可視化のための非構造化データの表化手法

善明　晃由† 　津田　 均† 　田中　克季†

† 株式会社サイバーエージェント

E-mail: †{zenmyo_teruyoshi,tsuda_hitoshi,tanaka_katsuki}@cyberagent.co.jp

あらまし　データ分析の際には多様な観点での可視化が重要であるが，データの収集・生成の段階で，あらゆる状況に対応可能なデータを整備することは難しい．また，分析対象のデータに変更が多い場合，構造化データを前提とする手法では保守の手間が大きくなる．本稿では非構造化データを，与えられた軸情報をもとに表形式に変換する手法を提案する．様々なデータソースのデータを表形式で統一的にあつかうことが可能となるため，汎用性の高い可視化ツールを実現することが可能となる．また，軸情報の設定をかえることで目的の可視化に適した表を定義できるため，可視化ロジックを単純化し，可視化コンポーネントの再利用性を向上できる．

キーワード　非構造化データ，可視化

1. はじめに

データ可視化では表形式のデータをあつかうことが多い．例えば，JavaScript の可視化ライブラリである d3.js [1] では TSV などの表形式データを折れ線グラフなどの SVG に変換し描画する．また，様々なデータの可視化・分析手法がデータが表形式で用意されていることを前提に提案されている [7] [5].

データ分析の際は様々な観点で可視化できることが望ましい．しかしながら，そのような柔軟な分析に対応可能なデータを表形式で事前に整備することは難しい．例として，ある Web サービスの時間別のユニークユーザ数の可視化を考える．この場合の表形式としては例えば図 1 に示す二つがある．表 (1) では各行に各時間のユニークユーザ数が設定されており，表 (2) では時間ごとに列がもうけられ各行に日毎のデータがまとめられている．時間ごとのユーザ数の変化をみるには表 (1) が適しているが，特定の時間帯のユーザ数の変化をみるには表 (2) が適している．

(1)

time	uu
2017-03-06 22:00:00	100
2017-03-06 23:00:00	200
2017-03-07 00:00:00	300
:	:

(2)

date	00時	01時	...
2017-03-06	150	250	...
2017-03-07	300	400	...
:	:	:	

図 1　表形式の例

可視化対象のデータが目的に適した表形式でない場合は，可視化の際にデータを変換するなどの追加の処理が必要となる．例えば，図 1 の (2) の形式でのみデータが用意されている場合に，時系列のユニークユーザ数変化をみるためのグラフをつくるためには，行方向と列方向の両方を走査しながら描画するロジックが必要となる．可視化ロジックと特定の用途に特化したものにしてしまうと，例えば，特定の列のみ描画するといった他の要件に対応できなくなり再利用性が低下する．

表形式のデータを扱うには関係データベースをもちいることが多い．しかしながら，分析軸が追加されるなど分析対象のデータに変更がある場合，構造化データをあつかう関係データベースではスキーマの変更等の作業が発生するため，保守に手間がかかる．例えば，図 1 において新規会員数など別の指標の管理が必要になったとする．この場合，表や列の追加が必要となり，それに合わせて可視化ロジックの修正が必要となる場合もある．

本稿では非構造化データを表形式に変換するための汎用的な手法を提案する．本手法では非構造データを 2 段階のマップとして抽象化し，与えられた軸情報をもとに表形式に変換する．種々のデータソースを 2 段階のマップとしてモデル化することで，多様なデータを表形式で統一的に扱うことが可能となる．目的の可視化に適した形式の表を，元データを修正することなく軸を設定するのみで構成できるため，可視化処理を簡素化し再利用可能なコンポーネントとして実装できる．また，入力データが構造化されている必要がないため，分析対象のデータに変化がある場合にも柔軟に対応できる．さらに，関係モデルに基づいた複数の表を統合が可能となり，データ統合も容易になる．

我々は提案手法に基づくレポーティングツールを開発し，社内での導入をすすめている．本稿では開発したレポーティングツールについても説明する．実装したレポーティングツールでは，SQL ライクな DSL により表形式のデータに算術演算や複数の表の結合など定義でき，可視化ロジックを単純化しつつ様々なレポート作成を可能にしている．

2. 提案手法

図 2 に提案手法の概要を示す．**メトリクスソース**とは可視化対象のデータをもつデータソースを抽象化したものである．**表定義 DB** では表ごとにキー列を構成する軸と，非キー列を構成する**ラベル**とを管理する．また，可視化対象の各値を**メトリクス値**と呼ぶことにする．提案手法は，表定義にもとづいてメト

