SVG によるユーザインタフェース構築の検討
A Study on Development of SVG Based User Interface

前田和昭
Kazuaki Maeda
中部大学　経営情報学部
Faculty of Business Administration and
Information Science, Chubu University

要旨

２次元グラフィックスを Web ページ上に描くために SVG が広く使われている．SVG は，ベクター画像を XML で記述するように設計されているため，プログラムから操作することで多様な表現が可能となる．本稿では，SVG を使ってユーザインタフェースを構築してきた経験を踏まえ，SVG の概要を説明し，品質の良いユーザインタフェースを作り上げるために行った性能評価実験について述べる．

1. はじめに

インターネットの爆発的な普及は，Web ページを記述するための言語 HTML とそれを表示する Web ブラウザがきっかけとなって始まった．HTML は，文書中に印刷命令を意味するタグを埋め込むためのもので，それを解析・整形して，ページを表示するのが Web ブラウザである．これに加え，最近の Web ブラウザには，SVG(Scalable Vector Graphics) によるベクター画像を表示する機能が実装されている．

SVG は，ベクター画像を作成するための XML ファイル形式であり，W3C により仕様が作成されている．SVG 1.0 が 2001 年に，SVG 1.1 が 2011 年に勧告として公開された [1, 2]．開発者らによって保守されている Web サイト [3, 4] によれば，代表的な Web ブラウザには，SVG 1.1 を満たす機能が実装済みと報告されている．

SVG を説明するために，SVG による簡易な記述を図 1 に示す．この記述を Web ブラウザが読み込むと，図 2 で示す円・線分・テキストの組み合わせが表示される．SVG は，Web ブラウザで表示されるためだけではなく，グラフィックスツールで読込み・保存できる．今では，グラフィックスツール間でデータを交換するためにも使われている．

```
<svg width='400' height='200'>
  <circle cx='200' cy='100' r='80' fill='none'></circle>
  <line x1='200' y1='20' x2='200' y2='180'></line>
  <text x='40' y='160'>晴れ</text>
</svg>
```

図 1: SVG で円・線分・テキストを表示する例

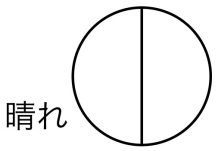

図 2: Web ブラウザでの表示結果

本稿では，SVG を使ってユーザインタフェースを構築してきた経験を踏まえ，品質の良いユーザインタフェースを作り上げるための調査結果について述べる．

2. SVG の３つの使い方

筆者のこれまでの経験から，SVG は

1. 静的に使う
2. アニメーションに使う
3. 動的に使う

の３通りの場合があると考える.「静的に扱う」ときは，Adobe 社 Illustrator などのグラフィックツールを使って絵を描き，SVG で保存する場合である．例えば，図 3 に，橋本氏が描いた東京近辺の鉄道路線図の一部を示す [5]．この路線図はパブリックドメインで公開されているため，SVG を説明するためのデモや，JavaScript プログラムで操作する素材として使うことを検討した．しかし，Adobe 社 Illustrator を使って丹念に生成されたベクタ画像を SVG で保存したファイルは，ファイル 1 つが約 9,000 行の記述からなり，これを JavaScript プログラムから操作することは困難と判断した．

「アニメーションに使う」ときは，SVG で記述された特定の要素に対して，例えば animateTransform 要素を使い，移動や回転，変化の始点・終点，変化の開始・終了タイミングを指定して，動きを表現する場合である．図 4 に，書籍 [6] にて紹介されているニュートンのゆりかごの例を示す（説明のために筆者が修正した）．このアニメーションでは，SVG にしたがった XML のみで動きを記述し，タイミングを合わせて緑色の線分とそれに接続する赤丸を左右に 30 度ずつ交互に回転させている．

図 3: 鉄道路線図の一部

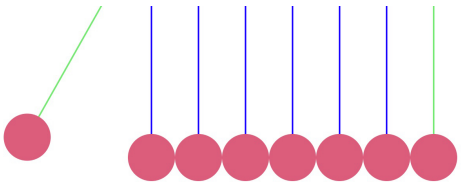

図 4: アニメーションの例

3 つめの「動的に使う」ときは，定期的（例えば毎秒毎）に表示を更新する場合である．性能評価のために筆者が作成した，毎秒針の表示を更新する時計の例を図 5 に，毎秒データを更新する棒グラフの例を図 6 に示す．

