

## 時間情報を有するデータの ビジュアルの Web プログラ ミング(2)

松原 伸人

[matubara@sra.co.jp](mailto:matubara@sra.co.jp)

### ◆ はじめに

Vol.104 で、開発している年表表示プログラム **orca** を使った年表表示処理の前半部分を紹介しました。Vol.100、102、103 で紹介した年表の表示処理に使われています。

一般的に、年表には出来事を分類して記述し、同時期、同時刻に起こった出来事の位置が揃うように配置します。年表に掲載する情報を集めてきて、分類し、全ての出来事を同じ時刻ごとに位置が揃うように配置していくのは大変な作業です。**orca** は、時間情報に基づいて出来事の位置を揃える作業を自動化します。

今回は、**orca** を使った処理の後半を紹介します。

### ◆ **orca** の表示処理

Vol.104 で紹介したように、**orca** の年表表示では、出来事が時間順に並ぶタイムライン群を、同時刻の出来事の位置が揃うように並べて提示します。<sup>1</sup>

出来事には、ルーラーやタイムセパレーターやグラフなど付加的なデータをつけられるようになっていきます。<sup>2</sup>

<sup>1</sup> GSGLetterNeo Vol.100 をご参照ください。

<sup>2</sup> GSGLetterNeo Vol.102, 103 をご参照ください。

**orca** の表示処理は、大きく分けると、

- 年表データファイルを読み込む処理
- HTMLDOM 要素の生成
- 出来事群の大きさの取得処理
- 出来事の高さを時刻ごとに揃える処理
- 付加的なデータの配置処理

の5つを、順に実行します。

今回は、タイムラインが縦方向の年表を例にして、5つの表示処理の3番目以降を解説します。<sup>3</sup>

タイムラインが縦方向の年表の具体例としては、GSLetterNeoWeb 年表があります。



図 1 GSletterNeoWeb 年表

(<https://www.sra.co.jp/kti/gsletterneo/#chronicle>)

### ◆ 出来事群の矩形領域の取得処理

Web ページの全 DOM 要素は矩形で表されるため、年表の全データも矩形として扱います。

全 Timeline の全 Event について、Event の内容と CSS によるビジュアル表現ルールに基づいた、表示エ

<sup>3</sup> 1番目と2番目は GSGLetterNeo Vol.104 をご参照ください。

リアの座標面における、Event の大きさを表す矩形領域を計算し、時刻と矩形領域の関係を求めます。

DOM 要素が DOM ツリーに入った Timeline、Times、Event たちの矩形領域は、Web ブラウザが計算を行ってくれます。

DOM ツリーに入った DOM 要素の矩形領域は、getBoundingClientRect 関数を呼ぶと計算が実行されます。

getBoundingClientRect 関数は、表示エリアの左上を原点とする、矩形の左上角の水平方向の位置 left、垂直方向の位置 top、矩形の幅 width、高さ height を算出します。

Times の矩形領域は、内包する Event 群の矩形領域と配置方法によって決まります。

Event 群の配置方法も CSS によって記述しています。CSS の記述方法を変えることで、縦書きの年表作成も実現できます。(縦書き年表の例 GSLetterNeoVol.100 の図 1)

#### おまけ1:

この getBoundingClientRect 関数を使うときは、実行時間がかかるため、呼び出すタイミングやキャッシュするなど工夫が必要です。

#### おまけ2:

CSS で 3D 空間上に HTML を表示する場合に DOM 要素のサイズを得るには、clientWidth 及び clientHeight プロパティを参照します。getBoundingClientRect が算出する矩形領域は、ディスプレイ表示面における大きさであるため、3D のアングルや、DOM 要素の向きにより変化します。