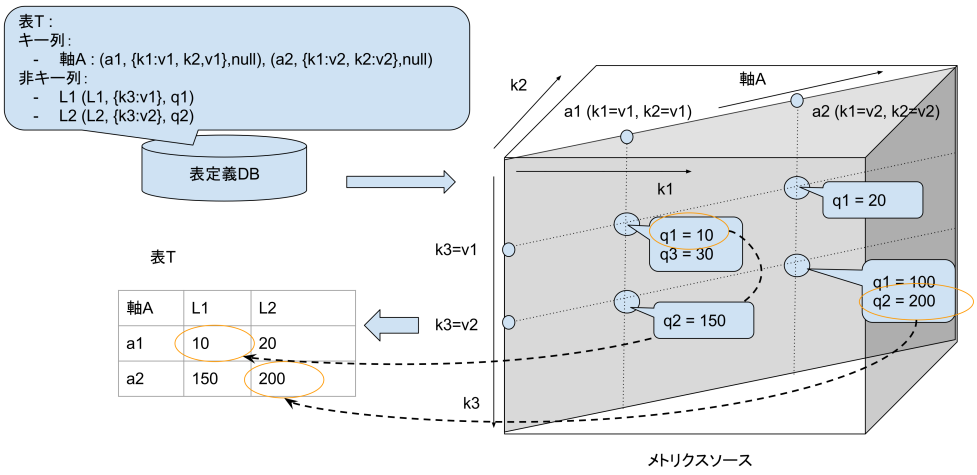

図 2　提案手法の概要

リクスソースの中で必要なメトリクス値を決定し，それを整列した表を生成する．

2.1　データモデル

2.1.1　メトリクスソース

メトリクスソースは各要素が複数の値をもつ多次元配列を考え，2 段階のマップ構造としてモデル化する．1 段目のマップは多次元配列中の要素を決定するためのものであり，**キーパラメータ**と呼ぶ．2 段目のマップは各要素の中で，メトリクス値を管理するものであり，そのキーを**限定子**と呼ぶ．

図 2 は $k1$，$k2$，$k3$ の 3 軸からなる 3 次元のメトリクスソースの例であり，各要素は $q1$，$q2$ などの限定子で指定されるメトリクス値を持つ．提案手法では，キーパラメータを指定しないことにより，より低次元のメトリクスソースをあつかうことを許容する．たとえば，$k3$ の値が与えられない場合は，$k1$-$k2$ 平面上の要素のみを対象とする．

2.1.2　表

本手法であつかう表は**キー列**と**非キー列**で構成される．キー列には，その列の値域を定義した**軸**を設定し，非キー列にはその列に表示すべきメトリクス値を定義した**ラベル**を設定する．

ラベルは，以下の 3 つ組で構成される．

$$L = (Id, KP, Q)$$

Id はラベルを識別するための文字列，KP，Q はラベルに対応するメトリクス値のキーパラメータの一部と限定子である．Q は未定義 $(null)$ の場合もある．図 2 ではラベルとして以下の二つが設定されている．

- $L1 : (L1, \{k3 : v1\}, q1)$
- $L2 : (L2, \{k3 : v2\}, q2)$

これらは $k3$ 軸上の 2 点（$k3 = v1, k3 = v2$）に対応する．

軸はラベルの有限集合，または無限集合となる．図 2 では，軸の例として，以下の軸 A を設定している．

$$\text{軸 } A = (a1, \{k1 : v1, k2 : v1\}, null), (a1, \{k1 : v2, k2 : v2\}, null)$$

この軸は $k1 - k2$ 平面上の 2 点 $(a1, a2)$ に対応する．

無限集合となる軸の例としては日付軸がある．日付軸は以下のように定義できる．

$$DateAxis = \{(date, \{'dt' : date\}, null) \mid date \sim' \backslash d\{4\}\text{-}\backslash d\{2\}\text{-}\backslash d\{2\}'\}$$

2.2　非構造化データの表化

ここでは，表の非キー列に入るメトリクス値を抽出する方法について説明する．

非キー列のメトリクス値は，対象行のキー列のラベルと非キー列のラベルに含まれるキーパラメータと限定子をマージすることで決定する．なお，現在は限定子は非キー列のラベルにのみ設定されており，キー列と非キー列のラベルで限定子の競合は発生しないと仮定している．

図 2 の例では，軸によって規定される平面とラベルによって規定される平面が交差している部分の要素から，ラベルに指定された限定子のメトリクス値を取得することとみなせる．

メトリクスソースから表への変換例を図 3 に示す．時系列変化比較表では，キー列に日付 (dt) と時刻 (hh) に関するキーパラメータをもつ TimeAxis，非キー列に限定子が指定されたラベルを設定することで，各行に各時間帯のメトリクス値がはいった表を構成している．時間別変化比較表ではキー列に DateAxis，非キー列に各時刻のキーパラメータを含んだラベルを設定することで，日毎に各時間帯のメトリクス値を並べた表を構成している．図 3 に示すように提案手法により，表の定義を変更することで単一のメトリクスソースから様々な表を構成することができる．

