

DSL を利用したスライドの効果的なデザイン支援に関する検討

A Study on an Effective Support of Slide Design with DSL

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、Web ブラウザを利用したプレゼンテーションについて、また、効果的にスライドを作成するための DSL (Domain-Specific Language) について述べる。研究発表などで使われる PowerPoint は、Web 以前の時代に設計されたものであり、Web との親和性に乏しい。これに対して、最新の Web ブラウザを利用してスライドを表示すれば、Web との親和性が高くなる。しかし、HTML タグを手入力する面倒な作業が必要となる。そこで、DSL を使ってスライドを記述し、支援ツールにより生産性を向上させることについて検討する。

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、スライド作成ツールとして PowerPoint を使うことが多い。PowerPoint は、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。しかし、現在の Web 中心の世界に追隨しているとは言い難く、作成したスライドを公開する場面では扱いにくいことがある。

データを公開するときの理想的な姿について、Tim Berners-Lee が提唱する 5 star open data[1] が参考になる。彼は、公開のためのデータ形式に関する 5 つの分類を提唱している。

- 1 star データ形式は何でも良いので、オープンライセンスの下で Web 上に公開されるデータ
- 2 star 独占的な形式で公開され再利用可能な構造化データ (例: Excel データ)
- 3 star 非独占的な形式で公開されるデータ (例: CSV データ)
- 4 star URI を使って、Web からデータ内部へアクセス可能なデータ
- 5 star データ内部から Web 上の他へのリンクがあるデータ

これら 5 つの分類の中で、データを公開するためには、5 star であることが望ましいと Tim Berners-Lee は主張している。プレゼンテーションのために作成したスライドは、文書化される前の貴重なアイデアや知恵が含まれている場合が多く、5 star に近づけることを考慮した上で最適な形式で公開すべきであろう。

PowerPoint で作成したスライドから、Web 上で公開するための Web ページを生成するサービスとして SlideShare[2] 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、SlideShare の Web サイトから閲覧できるようになる。生成される Web ページの上部には (PowerPoint ファイル内に書かれていたリンクが埋め込まれて) スライドビューが表示され、さらに下部にはスライド内から抜き出された全てのテキストが羅列され修正可能になっている。

ズームブルユーザインタフェース (Zoomable User Interface, 以下 ZUI と記す) を活用するツール (Google Earth[3] や Prezi[4]) が増えてきている。ZUI では、利用者が詳細を見たり、全体を俯瞰したりするために表示領域を拡大縮小する。最近では、三次元空間での拡大縮小や回転の機能が Web ブラウザに装備されるようになり、その機能を使って ZUI を作り上げることができる。スライドショーを Web ブラウザで実施する場合でも、ZUI を作り上げるライブラリが提供され始めている。しかし、スライド作成者は HTML と CSS を駆使して ZUI を実現することがほとんどで、スライド作成のときの PowerPoint に匹敵するような実用的なツールは未だ存在しない。

本稿では、公開のためのデータ形式について議論した後、Web ブラウザ上で ZUI を駆使したスライドショーを実施するためのライブラリ impress.js[5] を紹介し、DSL による支援について検討する。

2. 公開のためのデータ形式

PowerPoint を使って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 の内容を表示したいとき、該当ページのスナップショットを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 Web ページを表示することになる。これは、Web が提供する様々なサービスを直接利用できないことを回避しているにすぎず、PowerPoint で作成されるスライドは、そのままでは公開のためのデータ形式として優れているとは言えない。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。

SlideShare を使えば、PowerPoint で作成したスライドから、Web 上で公開するための Web ページを生成することができる。SlideShare で作成される Web ページは、非独占的なデータ形式で公開されるものの、上部のスライドビューの部分だけに着目すると、画像を順番に表示しているにすぎないことから 5 star open data での 1 star となる。しかし、スライドビューに Web ページへのリンクを埋め込むことができることから、5 star の資質も持ち合わせている。また、スライドをダウンロード可能として公開する場合は、データをダウンロードして再利用できるので 2 star と考えて良い。下部にはスライド内から抜き出された全てのテキストが羅列され、手作業でテキストを修正すれば、スライドビューの欠点を補うことができる。