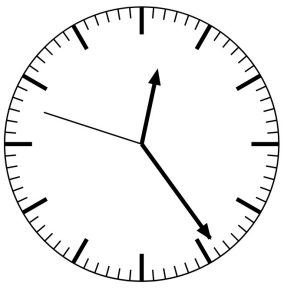

図 5: 毎秒表示を更新する時計

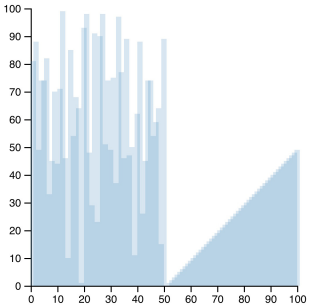

図 6: 毎秒更新する棒グラフ

次節では，3 つめの「動的に使う」ときのみを対象とし，SVG で作成されたベクター画像を，JavaScript を使って決められたタイミングで更新していくプログラムを使って性能評価を行った結果について報告する．

3.JavaScript プログラムによる Web ブラウザの性能測定

これまで SVG を使ってユーザインタフェースを描いていた経験では，Web ブラウザを変えると動作速度が異なることを何度も感じることが多かった．この動作速度について調査をしたものの，SVG を使ってデータを表示する場面で，Web ブラウザによる違いを示した研究論文または開発記事が見つからなかった．そこで，各 Web ブラウザでの実行時間を計測するための JavaScript プログラムを作成し，代表的な Web ブラウザを使って性能評価を実施することにした．

3.1.SVG を活用する JavaScript プログラムの Web ブラウザ毎の性能評価実験

実験を行うハードウェアとして，以下の仕様のノートブック PC を 3 台準備した．それらの購入時期が 2017 年，2016 年，2014 年であることから，それぞれ PC 2017，PC 2016，PC 2014 と名付けることにする．まずは，これら全ての PC に内蔵された HDD または SSD を初期化した後，Windows 10 をインストールし，その後ウィルス対策ソフトウェアなど実験に関係ないソフトウェアは一切インストールしないことにした．

- PC 2017 – Intel Core i7 (7820HQ) 2.90GHz，主記憶：16GB，Windows 10 Pro, Version 1703
- PC 2016 – Intel Core i5 (6200U) 2.30GHz，主記憶: 8GB，Windows 10 Home, Version 1803
- PC 2014 – Intel Core i5 (4308U) 2.80GHz，主記憶: 16GB，Windows 10 Pro, Version 1803

さらに，以下に示す代表的な 6 つの Web ブラウザをインストールして，それらの上で筆者が作成した SVG データを含む Web ページを表示し JavaScript プログラムを動かして，実行時間を測定した．

- Google Chrome, Version 70.0.3538.110
- Mozilla Firefox, Version 63.0.3
- Vivaldi, Version 2.1.1337.51
- Opera, Version 56.0.3051.116
- Internet Explorer (IE と略記する), Version 11.1418.15063.0
- Microsoft Edge, Version 40.15063.674.0

性能測定のために作成した JavaScript プログラムは以下の 3 種類である．

- 時計 (SVG を主) と時計 (D3 を主)

  2 つの乱数を生成し，その値にしたがって図 5 で示す時計の長針・短針を回転させる．この時計を同一 Web ページ上に 9 個配置して，長針・短針の回転処理を 100 回行い，プログラム実行開始前と終了後の時間の差から実行時間を計算する．さらに，3 秒に 1 回 Web ページ全体を更新する処理を 100 回実行して平均値を求める．実装の違いによる性能を評価したかったので，ベクター画像の大部分を SVG で描くことで JavaScript のプログラム量を最小限にしたもの（時計 (SVG を主) と記す）と，SVG による記述を最小限にして大部分を D3[7] の API を使って JavaScript で記述したもの（時計 (D3 を主) と記す）の 2 つを準備した．

