小串 1-1-1

E-mail: † p012vk@yamaguchi-u.ac.jp, ‡ ichikay@yamaguchi-u.ac.jp

あらまし 現在 Web browser 上で情報の可視化を行う機能が提供されているが、これを利用するにはプログラミングが必須であり、また多数の技法の中から適切なものを選択することは容易ではない。一方ブログの普及によってブログ用のアクセサリのデータベース化が進んでおりこれらブログパーツの検索、カスタマイズが容易に行えるようになってきている。

そこで我々は情報の可視化を行うための HTML 断片である可視化パーツをデータベース化し、ブログパーツデータベースと同様の機能の提供を考えた。これにより技法別の検索機能の他に、可視化目的による検索機能、簡単な対話操作で可視化パーツを自動生成する機能の提供を行う。本報告では作成したシステムについて報告を行う。

キーワード 情報可視化, データベース, ブログ

1. はじめに

現在 Web browser はコンピュータを使う上での必須のツールとなっているが、Ajax[1], Adobe Flash Player[2], Firefox add-on[3]などの普及により単なる HTML viewer ではなくプログラム実行環境としての性質を帯びるようになってきた。また、情報可視化についても、Web browser 上で可視化が行われるようになってきおり[4,5], 折れ線グラフや棒グラフ等の標準的なビジネスチャートや、tag cloud や有向/無向グラフ表示のような凝った視覚的データ表示も容易に Web browser で実現可能となっている。しかし、通常これらの機能を利用するためには何らかのプログラミングが必要であるため、プログラミングを行わない多くの利用者にとって使いにくいのも事実である。

一方、ブログの急速な普及によりブログで利用可能な様々なアクセサリが開発され、これらブログパーツ[6]のデータベース化が進んでいる。利用者は階層カテゴリを適宜たどりながら、あるいはキーワード検索を用いてブログパーツを調べることができる。通常ブログパーツには HTML テキストが関連づけられており、利用者はこれを自身のブログにそのまま埋め込むことで容易にアクセサリを利用することができる。また、利用者自身が簡単な対話操作でカスタマイズ可能なブログパーツもある。

しかしながら、多様な可視化技法の中から適切なものを選択することは容易ではない。実際、情報可視化における適切な技法選択を支援するために、GADGET/IV[7], C²IMA[8]のような半自動化システムが開発されている。これらは利用者の意図を表す可視化目的から候補となる技法を提示し、さらにその技法を実現したプログラムを示すことで利用者の可視化プロセスを支援する。

このシステムと同様に、各種ブログパーツの利用支援を行うことができれば、単にチャートの種類から試行錯誤によって技法を選択するのではなく、より効率的かつ的確に効果的な可視化技法に辿り着くことができるようになると考えられるが、このような試みは著者らの知る限りではなされていない。

そこで本研究では、情報の可視化を行うための

HTML 断片である可視化パーツをデータベース化し、ブログパーツデータベースと同様の機能の提供を行うことを目的とする。

我々は作成したシステム全般を C²IMA/JS と呼んでいる。これは本システムが先行研究である C²IMAなどを参考に作成しているため、C²IMA では対象としていなかった JavaScript での情報可視化を対象としているためこの名称をつけている。

本システム利用することで、利用者は可視化パーツを可視化技法と可視化目的という2種類の観点から検索可能で、また、可視化パーツ生成時に支援を受けることができる。可視化パーツのデータベースを作成する前段階として、Google Visualization API と JavaScript InfoVis Toolkit[9], Protovis[10] のギャラリー及び自作した事例を現時点で 128 件収集し、チャート種類別に事例のデモンストレーションやソースコードを閲覧可能にした。さらに、収集した事例をデータベース化し、GADGET/IV と同様に Wehrend 分類[11]および Shneiderman 分類[12]を取り入れた分類をおこない、可視化目的およびいくつかの検索オプションに基づいて検索可能にしている。また、可視化パーツのテンプレートを基にして、Web サービスが提供するデータを可視化するパーツを生成する機能の初期設計及び、部分的な実装も行った。本機能を利用することによって、利用者は簡単な対話操作のみで JavaScript のプログラムを記述することなく、可視化パーツを生成することができる。また可視化パーツを生成する上で各事例に共通な部分と事例ごとに異なる部分を明確化し、自動生成の簡略化を図った。

本論文の構成は次のとおりである。第2節では、可視化事例を収集した JavaScript 用可視化ライブラリの種類と、可視化事例の分類に用いた分類方法及びユーザ支援についての関連研究について報告し、第3節ではシステムの利用例を用いて概要を説明する。第4節では本研究の目的である可視化パーツデータベースと可視化パーツコレクションについての関係を言及し、第5節で可視化パーツコレクションの詳細を述べる。第6節では収集した事例について目的指向分類を適用した結果について紹介する。第7節では可視化パーツデータベースについて述べ、第8節では可視化を自動

生成するためのシステムである可視化パーツクリエータの全体像及び作成した部分について報告する。第9節で本システムを利用事例を用いて評価し、第10節では本研究のまとめと今後の課題を述べて結論とする。

2. 関連研究

JavaScript 用可視化 API ライブラリとしては Google Visualization API, JavaScript InfoVis Toolkit, Protovis などが公開されている。本研究ではこれらのライブラリのギャラリーから事例を収集している。

