

Web ブラウザを利用した効果的なプレゼンテーション支援 に関する検討

A Study on Impressive Presentation Supports with Web Browsers

前田和昭

Kazuaki Maeda

中部大学 経営情報学部

School of Business Administration and Information Science, Chubu University

要旨

本稿では、最新の Web ブラウザを利用したプレゼンテーションについて概説し、今後開発すべき支援ツールについて検討する。学会発表や打合せなどでプレゼンテーションを実施するとき、PowerPoint に代表される従来型のプレゼンテーション・ツールで作成したスライドを使うのが一般的である。しかし、従来型のプレゼンテーション・ツールは、Web 以前の時代に設計されたものであり、Web への親和性に乏しい。これに対して、最新の Web ブラウザを使うことで Web で提供されているサービスを全て利用したプレゼンテーションを実現できる。しかし現段階では、HTML タグを手入力しがなく、支援ツールも出始めたばかりで未だ実用的とはいえない。誰もが効果的なプレゼンテーションを気軽に実現できるように優れたツールの登場が待ち望まれている。

1. はじめに

学会発表や打合せなどでプレゼンテーションを実施するときプレゼンテーション・ツールとして PowerPoint を使うことが多い。Businessweek の記事 [1] によれば、世界中で少なくとも 10 億台のコンピュータにインストールされ、世界中で毎秒 350 回のプレゼンテーションが行われていると推定されている。PowerPoint は、20 年以上の長い年月を経て進化し大勢の利用者が使い続けている素晴らしいアプリケーションである。現在、Windows と Mac のためのアプリケーションがリリースされているものの、急激に利用者が増えているタブレットでは基本的に利用できない。さらには、現在の Web 中心の世界に追随しているとは言い難く「Web フレンドリ」ではない。ここで言う Web フレンドリとは、Web で利用可能なサービスをそのまま全て利用できることを意味する。

プレゼンテーションのための資料として PowerPoint でスライドを作成していると、Web ブラウザに表示された内容をそのまま取り込みたいことがある。例えば、スライドに Wikipedia ページを表示したいとき、その Web ページのスナップショットをコピーしてスライドに貼り付けるのが一般的な方法であろう。また別の方法として、Web ブラウザへのリンクを埋め込む方法もあり得る。この場合、埋め込まれたリンクをクリックするとスライドショーは停止し、Web ブラウザが起動して独立したウィンドウ（またはタブ）で Web ページを表示することになる。これは、PowerPoint が Web フレンドリでなく、Web が提供する様々なサービスをそのまま利用できないことを回避しているにすぎない。日常生活のあらゆる場面で Web を使ったサービスが当然のように使われる現状に対して、PowerPoint の機能不足が目立ってきていると思われる。

PowerPoint でスライドを作成した後、スライドの修正やスライド・ショーの実施のためには PowerPoint が必要となる。最近では、Keynote on iCloud[2] のようなツールが登場し、Web ブラウザ上でスライドを作成して、Web ブラウザ上でスライドショーを実施することができる。しかし、Keynote on iCloud は Web 上のサービスを全て使えるわけではなく、Web フレンドリとはいえない。

PowerPoint で作成したスライドを Web 上で公開するための素晴らしいサービスとして SlideShare[3] がある。自分用のアカウントにログイン後、PowerPoint ファイルをアップロードするだけで、自動的に変換処理され、写真ビューアのように閲覧できる Web ページが生成される。筆者がスライドを公開した経験では、アップロード後にスライド・ビューアの下にスライド内の全てのテキストを羅列したページが生成された。これは、Web クローラに、スライド内の全テキストを検索情報として与えることが主とした理由であろう。

最近、ズームング・ユーザ・インタフェース (Zooming User Interface, 以下 ZUI と記す) を使ったプレゼンテーション・ツールとして Prezi[4] が注目されている。二次元空間上にスライドを配置し、拡大縮小を指定して、ズームイン・ズームアウト効果を使ったスライドショーを実施できることが特徴である。prezi.com での記述によれば、年々利用者数が増え続け、2013 年第 2 四半期のユーザ数は 5 億人であり、その中の登録済みユーザ数は 2,600 万人となっている。Prezi でスライドを作成するには、ユーザ登録が必要となる。さらにデスクトップ上で利用できるアプリケーションを使うには有料会員になる必要がある。