ZUI を使ったスライド作成ツールとして Prezi[4] が注目されている。三次元空間上にスライドを配置し拡大縮小や回転を指定して、効果的なスライドショーを実施できることが特徴である。世界中で利用者が増え続け、2014 年 4 月には利用者数が 4000 万人に達したとの報告がある [6]。Prezi は、各スライド内から Web 上の様々なサービスを利用することができるため、5 star の資質を持っている。しかし、Prezi で作成して公開するスライドは独占的なデータ形式となっているため、2 star とも言える。

SlideShare や Prezi を、5 star open data の分類から考えてみると、複数 star の資質を合わせ持っていることが分かり、各分類には直交する考えが混在しているように思える。そこで、1 star から 5 star にある直交した資質を分離するのが適切と考え、

- オープンライセンスで公開しているか
- 再利用可能な構造化データになっているか
- 非独占的なデータ形式か
- データ内部から他へのリンクが使えるか
- 外部からデータ内部へのリンクが使えるか

の 5 つについて、Yes または No (どちらもあり得るときは Yes-No) を考えて、各データ形式を特徴付けるべきであろう。表 1 に、PowerPoint スライドのファイル公開、SlideShare、Prezi についての特徴を記す。PowerPoint のファイル公開の場合は、ppt 形式は独占的なデータ形式、pptx 形式は非独占的なデータ形式と考え、Yes-No とした。また、SlideShare の場合は、アップロードした PowerPoint ファイルを公開し再利用可能とできるが、Web ページ上のスライドビューは再利用不可能と考え、Yes-No とした。

表 1: 公開のためのデータ形式に関する判定

	オープンライセンス	再利用可能	非独占的形式	内部からリンク	外部からリンク
PowerPoint 公開	Yes	Yes	Yes-No	Yes	No
SlideShare	Yes	Yes-No	No	Yes	No
Prezi	Yes	Yes	No	Yes	No

3.Web ブラウザを使ったプレゼンテーション

筆者は、Web ブラウザを使ったプレゼンテーションを何度も行って来た。その中でも、JavaScript ライブラリ impress.js[5] を使った経験から、impress.js を使ったスライドショーは、スライド作成ツールの現状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、CSS3 のトランスフォーメーションとトランジションを活用し、Web ブラウザ上で ZUI を使ったスライドショーを実施できる。スライド作成時には、HTML ファイル内に座標を記述してスライドを三次元空間に置き、さらに、拡大縮小と回転を指定することでスライド切り替え時に特別な効果を出すことができる。

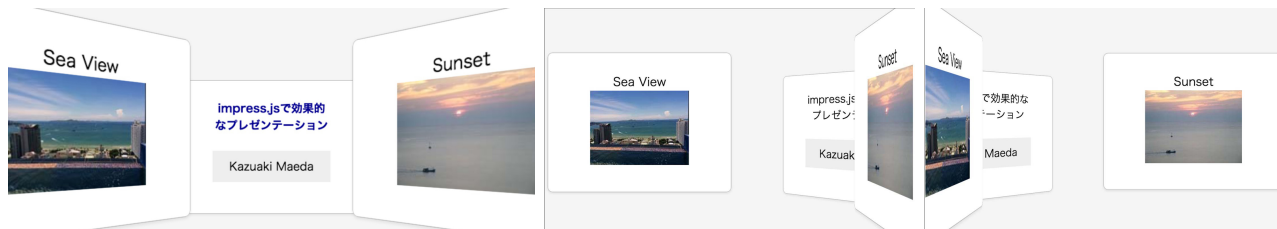


図 1: 3 枚のスライドを順次進めている様子

impress.js を説明するために、3 枚のスライドを順番に 1 枚ずつ表示している様子を図 1 に示し、そのための HTML コードの一部を図 2 に示す。図 2 の HTML コードにある 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 1 つが、スライド 1 枚の記述になる。

```
<div id="impress">
  <div class="step cover" data-y="30" data-z="-400">
    <h1>impress.js で効果的</h1>
    <h1>なプレゼンテーション</h1>
    <br/>
    <h2>Kazuaki Maeda</h2>
  </div>
  <div class="step" data-x="-250" data-rotate-y="50">
    <h1>Sea View</h1>
    <div></div>
  </div>
  <div class="step" data-x="250" data-rotate-y="-50">
    <h1>Sunset</h1>
    <div></div>
  </div>
</div>
```