本研究では可視化目的等を用いて、可視化事例や技法を検索する。その際には Whehrend 分類と Shneiderman 分類を組み合わせて利用している。Whehrend 分類では、与えられたデータセットからユーザが何を抽出したいかを表す可視化目的 (goal) に基づいて、それを満たす適切な技法が決定できる。Whehrend のアプローチでは、どのような可視化を行うかを表す action (A) と、オブジェクトの性質を表す target (D) より可視化目的である goal が表現される。Shneiderman 分類も同様であるが、action に相当する task, target に相当する type が異なっている。

また、これらの分類を用いた例として、GADGET[13] および GADGET/IV がある。前者は科学技術データの可視化を対象としており、可視化プログラムの実行プラットフォームとして、モジュール型可視化環境の一つである、AVS を利用している。後者はそれぞれ情報可視化、および AVS の後継の AVS/Express である。本研究では対象が情報可視化である点は GADGET/IV と同様であるが、可視化プラットフォームとして、JavaScript の実行環境をもつ Web ブラウザを想定している点と、利用者としてデータ分析の専門家ではなくより一般的な利用者を想定している点が異なる。

3. システムの概要と利用例

本研究で作成したシステムの利用事例をユーザの視点よりみていく。図1が本システムのユースケース図[14]である。

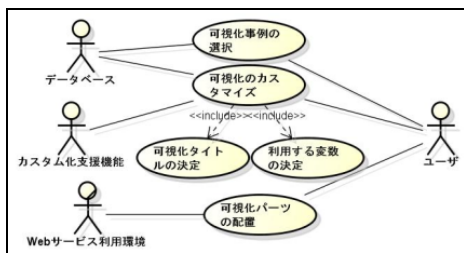


図 1 可視化パーツクリエータのユースケース図 (astah*により作成 include を使う)

はじめに、ユーザはある株式の始値(はじめね)と終値(おわりね)の移り変わりをブラウザで視覚的に閲覧したいと考えているとする。今回は指定された株式のデータを返す簡単な Web サービスを構築してデモ及び実装に用いている。このサービスは株式コードを入力すると、Yahoo!ファイナンス[15]表示されている始値、終値などの株式に関する表データを XML 形式に変換して返すものである。また、このデータにはメタデータとして変数名、変数にアクセスするためのパス式などの情報が付加されているとする。メタデータの詳細については8節で述べる。

ここで、Excel¹等での可視化結果の画像を Web ページに“張り付ける”のとは異なる点に注意されたい。画像を Web ページに張り付けた場合、可視化とデータとの結びつきが完全に断たれてしまうため、動的な可視化、すなわち、データソースの内容が変わった際に自動的に可視化結果に反映される機能や、対話的な操作を施す機能を提供することができない。一方、我々のシステムにより生成した場合であれば、動的な可視化の提供が可能となる。



図 2 可視化事例検索ユーザインターフェースのトップ画面(一部のみ)

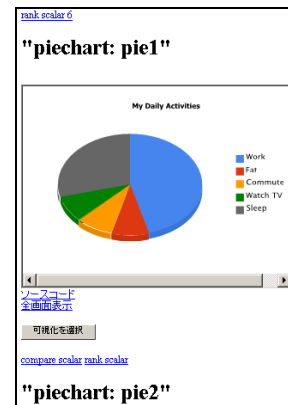


図 3 可視化目的 compare scalar を設定した際の検索結果(一部のみ)

図2が検索画面である。詳細については7節で説明を行う。通常の pie chart や bar chart のような chart 分類による階層的な閲覧も可能だが、我々のシステムでは可視化目的による検索ができるという点に特徴がある。可視化目的とは、詳細については6節で述べるが、可視化の意図を表す記号列のことである。ここでは1銘柄の株を日付単位で比較したいので compare scalar を検索キーとして利用する。図2の action とラベル付されたプルダウンメニューで compare を、target1 で scalar を指定して検索ボタンを実行すると、図3のような検索結果画面に遷移し、該当する可視化事例が一覧で表示される。この例で検索される事例に line chart があるため、ここでは line chart の事例を選択したとしよう。ユーザはこの事例をもとに可視化パーツを作成する。検索された事例はソースコードを伴っているため、ユーザは直接このサンプルコードから自身のパーツを作成することもできる。ただし、そのような事が可能なユーザは限られており、より広範なユーザを対

¹ Excel は Microsoft 社の商標である。

象とするには、何らかの利用者支援の提供が望ましい。そこで我々は可視化パーツクリエータの構築を行った。詳細は8節で述べるが、「可視化を選択」というボタンをクリックした後、簡単な対話操作のみでソースコードを編集することなく自身の可視化を作成することができる。作業の最終工程では作成した可視化パーツをWebページに張り付けるのであるが、直接張り付けた場合にはライブラリの衝突や各種ラベルの衝突などが懸念されるので、この例では可視化パーツを組み込んだWebページをサーバ上に配置し、iframeを用いて参照することとした²。結果を図4に示す。