図 3 のような変換を実現するためには，メトリクスソースのキーパラメータが適切な粒度で設定されている必要がある．我々は非構造化データの事後的に定義されたスキーマでアクセスする手法を提案している [9]，この手法をもちいることで表が構成しやすいキーパラメータをもったデータとしてモデル化することができる．また，キーパラメータ内の用語も統一されているが，データの標準化にもあわせて取り組んでいる [10]．

時系列変化比較表

TimeAxis	('00', {},'val')	...
('2017-03-06 00', {'dt':'2017-03-06', 'hh':'00'}, null)	100	...
('2017-03-06 00', {'dt':'2017-03-06', 'hh':'01'}, null)	200	...
:	:	:

時間別変化比較表

DateAxis	('00', {'hh':'00'},'val')	('00', {'hh':'01'},'val')	...
('2017-03-06', {'dt':'2017-03-06'}, null)	100	200	...
('2017-03-07', {'dt':'2017-03-07'}, null)	110	210	...
:	:	:	:

メトリクスソース

key parameter	val
{'dt':'2017-03-06', 'hh':'00'}	100
{'dt':'2017-03-06', 'hh':'01'}	200
:	:
{'dt':'2017-03-07', 'hh':'00'}	110
{'dt':'2017-03-07', 'hh':'01'}	210
:	:

図 3　表形式への変換

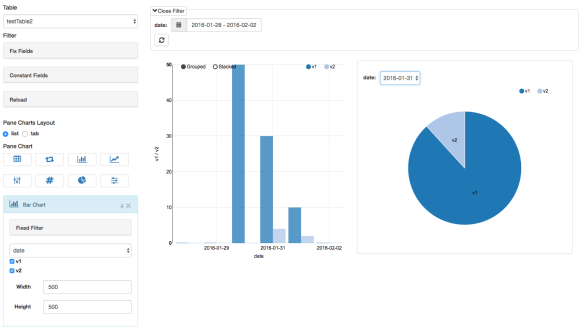

図 4　レポート編集画面

```
CREATE TEMPORARY METRICS TABLE tmpTable (
  dt TIMESERIES('daily', 'yyyy-MM-dd',
                {'dt': 'yyyy-MM-dd'}),
  v1 LABEL(NULL, {'k': 'v1'}),
  v2 LABEL(NULL, {'k': 'v2'})
) SOURCE local;

CREATE VIEW sumTable (dt, v1, v2, sum) AS
SELECT
  dt,  v1, v2, add(v1,v2)
FROM
  tmpTable0
```

図 5　表定義 DSL

3. 実　　装

ここでは提案手法に基づいて実装したレポーティングツールについて説明する．

実装したレポーティングツールは React フレームワーク [2] を用いた Single-Page-Application として実装されている．サーバサイドではメトリクスソースからのデータ取得と表形式への変換を行い，グラフによる表示などの可視化はクライアントサイドで実装されている．本手法ではサーバ-クライアント間のデータが表形式に統一されるため，可視化ロジックを汎用性の高い React コンポーネントとして実装できる．

3.1　表定義 DSL

実装したツールでは SQL ライクな DSL によりテーブルを定義できる．この DSL を用いることで，算術演算の適用や複数の表の結合などメトリクスソースから抽出したデータに様々な処理を適用した新たな表を定義可能である．これにより，クライアントサイドでの処理を簡略化し，可視化ロジックを単純化できる．

図 5 では，時間軸 (`dt`) と二つのラベル (`L1`，`L2`) からなる表 (`tmpTable`) に対して，`L1`，`L2` のメトリクス値の合計を加えた新たな表 (`sumTable`) を定義している．

3.2　データの選択

キー列に設定された軸がラベルの無限集合の場合，表のデータは無限集合となる．この場合，可視化対象のデータを選択する必要がある．

このために本ツールでは各軸の型を管理し，型に応じたフィルタをレポート画面に表示する．フィルタは複数の値を選択するものと単一の値を選択するものを切り替えることができる．現在実装している軸の型とフィルタを表 1 に示す．

4. 適　用　例

データソースに応じたメトリクスソースの定義と問い合わせ処理の構成方法について述べる．ここではメトリクスソースをキーパラメータのキーの集合 (K) とすると限定子の集合 (Q) の 2 つ組 ((K, Q)) により定義する．

4.1　関係データベース

本手法は構造化データにも適用できる．関係データベース上のテーブル T のキー属性を $key(T)$，非キー属性を $val(T)$ とすると，T にアクセスするためのメトリクスソース M は $(key(T), val(T))$ と定義できる．また，軸の型は対応するテーブル T の主キーの型に応じて決定できる．このときフィルタからは主キー属性 $key(T)$ に関する述語 P が入力され，必要なメトリクス値はクエリ SELECT $*$ FROM T WHERE P を実行