図 2: impress.js のための座標と回転角度を埋め込んだ HTML コード

impress.js を使って ZUI を駆使した印象的なスライドショーを実施するには、三次元空間上の座標、拡大縮小、回転に関して「巧い値」を指定する必要がある。スライドに値を割り当てるために、決められたデータ属性を div 要素に書き込む。各スライドの三次元空間上での座標は、3 つの属性 data-x, data-y, data-z を使って指定する。もし座標を指定する属性が省略された場合、前のスライドの座標値が使われる。スライドを拡大縮小するには、data-scale を使う。data-scale 属性の値が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、前の場所から離れるようなズームアウト効果となる。また逆に、data-scale 属性の値が前のスライドより小さければ、表示サイズが調整され、前の場所から近づくズームイン効果となる。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。

巧い値を決めてから，HTML ファイル内の属性に順次値を記入していけば，`impress.js` を使って ZUI を駆使した印象的なスライドショーが可能となる．しかし，実際には，HTML と CSS のコードを手作業で書き進める途中で，スライドの追加・削除・修正が必要になることが多い．その場合，隣り合うスライド同士や，全体の配置を見直す必要がでてくる．これは，かなりの手間を要する作業であり，その作業軽減のために，Prezi が提供するようなスライド作成ツールが必要であるものの，現段階では適切なツールが見つからない．そこで，スライド作成ツールを実装することを目標にした最初の一步として，DSL を使った解決方法を検討することにした．

4. スライド記述のための DSL

ここでは，はじめに DSL を概説し，本研究にて開発を進めているスライド記述のための DSL について述べる．

4.1. 外部 DSL と内部 DSL

構造化データを表現することに特化した言語として XML や JSON があり，このように特定ドメインに特化して設計された言語のことを DSL (Domain-Specific Language) と呼んでいる．Martin Fowler の著書 [7] では，DSL に関して以下の 4 つの特徴が述べられている¹．DSL は，

- 人間がコンピュータに処理を指示するための言語であり，
- 流れるような感覚を備えていて，
- 対象となるドメインを記述するための必要最低限の機能を提供し，
- 特定ドメインに集中することで，限定された価値のある言語になる

特徴を持つ．さらにその著書では，DSL を以下の 2 種類に分類している．

外部 DSL 開発すべきアプリケーションの主要言語から独立し，その主要言語とは異なった記述能力を持つ言語（例：SQL, AWK, Struts や Hibernate で利用する XML 設定ファイルなど）

内部 DSL 汎用言語の構文の一部を使って表現される言語（例：Ruby on Rails や Rake の設定ファイルなど）

外部 DSL は，独自の構文を持ち，Java, C++, Ruby などの汎用言語とは全く異なる．HTML 記述の煩雑な作業を緩和するための記述言語として，Jade[9] が実用的に使われている．Jade の場合も，既存のプログラミング言語とは全く違った記述が必要になるため，外部 DSL といえる．

筆者は，構造化データ表現のための DSL，そのスキーマ定義のための DSL，パーザ生成系に入力する構文定義のための DSL などを設計し実験的に使用してきた．これらは汎用言語 Ruby 構文の一部を使った内部 DSL であり，簡潔な記述が可能となっている．本研究で開発を進めているスライド記述のための DSL も，Ruby を土台とした内部 DSL として設計することで，簡潔で，かつ，Ruby プログラムと統一的な記述が可能になるように設計している．