最新の Web ブラウザには様々な機能が実装され、その機能をうまく使ってプレゼンテーションを実施するケースが増えてきている。Web ブラウザを使うことで Web で提供されるサービスを全て利用することができる。しかし、利用者のほとんどは HTML コードを手入力するしかなく、支援ツールも出始めたばかりで未だ実用的とはいえない。本稿では、Web ブラウザを使って印象的なプレゼンテーションを実施できるライブラリの一つとして impress.js を紹介し、その支援ツールについて検討する。

2. Web ブラウザを利用したプレゼンテーション

HTML と JavaScript の上手な組み合わせは、Web アプリケーションを高機能にする。さらに、CSS3 のトランスフォーメーションとトランジションを使えば、拡大縮小と回転の機能を簡単に指定できる。

これらを活用して Web ブラウザ上で印象的なプレゼンテーションを作り上げるためのライブラリの一つとして impress.js がある。impress.js の作者は Bartek Szopka であり、github から無料でダウンロードできる [5]。最初の版は、MIT と GPL ライセンスの下で 2011 年 12 月に公開された。JavaScript で記述された impress.js は、他のライブラリにも依存することがなく単独で利用できる。800 行程度の小さなライブラリであるにもかかわらず、Prezi のような ZUI によるプレゼンテーションを可能にする。

筆者は、impress.js を使ったプレゼンテーションを何度も行ってきた。その経験から、impress.js によって可能となる ZUI は、現状のプレゼンテーション・ツールの状況を一変する能力を持っていると感じている。impress.js を使えば、CSS が提供する強力なトランスフォーメーションとトランジションを活用してスライド・ショーを実施できる。座標を指定することで、スライドを 3 次元空間の任意の場所に置くことができ、さらに、スライドに拡大縮小と回転を指定することができる。

impress.js を使ったスライド記述の概略を説明するために、鳥を模倣して各スライドに一つの数字を描き、それをスライド・ショーで一つずつ表示している様子を図 1 に示す。また、図 2 に、図 1 で示した鳥の HTML コードの一部を示す。HTML タグ内に impress を id として持つ div 要素が、スライド全てを含む「入れ物」となる。この入れ物の内部にあり、かつ、step クラスを持つ div 要素一つが、一枚のスライドとなる。図 1 で示した鳥は、6 つの文字 o, 1, 2, 3, 5, 8 に対して、拡大縮小回転を組み合わせで作られている。最初に全体を俯瞰するスライドが一枚あり、その後一文字からなるスライドが 6 枚ある。スライド記述のあと、impress.js がロードされて、初期化のための impress().init() が実行される。

各種データをスライドに割り当てるために、div 要素に決められたデータ属性を書き込む。各スライドの座標は、3 つの属性 data-x, data-y, data-z を使って三次元空間上で指定する。もし座標を指定



図 1: 数字を使って作成した鳥のスライド・ショー

```

<div id="impress">
  <div class="step overview" data-x="0" data-y="0" data-scale="0.7"></div>
  <div class="step zero" data-x="-10" data-y="100"> 0</div>
  <div class="step one" data-x="10" data-y="0" data-rotate="-140">1</div>
  <div class="step two" data-x="20" data-y="50" data-rotate="-30"> 2</div>
  <div class="step three" data-x="145" data-y="25" data-rotate="-20"> 3</div>
  <div class="step five" data-x="80" data-y="40" data-rotate="180"> 5</div>
  <div class="step eight" data-x="85" data-y="-45" data-rotate="245"> 8</div>
</div>
<script src="impress.js"></script>
<script>impress().init();</script>

```

図 2: 鳥を描くために座標と回転角度を埋め込んだ HTML コード

する属性が省略された場合，一つ前のスライドの座標値が使われる．デフォルトの設定では，各座標の値は 0 になっている．スライドを拡大縮小するためには，data-scale を使う．data-scale 属性の値が一つ前のスライドより大きければ，スライドを切り替えるときに表示サイズが調整され，一つ前の場所から離れるような効果となりズームアウトとなる．また逆に，data-scale 属性の値が一つ前のスライドより小さければ，表示サイズが調整されて，一つ前の場所から近づく効果となりズームインとなる．さらには，3つの属性 rotate-x, rotate-y, rotate-z を使うことで，X,Y,Z 軸に沿ってスライドを回転させることができる．表 1 に，impress.js で準備されている基本的なデータ属性を示す．

表 1: impress.js のためのデータ要素

属性	説明	属性	説明
data-x	スライドの x 座標	data-rotate-x	X 軸中心の回転角度
data-y	スライドの y 座標	data-rotate-y	Y 軸中心の回転角度
data-z	スライドの z 座標	data-rotate-z	Z 軸中心の回転角度
data-scale	スライドの拡大縮小率	data-rotate	data-rotate-z と同じ