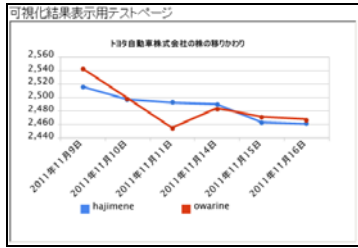


図4 クリエータが生成したHTML断片をWeb公開スペースに貼り付けた結果

4. 可視化パーツデータベースと可視化パーツコレクション

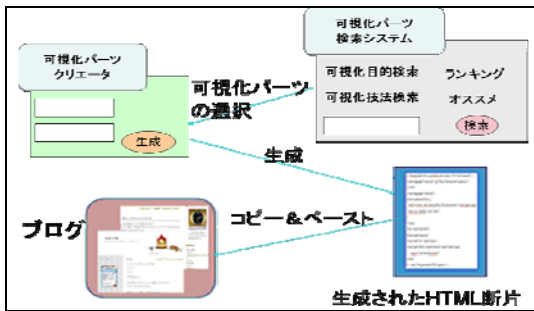


図5 可視化パーツデータベースを用いたシステムの概念図

本節では、可視化パーツデータベースを構築するための前段階として作成した可視化パーツコレクションについて説明を行う。なお、本節の内容の一部は文献[16]で報告済みである。

ここでは、HTMLで記述された可視化のためのアクセサリ全般を可視化パーツと呼ぶことにする。描画処理や対話処理についてはJavaScriptやFlashなどのクライアントサイドスクリプトを用いることを念頭に置いている。また、可視化対象となるデータはWebサービスを通じて取得することを想定している。さらに一般的な可視化構築支援を行うための汎用の枠組みの提供も考えている：情報可視化は可視化対象データに対しての視覚的構造 (visual structure) [17]の定義によって記述されるので、利用者はデータソースへのアクセス手段の指定と変数の視覚的構造へのマッピングを指定することによって可視化事態を定義することがで

² 本システムの現在の実装ではユーザはWebページの公開サービスを利用できるものとしており、このような、可視化に必要なWebページの配置も適宜行えるものと仮定している。

き、従って新たな情報可視化を対話操作によって行うことができる。実際の可視化にはGoogle Visualization APIなどの各種ライブラリの利用を想定している。

可視化パーツデータベースは単なる可視化パーツの集合体ではなく、検索等の利用者支援機能も提供する。利用者支援機能としては、単なる階層的ブラウジングやキーワード検索だけではなく、利用者の意図の表現に基づく検索機能を提供する。また、既知の事例を基にしたパーツの自動生成機能も提供する。

可視化パーツデータベースを作成するためには、まず、多様な可視化事例の収集が必要となる。本研究ではこの基データの集合を可視化パーツコレクションと呼ぶことにする。

図5に可視化パーツデータベースを利用したシステムの概念図を示す。利用者はまず、可視化パーツ検索システムを用いて利用したい可視化サンプルを選択する。その際、階層的ブラウジングによる可視化技法分類の検索機能だけでなく、どのようなデータを可視化するか、あるいはどのような目的で可視化を行いたいのか、といった可視化を用いる意図を考慮した目的指向分類別の検索機能を用いて可視化選択の支援を行う。その他にもオススメを表示する機能や、ランキング表示機能なども考えている。可視化サンプルを選択後、サンプルをそのまま利用したい場合は、サンプルのソースコードをHTMLの断片としてコピー&ペーストで利用する。

しかし、多くの場合、利用者はサンプルをそのまま用いることは少なく、手元にあるデータを可視化したものを利用すると考えられる。そこで、可視化パーツクリエータにより、サンプルを自分専用の可視化にカスタマイズする。その際に可視化するデータや、軸の名称など最低限必要な情報を入力することにより、それらを基に、ソースコードを適宜書き換え、利用者に合わせて可視化を実現するためのHTMLの断片であるソースコードを自動生成する。それをブログなどHTMLが利用できるスペースに貼り付けることにより、自身の可視化を実現できるというものである。これにより、利用者はソースコードを直接編集する必要がなく、データなど必要最低限の入力を行うのみで、自身の可視化を作成することができる。

5. 可視化パーツコレクション

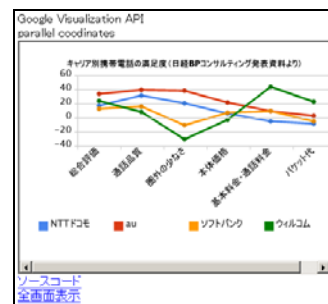


図6 可視化パーツコレクションに含まれる事例のブラウジング結果(一部のみ)

本節では4節で述べた可視化パーツの事例集である可視化パーツコレクションについて述べる。

コレクションの作成にあたっては、収集事例の件数よりも網羅性を重視する。具体的には文献[18]で紹介されている可視化技法の分類に関して、最低限1つの可視化事例を含むように収集を行う。そのため、多く

の可視化ライブラリに関して事例を収集するのではなく、Google Visualization API, JavaScript InfoVis Toolkit, Protovis の3つのライブラリに絞って詳細に調査を行い、事例を収集し、その結果網羅できなかった分類カテゴリについてはこれらのライブラリを用いて新規に事例を作成することでコレクションの補完を行った。収集結果は Google Visualization API より 64 件、JavaScript InfoVis Toolkit より 15 件、Protovis より 43 件、新規作成 6 件である。

これらの収集事例を個々の事例のソースコードを適宜編集しながら文献[17, 18]を参考に作成した分類カテゴリに従ってブラウジングできるように編成した。分類カテゴリは、bar chart, line chart, area chart, pie chart, scatter plot, link, cloud, table, map, tree, histogram, parallel coordinates, thumbnail, timetable, distortion, datasheet, other (layout, design, other の3種類)の計 19 種類である。ブラウジングの初期画面はこれらの技法のリストになっており、各技法をクリックすると該当する可視化結果の一覧にたどり着くようにしている。

ディレクトリを辿った後に表示される内容は、図 6 のように、用いている可視化ライブラリ、可視化を行う iframe、可視化のソースコードへのリンク、全画面で表示するためのリンクを含んでいる。ソースコードに容易にアクセスできるので、利用者自身がカスタマイズすることも可能である。