- 棒グラフ

  乱数 50 個を生成し，図 6 で示す棒グラフを更新する．各データは左側（棒グラフの原点）から順番に挿入し，表示されているデータを 100 個に限定するため，右側（棒グラフの x 座標値 100）から削除している．この棒グラフを同一 Web ページ上に 3 個配置して，データの更新処理を 50 回行い，プログラム実行開始前と終了後の時間の差から実行時間を計算する．さらに，3 秒に 1 回 Web ページ全体を更新する処理を 100 回実行して平均値を求める．

表 1: 各 Web ブラウザ毎の JavaScript プログラム実行時間（単位：ミリ秒）

		Chrome	Firefox	Vivaldi	Opera	IE	Edge
PC 2017	時計 (SVG を主)	17.77	29.79	18.54	20.09	102.19	126.10
	時計 (D3 を主)	22.03	31.13	21.88	29.76	108.05	147.66
	棒グラフ	108.86	233.64	110.56	97.88	433.12	360.06
PC 2016	時計 (SVG を主)	13.78	16.24	15.46	15.13	90.91	103.42
	時計 (D3 を主)	15.20	16.07	15.29	15.85	84.74	100.49
	棒グラフ	108.08	197.7	111.30	103.08	447.88	517.08
PC 2014	時計 (SVG を主)	28.31	37.40	33.80	33.23	97.73	126.10
	時計 (D3 を主)	28.58	41.46	35.94	40.60	100.12	132.06
	棒グラフ	171.42	315.64	163.10	164.80	510.70	586.94

性能評価実験で求めることができた実行時間を表 1 に示す．この結果から，IE と Edge のときが他に比べて処理時間がかかっていることが分かる．また，Chrome，Vialdi，Opera の実行時間が似た値を示していることは，Web ブラウザでの表示機能を実装しているレンダリングエンジンが同じであるという事実を示唆しているものと思われる．表 1 で示した実験結果を求める前段階で，同一 Web ページに複数の SVG データを置いてから，1 秒に 1 回の頻度で表示を更新していた．しかし，Web ブラウザによっては更新のタイミングに追いつくことができず異常な値を示していたため，3 秒に 1 回の頻度で実験を行うことになった．これら結果として出てきた Web ブラウザ間での優劣は，筆者が経験したことに近い感触を得ている．

4. まとめ

本稿では，SVG を使ってユーザインタフェースを構築してきた経験を踏まえ，SVG の概要を説明し，SVG に基づくベクター画像を表示するときの各 Web ブラウザ毎の性能評価について述べた．現段階で示すことができる性能評価実験の結果は，各 Web ブラウザの限定した特徴を示しているにすぎない．今後さらなる改良と実験を続けていく予定である．

参考文献

[1] Scalable Vector Graphics (SVG) 1.0 Specification, https://www.w3.org/TR/2001/REC-SVG-20010904/ (accessed on Nov. 29, 2018).
[2] Scalable Vector Graphics (SVG) 1.1 (Second Edition), https://www.w3.org/TR/SVG11/ (accessed on Nov. 29, 2018).
[3] Can I use... Support tables for HTML5, CSS3, etc, https://caniuse.com/#feat=svg (accessed on Nov. 29, 2018).
[4] HTML svg tag, https://www.w3schools.com/tags/tag_svg.asp (accessed on Nov. 29, 2018).
[5] 東京の鉄道路線図 SVG を作りました＆パブリックドメインで配布します - Liner Note, https://note.openvista.jp/2014/svg-rail-map (accessed on Nov. 29, 2018).
[6] Damian Dawber, Learning Raphael JS Vector Graphics, Packt Publishing (2013).
[7] D3.js - Data-Driven Documents, https://d3js.org/ (accessed on Nov. 29, 2018).

付録：本稿で紹介した SVG 例の保管 URL

- 図 4：ニュートンのゆりかご：http://jsfiddle.net/kmaeda/c5uqb9xt/4/
- 図 5：乱数で長針・短針を更新する時計：http://jsfiddle.net/kmaeda/2d7a0kqs/3/
- 図 6：乱数データを挿入していく棒グラフ：http://jsfiddle.net/kmaeda/bjr23twk/4/
- 大部分を SVG で書いた時計の文字盤：http://jsfiddle.net/kmaeda/ksq1oc6v/6/
- 大部分を D3 で描画する時計の文字盤：http://jsfiddle.net/kmaeda/v5utg6rq/6/