各スライドを見栄えを調整するときには，CSS を使って指定する．数字を使った鳥を作成したときは，図 3 にあるように，フォントサイズを指定するために 6 つのクラス (zero, one, two, three, five, eight) を作った．impress.js では，特別な 3 つのクラス (future, present, old) が準備され，自動的に各スライドへ割り当てられる．最初に impress.js を初期化する init() が実行されると，step クラスを持つ全ての要素に future クラスが埋め込まれる．スライドが表示されると，表示されたスライドから future クラスが削除され，present クラスが追加される．そのスライドが最前面から去ると，present クラスが削除され，past クラスが追加される．図 1 に示したスライド・ショーを実施するとき，各数字は順番にブラウザ・ウィンドウの中央に移動して通常の向きで見えるようになる．さらには，present クラスに red を指定しているため，最前面スライドの数字だけが赤色に変わる．

```

.zero { font-size: 14px;}
.one { font-size: 72px;}
.two { font-size: 120px;}
.three{ font-size: 140px;}
.five { font-size: 170px;}
.eight{ font-size: 130px;}
.present { color: red;}

```

図 3: 鳥を描くための CSS 記述

3. 検討

impress.js を使ったスライドを作成するとき，ほとんどの利用者は HTML コードを直接入力している．Eclipse や NetBeans などの IDE を使ってスライドを作成すれば，HTML コード支援があり入力が楽に

なるものの、入力したタグの複雑さは変わらない。この複雑さを解消するため、`impress.js` を使ったスライド作成のためのツールがいくつかリリースされているものの、決定的なツールは未だ見当たらない。

HTML コードの複雑さに対処するアプローチの一つは Markdown を使うことである。Markdown を使えば、空行で改段落を指定したり、タグの代わりに記号を使うなど、簡素化した記述が可能となる。Markdown の記述から `impress.js` 向けの HTML を生成できるツールとして S9[6] がある。S9 を使うと、例えば以下のコマンド

```
$ slideshow build impress.js.text -t impress.js
```

を入力することで、`impress.js` のための出力を指定しているため、Markdown で記述された `impress.js.text` を読み込み HTML ファイルを生成できる。

別のアプローチとして、WYSIWYG スライド作成ツールの Impressionist[7] がある。Web ブラウザ上で動作し、ワープロ感覚でスライドを作成できる。しかし、まだ開発の初期段階にあり `impress.js` の機能をうまく活用できているとはいえない。

筆者は、PowerPoint ファイルを入力して、そのファイル内部を解析し、位置情報などの属性を付加した上で、`impress.js` 向けの HTML を生成するツールを試作中である。試作ツールが生成したスライドの例を図 4 に示す。現在、PowerPoint ファイル以外に何を入力とすべきか、内部でデータをどのように保持すべきか、`impress.js` のための HTML コード以外に何を出力とすべきかを検討中である。支援ツールの完成度が、ある一定のレベルにまで達した後、一般公開することを考えている。



図 4: 前田が試作中のツールが生成したスライドの表示例

4. おわりに

本稿では、最新の Web ブラウザを利用したプレゼンテーションについて概説し、今後開発すべき支援ツールについて述べた。Web ブラウザを使うことで Web で提供されているサービスを全て利用しながら、CSS を活用した印象的なプレゼンテーションを実現できる。今後、Web ブラウザを活用するための支援ツールの開発を進めていき、いろいろな応用を通して評価を進めていきたい。

参考文献

- [1] Bob Parks, “Death to PowerPoint!,” Bloomberg Businessweek, businessweek.com, 2012-08-30, retrieved 2012-09-06.
- [2] Keynote on iCloud, <http://www.icloud.com/> (accessed at October 26).
- [3] Upload & Share PowerPoint presentations, documents, infographics, <http://www.slideshare.net/> (accessed at October 26).
- [4] Prezi - Ideas matter, <http://www.prezi.com/> (accessed at October 26).
- [5] bartaz/impress.js GitHub, <https://github.com/bartaz/impress.js/> (accessed at October 26).
- [6] impress.js - S9 Template Pack by slideshow-s9, <http://slideshow-s9.github.io/slideshow-impress.js/> (accessed at October 26).
- [7] harish-io/Impressionist GitHub, <https://github.com/harish-io/Impressionist/> (accessed at October 26).