表 1　軸の型とフィルタ

型	複数値用フィルタ	単一値用フィルタ
時間	日付ピッカー (範囲)	日付ピッカー (日指定)
列挙	チェックボックス	セレクトボックス
文字列	テキストフィールド（正規表現）	テキストフィールド（文字列）

することで取得できる．

4.2 Key-Value ストア

Key-Value ストアへの適用方法はスキーマによって異なる．ここでは，Key-Value の Key にキーパラメータがシリアライズされ，Value にメトリクス値が設定されるスキーマを考える．このスキーマに対応するメトリクスソースは以下のように定義できる．

$$M = (\text{String}, \text{'dummy'})$$

ここではキーパラメータは任意の文字列をとることが可能であり，軸の型は事前に設定しておく必要がある．また，このスキーマではキーパラメータのみでメトリクス値を特定できるため限定子は任意のダミー文字列を用いればよい．メトリクス値を取得するためのクエリは，フィルタ経由で入力される条件 P を選言標準形に変換し，各連言節から取得すべきキーを決定できる．なお，フィルタ条件に正規表現が含まれる場合は，フィルタの位置に応じて Key-Value ストアの範囲走査を行う必要がある．

4.3 REST HTTP サーバ

JSON 形式でレスポンスを返す REST HTTP サーバを考える．REST のリソース識別子のパターンに応じて K を，レスポンスの JSON オブジェクトの属性名に応じて Q を決定すればよい．

メトリクス値の取得は，Key-Value ストアの場合と同様にフィルタ条件 P を選言標準形に変換し，各連言節に対応する HTTP リクエストを送信し，JSON レスポンスから限定子に応じて必要な属性を取得すればよい．

なお，現在，このタイプのメトリクスソースでは正規表現を含むフィルタ条件をサポートしていない．

5. 関 連 研 究

データの可視化手法としては Polaris [7] がよく知られている．Polaris は構造化データをグラフィカルに分析するため手法である．それに対して，本稿で提案する手法は，表構造のデータを構成するための汎用的な手続きを提供することで可視化ツールの拡張性や保守性を改善することを目的としている．

[4] [5] では，複数のデータの関連を可視化する手法が提案されている．これらの手法も表形式のデータを前提としており，軸の情報をもとに複数のテーブルの関係などを解析し，テーブル間の関係を可視化する．提案手法はこのような表形式データを前提とした様々な可視化手法の適用可能性を向上できると考える．また, [5] には軸の付け替えなど表の形式を変換する手続きも含まれているが，日付などを対象とした限定的なものであり，提案手法のように様々なデータソースを統一的に扱えるものではない．

グラフィカルにデータを分析する手法としては, [8] などが提案されているが, [8] ではインタラクティブなデータ分析をおこなうためのフローをグラフィカルに定義する手法を提案している. [8] も扱うデータは表形式であり，このような手法を参考に，多様なデータソースに対してより高度でインタラクティブな分析を行えるツールに拡張することは今後の課題である．

6. ま と め

本稿では非構造化データを与えられた軸情報にもとづいて表形式に変換する手法を提案した．提案手法は 2 段階の Map 構造として可視化対象データを抽象化することで，様々なデータソースへ適用可能となる．様々データソースに記憶されたメトリクス値を統一的な手続きで標準的なデータ構造である表形式に変換できるため，可視化ツールの構成を単純化でき，可視化コンポーネントの再利用性などを向上できる．今後の課題としては，可視化コンポーネントの充実やレポート設定の簡素化をすすめるとともに，データ分析におけるより多様なタスク [6] [3] を支援することを検討している．

文　　献

[1] D3.js - data-driven documents. https://d3js.org/.

[2] A JavaScript library for building user interfaces - react. https://facebook.github.io/react/.

[3] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS '05, pp. 15–, Washington, DC, USA, 2005. IEEE Computer Society.

[4] J. H. T. Claessen and J. J. van Wijk. Flexible linked axes for multivariate data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2310–2316, Dec 2011.

[5] S. Gratzl, N. Gehlenborg, A. Lex, H. Pfister, and M. Streit. Domino: Extracting, comparing, and manipulating subsets across multiple tabular datasets. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2023–2032, 2014.

[6] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages*, VL '96, pp. 336–, Washington, DC, USA, 1996. IEEE Computer Society.

[7] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65, Jan. 2002.

[8] B. Yu and C. T. Silva. Visflow - web-based visualization framework for tabular data with a subset flow model. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):251–260, Jan. 2017.

[9] 善明, 津田. スキーマ定義に基づく sql ライクな Key-Value ストアクライアント. 第 8 回データ工学と情報マネジメントに関

するフォーラム, 2016.
[10] 田中. サイバーエージェントにおけるデータの品質管理について. Cloudera World Tokyo 2016 http://www.slideshare.net/cyberagent/cwt2016.