6. 目的指向分類

ユーザは利用する可視化技法を特定することができるのであれば、可視化パーツコレクションで可視化技法を選択することによって、可視化事例に辿り着くことができる。しかし、実際に利用者が最初に想起するのは可視化結果からどのような知見を得たいのかであって、そのための手段である可視化技法ではない。そこで、本システムでは、技法による検索ではなく、ユーザが可視化を行う目的により検索を行い、より適切な可視化を選択するための支援をするために、収集事例に対して目的指向分類を行った。

目的指向分類とはどのようなデータに対してどのような知見を得たいのかを表す可視化目的に従って可視化技法を分類する考え方である。第2節で紹介した GADGET/IV では目的指向分類を使って可視化技法を分類しており、情報可視化での成果をあげている。そこで我々の研究においても今回収集した技法に対し目的指向分類による技法の分類を適用した。それに伴い適切な分類ができなかった技法も存在し、それらに対して可視化目的以外の特徴を用いた検索方法を用いることとした。

6.1 節では目的指向分類を行うために用いた可視化目的の種類について述べる。6.2 節では可視化技法別、6.3 節では可視化事例別に分類を行った結果について報告し、6.4 節では可視化目的だけでは分類しきれなかったものに関して補うために作成したオプションの内容に関して説明する。なお6 節、7 節の内容の一部は文献[19]で報告済みである。

6.1. 可視化目的の種類

可視化目的は、どのような目的で可視化を行うかということを表す action とオブジェクトの性質を表す target を組み合わせて用いる。これは Wherend 分類と Shneiderman 分類参考に行ったものである。今回用いた action の項目は、identify, locate, distinguish, categorize, cluster, rank, compare, associate, correlate で

あり、target の項目は scalar, nominal, direction, shape, position, SERO, structure である。なお、可視化目的の詳細については付録 A を参照されたい。

これらの可視化目的を用いて、収集した事例に対して、目的指向分類を適用した。その際に同じ可視化技法を用いている場合、技法共通の可視化目的があると考え、可視化技法別に分類を行った。その後、各事例に関して、技法別の分類だけでは網羅できなかった可視化目的が存在すると考え、事例別に分類を行った。

6.2. 可視化技法に基づく目的指向分類

目的指向分類を行う際に、可視化技法が同じ場合、共通の可視化目的が存在すると考え、まず5 節で述べた全ての可視化技法に関して目的指向分類を適用した。ここでは pie chart について分類した結果を報告する。

Pie chart は 1 独立変数、2 従属変数を可視化するための技法である。その可視化対象となるデータの種類の scalar が適合する。これらからどのような知見を得たいかという action に相当するものは、データ間の比較を行うことが目的と考えられるため、compare が相当すると考えられる。また可視化結果からデータの上下関係も見て取れるため、rank にも当てはまると考えられる。以上から、pie chart には compare scalar と rank scalar の2つが可視化目的として当てはまる。

6.3. 可視化事例に基づく目的指向分類

可視化事例の中には技法別の分類だけでなく、事例独自の可視化目的に適合するものも存在する。その例として、Focus + Context という可視化事例を図7に示す。これは可視化技法で分類すると area chart に分類される。Area chart に含まれる可視化目的は、compare scalar, rank scalar, correlate scalar, identify shape, compare shape, rank shape, correlate shape, associate shape and scalar が適合すると考えられるが、ピンク色の帯を用いることで下のグラフのどの領域が上のグラフで拡大表示されているかが見て取れるので、correlate SERO もまた適合すると考えられる。このように、技法のみの分類ではなく、事例独特の可視化目的に関して分類対象としている。

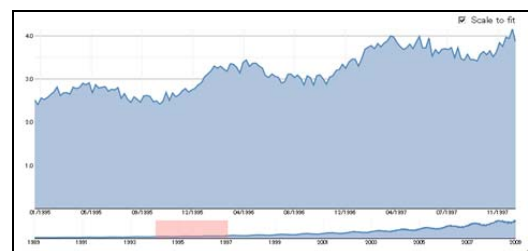


図 7 Focus + Context の可視化事例

6.4. 可視化目的以外の属性

可視化目的以外の検索オプションとして、4つの属性を追加した。その4つは、map, entertainment, design, 対話処理である。

Map は可視化に地図を利用したものを選択するために用いる。データ型として SERO を指定するとすれば地図を用いた事例も検索されるが、わかりやすさと、地図を用いた事例がよく見られるので、オプションとして追加している。2つ目が entertainment である。可

視化事例の中には、金額を表す棒グラフに紙幣の画像を用いたものや、通勤者の人数を表す棒グラフで電車の画像を用いているものがある。いずれも、通常の棒グラフの用いても表現できるものであるが、より、entertainment性の高い可視化となっている。

Designは何かを可視化したというよりはWebページの装飾用に用いられることを意識したグラフィックスを検索対象に指定するためのものである。4つ目が対話的技法を用いているものである。対話的技法とは、可視化結果より知見を得るためにzoomやfilterといった処理を行うために用いられる技法である。

7. 可視化パーツデータベース

6節で述べた目的指向分類による可視化事例検索機能の提供を行うために、可視化パーツデータベースを構築した。このデータベースは可視化目的や各種オプションによって事例を検索したり、技法単位で閲覧する際に利用される。データベースの概念設計の結果得られたクラス図を図8に示す。