### ◆ 出来事の高さを時刻ごとに揃える処理

全 Timeline には、それぞれ、時間順に Times が並びます。

時刻ごとに Times の高さを揃えると、時間ごとに出来事を揃えた年表が出来上がります。

まず同時刻ごとに Times を集めます。

時刻ごとに収集した Times 群から矩形領域の高さの

最大値を検出します。

検出した高さを各 Times の高さに設定します。

このようにして、出来事をタイムラインごとに時間順に並べ、同じ時刻ごとに出来事の位置と大きさを揃えて、年表表示しています。

### ◆ 付加的なデータの配置処理

orca 年表には、タイムライン以外に、ルーラー、タイムセパレーター、グラフの 3 種類のデータを記述できます。<sup>4</sup>

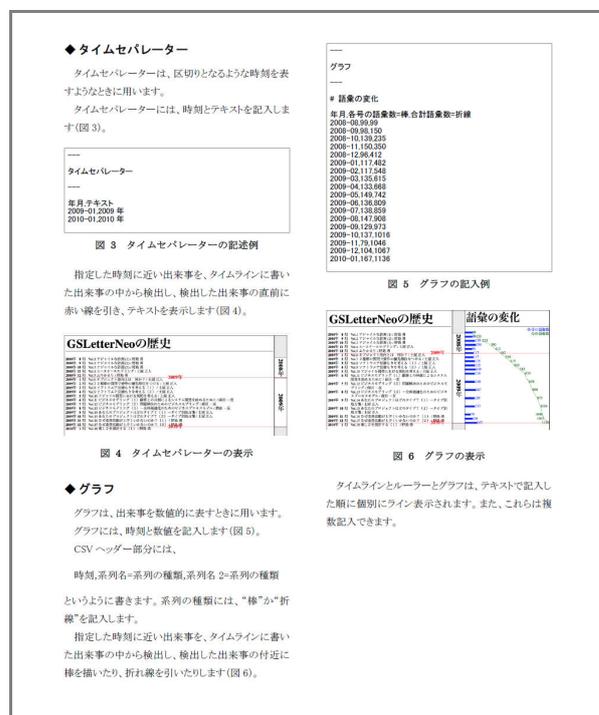


図 2 GSLetterNeo Vol.103 より 付加的なデータの記述

この 3 つは、前述の処理によって得られた、出来事の時刻と矩形領域の関係を参照し、表示位置を決定します。

ルーラーは、指定した開始時刻と終了時刻で表される時間範囲にラベルをつけます。

出来事群から開始時刻と終了時刻の範囲に含まれる出来事のうち、開始時刻に近い出来事と、終了時刻

<sup>4</sup> GSLetterNeo Vol.100, Vol.103 をご参照ください。

に近い出来事の 2 つの矩形領域から、ルーラーのラベルの矩形領域を計算しています。

タイムセパレーターは、年表の切れ目となるような時刻を指定して線を引きます。

指定した時刻に近い出来事を検出し、その出来事の矩形領域の直前の位置(縦方向の年表の場合 top)に、年表を横断する線を引きます。

グラフは、時刻と数値データ群で表されます。

折れ線グラフと棒グラフを表示できるようになっています。

折れ線の節の位置や棒の位置が、時刻に基づいて決まります。

タイムセパレーターと同様の方法で、グラフデータの時刻に近い時刻の出来事を検出し、矩形領域の垂直中央  $(top + height / 2)$  の位置を表示位置とします。

## ◆ 終わりに

時間情報を有するデータを時間軸上にプロットして表す方法を、GSLetterNeo Vol.92「カタマリを見つけたい履歴データブラウザ(2)」で紹介しました。

時間軸に沿ってデータをプロットするよう画面に出力するには、データ全体の開始時間と最小値、終了時間を最大値とし、時間を入力すると軸上の位置を出力する直線の式で表すことができました。

ここで言うところの時間軸は、時間軸の開始時刻  $st$  終了時刻  $et$  として、時間の刻みを一定間隔で表し、高さ  $h$  の画面上の垂直方向の座標を  $y$  とすると、時間  $t$  の線形式で表わせました。

$$y = (t - st) / (et - st) * h$$

この表示方法は、時間的に近い出来事はまとまり、遠い出来事は離れるため、パッとみて時間的に連続しているような一連の出来事を探しやすいです。

一方、年表表示では、着目している出来事の前後の出来事や、同時刻の異なるタイムラインの出来事が近くに並びます。

前者と比べると、出来事のまとまりや切れ目は分かりにくくなりますが、同時刻や同時期に起きていた出来事を見つけやすいです。

全然関係のない記録を並べてみて、今まで気づかなかった関係が見えたり、出来事の因果関係を想像するきっかけになったりするといいなあと思いました。

GSLetterNeo Vol. 105  
2017 年 4 月 20 日発行  
発行者 ● 株式会社 SRA 先端技術研究所  
編集者 ● 土屋正人

バックナンバーを公開しています ● <http://www.sra.co.jp/gsletter>  
ご感想・お問い合わせはこちらへお願いします ● [gsneo@sra.co.jp](mailto:gsneo@sra.co.jp)

夢を。



**株式会社SRA**

〒171-8513 東京都豊島区南池袋 2-32-8

夢を。Yawaraka Innovation  
やわらかいのバージョン