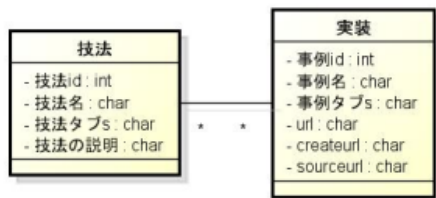


図8 可視化パーツデータベースのクラス図(astahを用いて作成)

ここでは、技法と事例を基本的な実体として捉え、その間に多対多の関連があるものとしている。可視化目的や各種オプションは技法や事例につけられたタグとしてみなすこととした。その場合タグの集合を技法や事例の属性とするのが自然であるが、ここでは多値属性として捉える代わりに、タグ集合を区切り文字で連結した単値属性として記録することとした。事例実体の属性には他に事例id、事例名、事例説明、事例を参照するためのurl、事例のソースコードを参照するためのsourceurlが存在する。Create urlには7節で述べる可視化パーツクリエータによって可視化プログラムを自動生成するためのプログラム呼び出し用のurlが存在する。技法実体には同様に技法id、技法名、技法説明、可視化目的で検索するための技法タブsといった属性が存在する。

検索ユーザーインターフェースは図2のようになっており、目的指向分類による検索とオプションによる絞り込み検索が可能である。オプションには5.4節で述べた4つの項目が存在する。目的指向分類を用いずにオプションのみの検索も可能で、その場合、目的指向分類の検索欄には何も指定しない。また、技法ごとにかategorizeもされており、これらの階層カテゴリを適宜たどることで技法を特定することも可能である。

実際に検索した結果が図3である。この検索結果はactionの項目にcompareをtargetの項目にはdirectionを指定したものである。検索した事例ごとに適合する可視化目的がタグとして事例の下に表示されており、このタグをクリックすることで、その可視化目的での検索が可能となっており、検索結果の事例に関する事例を順次検索することができる。またファセット型ナビゲーションにも対応している。ファセット型ナビ

ゲーションとは検索結果に対して、異なった切り口で検索を行う方法で、ここでは検索結果に含まれる可視化目的の一覧とその数を表示し、他の可視化目的により絞り込みを行うという方法で用いている。図3の上部にあるリンクはファセット型ナビゲーションを利用した絞り込みを行うためのものである。

8. 可視化パーツクリエータ

可視化パーツを自分用にカスタマイズするためには、プログラムを適宜変更する必要があるが、プログラミングを行わない多くのユーザーにとっては使い辛いという現状がある。そこで、プログラミングを行うことなく、簡単な対話操作のみで可視化パーツのカスタマイズを実現するために、可視化パーツクリエータの作成を行った。

可視化パーツクリエータに必要な機能として、可視化パーツで利用するWebサービスの指定とWebサービスに渡すリクエストを入力する検索欄の有無や、データのパラメータを可視化技法のどの視覚特性へマッピングするか指定をしなければならない。それらを指定するユーザーインターフェースが可視化パーツクリエータである。本節ではクリエータが持つべき機能および、現在までに作成した機能について述べる。8.1節で可視化パーツクリエータが持つべき機能の詳細を述べ、8.2節では現在までに作成しているクリエータの動作について報告する。8.3節ではクリエータを生成するために行った収集事例のテンプレート化について報告を行う。

8.1. 可視化パーツクリエータの概要

4節で述べたように、可視化サンプルを選択後は多くの場合、手元にある、あるいはWebサービスが提供するデータを可視化したものを利用すると考えられる。そこで、可視化パーツクリエータにより、サンプルに付随するテンプレートからカスタマイズされた可視化プログラムを生成する。生成されるパーツの概念図を図9に示す。

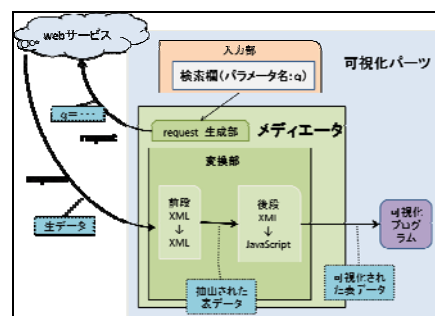


図9 可視化パーツクリエータのアーキテクチャ図

メディエータ部はWebサービスへの要求送付機能とWebサービスが返した結果を変換して可視化部に渡す機能を有している。この変換部は前段と後段とに分かれており、前段ではXMLからXMLへのデータ変換によって可視化対象のテーブルを構成し、後段ではXMLからJavaScriptへのデータ変換によって可視化部の処理対象を生成する。

8.2. 可視化パーツクリエータの動作

```

<?xml version="1.0" encoding="EUC-JP" ?>
<root>
  <meta>
    <variables>
      <variable name="hiduke" type="string" independent="true"
        path="/root/table/data/hiduke"/>
      <variable name="takane" type="number" independent="false"
        path="/root/table/data/takane"/>
      <variable name="yasune" type="number" independent="false"
        path="/root/table/data/yasune"/>
    </variables>
  </meta>

```

図 10 可視化パーツクリエータ前段部分処理終了後の XML データのメタ記述

Info-ファイル作成完了, ファイル名 [file0.php]
利用する従属変数を選択してください(1つ以上)

変数名	変数型	独立、従属	利用する従属変数
hiduke	string	独立変数	<input type="checkbox"/>
hajimene	number	従属変数	<input checked="" type="checkbox"/>
takane	number	従属変数	<input type="checkbox"/>
yasune	number	従属変数	<input type="checkbox"/>
owanine	number	従属変数	<input checked="" type="checkbox"/>
dekidaka	number	従属変数	<input type="checkbox"/>
tyouseiowanine	number	従属変数	<input type="checkbox"/>

可視化タイトルを設定してください
[会社の株の移かわり]

図 11 クリエータによる可視化結果のカスタマイズ項目決定画面

本システムでは 7 節で述べた検索機能を用いて事例を検索した後、可視化選択ボタンを押すことで、自分専用の可視化を自動生成するためのユーザ画面に遷移する。今回報告を行うものは、クリエータ後段部分以降の処理を行うものである。

現在クリエータで利用できる可視化事例は Google Visualization API に含まれる事例のみであるが、Protovis, JavaScript InfoVis Toolkit の事例に関しても同様の手法で作成することが可能と考えられる。また今回可視化対象とするデータは可視化パーツクリエータの前段部分の処理が行われていると仮定して作成している。前段部分の処理が行われた XML のデータの例を図 10 に示す。前段部分の処理の結果、XML のメタ記述として、変数名、変数の型、独立変数が従属変数か、変数にアクセスするためのパス式が記述されているとする。

可視化パーツクリエータのユースケース図を図 1 に示す。まず、ユーザは 6 節で述べた検索機能を用いて、事例の検索を行う。その際にクリエータを利用可能なものに関しては、可視化選択ボタンが表示されており、ボタンを押すと可視化対象となるデータ選択画面に移る。ここでクリエータ前段部分の処理が施されているデータを渡すと、カスタマイズ画面に遷移する。そのときの画面を図 11 に示す。

現在サポートしているカスタマイズ機能には可視化対象とする従属変数の選択機能と、タイトルなどの視覚的要素の指定機能がある。前者では、メディアエータの前段処理で取り出された変数のうち、どれをどのような視覚的特性にマップするかが指定され、これにより、グラフなどの描画内容が決定される。後者では、可視化タイトル、高さ、幅、凡例の位置、など様々な要素の指定が考えられるが、現在はタイトルのみをサポートしている。カスタマイズの内容を指定した後、送信ボタンをクリックすると、可視化結果が表示されるので、そのままよい場合は作成完了ボタンをクリ

ックすると iframe を用いた 2 行程度の HTML 断片が表示される。それを Web 公開スペースにコピー＆ペーストで貼り付けることにより、Web ページ上に自身の可視化を配置することができる。なお、第 3 節で述べたとおり、Web 公開スペースはユーザが必要に応じて利用可能と仮定している。

実際に Web 公開スペースに貼り付けたものが図 4 である。これにより、ソースコードを直接書き換えることなく自分専用の可視化を生成することができる。

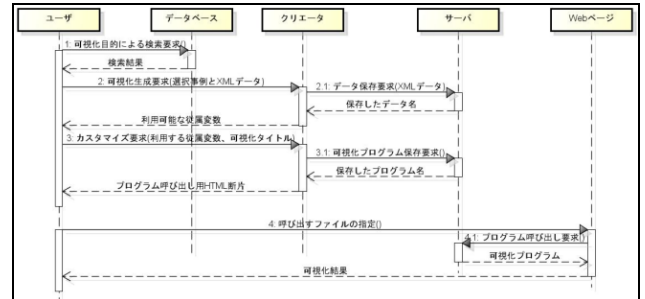


図 12 可視化パーツクリエータのシーケンス図 (astah*○○を用いて作成)

このときのシステムの動作のシーケンス図を図 12 に示す。ユーザはまずデータベースに検索要求を出す。その結果から利用したい可視化を選択し、可視化対象となる XML データとともに可視化生成要求をクリエータに対して出す。このときクリエータは XML データをサーバに保存すると同時に、従属変数の一覧を提示し、カスタマイズ内容の指定を要求する。ユーザは提示された一覧の中から可視化結果に含めたい従属変数と、可視化タイトルを決定し、再度可視化パーツクリエータに可視化生成要求を出す。クリエータはカスタマイズの内容を反映させた可視化生成用のプログラムを作成し、サーバに保存する。同時にユーザに可視化生成プログラムを呼び出すための HTML 断片を提示する。ここまですべて本システムの動作であり、その後、ユーザはその HTML 断片を Web 公開スペースに貼り付けると、Web ページがサーバに対して保存されているプログラムを呼び出し、可視化結果を Web 上に配置することができる。

本システムを利用することの利点として、データソースと可視化結果との間の動的な関係が保持されていることが挙げられる。そのため、可視化結果の画像を貼り付ける場合と異なり、データソースが更新された際には、適宜サーバ上に保存されているデータの再生成を行うことで、可視化結果の再生成を行うことが可能である。なお、メディアエータの前段部分の自動生成や、自動実行については現在未実装である。

8.3. 収集事例のテンプレート化

可視化パーツクリエータは各事例に対して個別にソースコードを準備する必要があるため、そのための労力の軽減を行うことが本システムの有用性を確保するためには必要である。そこで、可視化事例のソースコードを分析し、再利用可能な部分とパラメータに分離し、前者をテンプレートとしてまとめた。テンプレート化を行った対象の事例は Google Visualization API の事例のみであるが、他の 2 つのライブラリに関しても同様に行うことができると考えられる。

Google Visualization API を用いる場合、大半の部分

は全ての事例で共通しているが、事例ごとに異なる部分が3つある。それは、利用するパッケージの決定部分、データ記述部分、draw 関数呼び出し部分である。

利用するパッケージの部分は可視化を行う際に、予め用意されている各技法生成用のパッケージを利用するため、パッケージ名を記述する必要がある。この部分は変数としてパッケージ名を渡すことで、自動で生成するようにした。中にはパッケージを利用しない可視化が存在するが、その場合パッケージ名として空文字列を入れると、エラーになってしまうため、関係のないパッケージを入れることでエラーを防いでいる。

次にデータ記述部分であるが、この部分は可視化対象とするデータを記述する部分であるので、データが異なれば変化するものは当然である。また Google Visualization API を用いる場合、データ型も明記しなければならない。今回想定している前段部の処理内容のメタ記述部分に変数の型を利用しているのはこのためである。この部分は参照するデータのメタ記述を利用して生成する。

3つめは draw 関数部分である。Google Visualization API では可視化する際に draw 関数を呼び出して可視化を行う。その部分ではデータの他に可視化結果の表示の大きさ、可視化タイトル、2D か 3D かなど多くのオプションを指定することができる項目が非常に多い。本来であれば、これら全ての項目に関してカスタマイズできることが望ましいが、本システムでは収集した事例で設定されているオプションをそのまま利用し、可視化タイトルのみカスタマイズ可能にしている。

これにより、事例ごとのクリエイタを生成する際に変更する部分は変数受け渡し部分だけになったため、非常に効率よくクリエイタを生成することが可能になった。またクリエイタによって生成されるカスタマイズ後のプログラムはもともと 100 行程度であったが、現在は 10 行に満たない程度であり、生成されるプログラムの簡略化にも成功している。

9. 評価

本システムを用いる利点として可視化目的による検索ができることと、可視化結果とデータソースとの動的な関係が保持されていることが挙げられる。そこで、本システムを用いて作成した可視化事例と、Excel を用いて作成した可視化事例をそれぞれ Web 上に配置することを例として本システムの有用性を示す。今回可視化対象とするデータは 11 月 20 日に Yahoo!ファイナンスから得たトヨタ自動車株式会社の株の始値と終値のデータである。なお実験用に株のデータを XML データで返す Web サービスを作成して用いている。

まず、データを可視化するための可視化技法を選択する場合についてみていく。今回は、株のデータを日付ごとに比較するために可視化を行うものとする。適切な可視化技法を選択することは、可視化結果から知見を得る上で必須のことであるが、Excel の場合、可視化技法を選択するための支援は行われていない。そのため可視化目的を決めた上で、それを実現するための可視化技法を全ての技法の一覧から決定する必要がある。しかし、本システムでは可視化目的による検索を行うことで適切な可視化技法に到達することができる。今回であれば、compare scalar という可視化目的を選択することで bar chart, line chart, pie chart などが検索結果として得られる。この検索結果に含まれる事例であればどの事例を用いてもスカラー値の比較を行うことはできるので、可視化結果から知見を得るために

適切な可視化技法の選択を誤ることがない。

続いて、実際に可視化した結果から本システムを評価する。今回は両者とも折れ線グラフで可視化することを前提としている。本システムを用いて可視化を行った例は図 4 であり、Excel を用いて可視化した結果は図 13 である。この段階ではどちらの可視化結果も大きな違いはない。しかし、株価データは一日単位で変動があるため、その都度、可視化結果に反映されることが望ましい。しかし、Excel による可視化結果を用いる場合、データを入力し直した後、可視化結果を画像として再度 Web 上に配置するため、データソースに変更があるたびに同じ操作を最初から行う必要がある。しかし、本システムを用いている場合、データと可視化プログラムの関係が保持されているため、2 度目以降はサーバ上に保存されているデータの再生成を行うだけで、可視化結果の再生成が可能である。このため、日ごと、あるいは週ごとなど時間経過によって可視化対象となるデータが変化する場合、本システムを用いると作業の効率化が図れる。実際に可視化データのみ変更を行った図が図 14 である。なおデータの再生成部分の自動実行の仕組みなどは現在未実装であるため、ここで利用したデータの生成はプログラムの手動実行によっている。

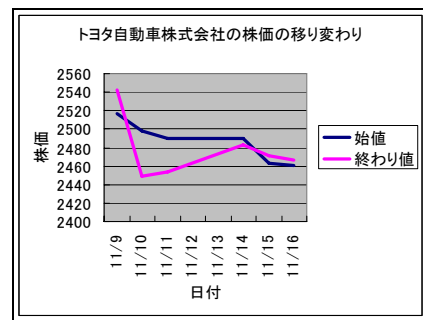


図 13 トヨタ自動車株式会社の始値と終わり値の可視化結果(Excel)

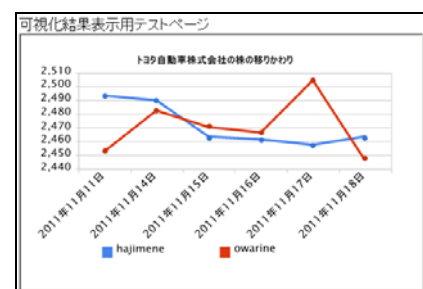


図 14 トヨタ自動車株式会社の始値と終わり値の可視化結果(可視化パーツクリエイタ)

10. おわりに

本研究では情報の可視化を行うためのアクセサリである可視化パーツを3つの可視化ライブラリより収集することで、可視化パーツの事例集である可視化パーツコレクションを作成した。また収集した事例に関して可視化技法による分類を行い、技法別にブラウジングを行った。また収集した事例に対して目的指向分類を行い、これを基に可視化パーツデータベースを構築したことで可視化目的による検索機能を実現した。

また収集した事例のうち、Google Visualization APIに含まれる事例に関して、簡単な対話操作のみでプログラムを編集することなく可視化パーツをカスタマイズする機能である可視化パーツクリエータを作成した。その後、Excelの可視化事例と比較することで、本システムの有用性を示した。

今後の課題として、Google Visualization API以外の事例をカスタマイズするための可視化パーツクリエータの作成、可視化パーツクリエータの request 生成部、メディアータ前段部分の作成が上げられる。

参 考 文 献

[1] J. J. Garrett, "Ajax: A New Approach to Web Applications," <http://www.adaptivepath.com/ideas/essays/archives/000385.php>, 2005.

[2] Adobe Inc., "Adobe Flash Player," <http://www.adobe.com/jp/products/flashplayer/>.

[3] The Mozilla Foundation, "Mozilla.org -- Home of Mozilla Project," <http://www.mozilla.org/>.

[4] Google, "Google Visualization API," <http://code.google.com/intl/ja/apis/visualization/>.

[5] W. Orcutt, "Lily," <http://www.lilyapp.org/>.

[6] blogparts.com, <http://www.blog-parts.com/about.html>.

[7] I. Fujishiro et.al., "GADGET/IV: A Taxonomic Approach to Semi-Automatic Design of Information Visualization Applications Using Modular Visualization Environment," in Proc. IEEE InfoVis2000, Oct. 9-10, Salt Lake City, Utah, 2000, pp. 77--83.

[8] 藤代 一成, 中村 恭子, 市川 哲彦, "データフロー型可視化システムにおけるモジュールネットワークワーキングの半自動化," 画像電子学会 Visual Computing '95, Jul. 1995, pp. 86-87.

[9] N. G. Belmonte, "JavaScript InfoVis Toolkit," <http://thejit.org/home>.

[10] M. Bostock and J. Heer, "Protovis: A Graphical Toolkit for Visualization,"

[11] S. Wehrend and C. Lewis, "A problem-oriented classification of visualization techniques," in Proc. IEEE Visualization '90, San Francisco, Oct., 1990, pp.139-143.

[12] B. Shneiderman, Designing the User Interface Strategies for Effective Human-Computer Interaction, 3rd ed., Addison-Wesley, 1998, Ch. 15.

[13] I. Fujishiro, Y. Takeshima, Y. Ichikawa, and K. Nakamura, "GADGET: Goal-Oriented Application Design Guidance for Modular Visualization Environment," in Proc. IEEE Visualization '97, Oct. 1997, pp. 245-252.

[14] J. Rumbaugh, I. Jacobson, G. Booch, *Unified Modeling Language Reference Manual, The second Edition*, Addison-Wesley, 2004.

[15] "Yahoo! ファイナンス" <http://finance.yahoo.co.jp/>

[16] 瓦谷 望, "可視化パーツコレクションの作成: 可視化パーツデータベース作成のための可視化事例収集," The 12th IEEE Hiroshima Student Symposium, 2010年11月, D-29.

[17] L. Wilkinson, *The Grammar of Graphics*, Springer-Verlag, 1999.

[18] 古畑 理香, "情報可視化文献リスト," <http://pfp7.cc.yamaguchi-u.ac.jp/~ichikawa/iv/>.

[19] 古江 由祐輔, *ブログ用可視化パーツデータベースの構築: Web ブラウザベースの手軽な情報可視化に向けて*, 山口大学大学院理工学研究科修士論文. 2011年2月.

付 録

A. 可視化目的記述用語彙の説明

可視化目的を記述するために、action と target を用いた。Actionは何のために可視化を行うのかという可視化をすることの目的を表し、target は可視化対象とするデータの種類を表す。これら2つを組み合わせると可視化目的を表している。その一覧を表 A.1 と表 A.2 に示す。

表 A.1 可視化目的の action リスト

action	説明
Identify	視覚的特性により、ものが何か分かる(特性を確立したもの)
Locate	特定の場所を決定する
Distinguish	異なるもの、あるいは顕著に異なるものを見分けること
Categorize	その属性ごとに分類する(予めグループを作成した後、どのグループに属するか決定)
Cluster	属性の特徴ごとに分類
Rank	階級 階層(似たタイプの他のオブジェクトに順位やポジションを与える)
Compare	属性値を比較(似ている部分や異なる部分に気付くために検査を行うこと。rankに似ているが、可視化した後に比較対象を決める点異なる)
Associate	相関関係を表現(性質が全く異なるものの関係を描くときに使われる)
Correlate	直接のつながりを確立すること。同じ種類の(性質が似ている)ものを比較

表 A.2 可視化目的を表す target リスト

target	説明
Scalar,	適切な単位の一つの値により完全に特徴づけられるデータ
Nominal	同値比較のみが可能な型(特徴、特性などで、他と区別されるプロパティ)
Direction	左右などの方角を表すもの
Shape	形状(アウトラインや表面の形状によって区別されたオブジェクト)
Position	場所(点)位置(オブジェクトの位置を表す)
SERO	空間上に面積や体積(なんらかの幾何学的な条件を満たしている平面あるいは空間上の連続的な一部分)
Structure	構造(1つのオブジェクトから考えられた他のデータからなる配置のこと。順序付けされたもの、周期になっているもの、階層的なもの)