4.2. スライド記述のための DSL

`impress.js` を使えば ZUI を駆使した印象的なスライドショーが可能となるものの，全ての値（三次元空間上での位置，拡大縮小，回転）を HTML コード中に入力する必要がある．作成したファイルには，HTML タグ，表示したい内容（箇条書きの文など），`impress.js` 用のデータ属性が混在して書き込まれ，複雑になる．Eclipse や NetBeans などの IDE を使ってスライドを作成すれば，HTML タグ入力支援があるものの，基本的な複雑さが変わるわけではない．また，スライドを追加・修正・削除するときの，位置関係を見直す厄介な作業は変わらず必要である．さらには，一度作成したスライドを部分的に再利用したい場合，CSS での設定項目を含めて，外観を保持したままスライドの一部を切り出すことは難しい．

¹日本語版 [8] の記述を土台にして筆者が意識した．

そこで，スライド作成ツールを実装することを目標にして，スライド作成のための DSL を設計することから始めることにした．

筆者は，これまでデータを単純に表現するために，構造化データ記述のための DSL を設計し，記述された構造化データをプログラムから扱うためのツールを試作してきた [10],[11],[12],[13]．これらの知見をもとに Ruby を土台として設計している内部 DSL を使った，スライド作成のための内容記述の一部を図 3 に示す．ここでは，図中の

```
author 'Kazuaki Maeda'
```

を要素と呼び，author を要素名，'Kazuaki Maeda' 値と呼ぶ．この要素は単なるデータ表現ではなく，メソッド名が author ，引数が 'Kazuaki Maeda' のメソッド呼び出し（Ruby における括弧を省略した形式）になっている．また，Ruby のブロックを使って階層構造を表現する．図 3 では，deck が 3 つの子要素 cover,slide,slide を持ち，slide が 2 つの子要素 header,body を持つことをブロックを使って記述している．

スライドの内容記述とは別に，三次元空間上の配置や外観を自由に変更できるようにするため，

- スライドの内容
- スライドの配置（三次元空間上の座標，拡大縮小，回転）
- スライドの外観（サイズや色など）

を分離して記述できるようにした．図 4 に，三次元空間上の座標，拡大縮小，回転を記述した例を示す．図 3 で記述した cover,slide,slide の順に対応して結合させ，各スライドに値を設定することができる．例えば，タイトル表示のためのスライドを示す cover は，y 座標 30，z 座標-400 となるように記述されている．これ以外にも，内容記述と配置記述を，id 属性を使って結合させることもできるようになっている．

現在，このように記述した複数のファイルを入力として読み込み，HTML コードを出力する生成系を試作して試行錯誤を繰り返している．この生成系からの出力の一部が図 2 で示した HTML コードである．この他にも，HTML タグを使うための機能（例：図 3 内の br）や，モジュール化機能を使い他のスライド記述を再利用できるようにするための機能などを検討しながら，生成系の改造を進めている．

```
deck {
  cover {
    title 'impress.js で効果的'
    title 'なプレゼンテーション'
    br
    author 'Kazuaki Maeda'
  }
  slide {
    header 'Sea View'
    body {
      img 'img/seaview.jpg'
    }
  }
  slide {
    header 'Sunset'
    body {
      img 'img/sunset.jpg'
    }
  }
}
```

```
config {
  style 'css/sample.css'
  cover {
    y '30'
    z '-400'
  }
  slide {
    x '-250'
    rotatey '50'
  }
  slide {
    x '250'
    rotatey '-50'
  }
}
```

図 3: DSL によるスライドの内容記述

図 4: DSL による三次元空間上の配置記述

5. おわりに

本稿では，作成したデータを公開する視点からプレゼンテーションで使うスライドを検討し，`impress.js` を使った Web ブラウザ上でのスライドショーについて解説した．`impress.js` を使えば，Web で提供されているサービスを全て利用しながら，Web ブラウザの豊富な機能による ZUI を駆使したスライドショーを実施できる．しかし，HTML タグ，表示したい内容，座標データ値を手入力することが必要で面倒な作業となる．そこで，DSL を使ってスライドを記述し，支援ツールにより生産性を向上させることについて検討した．

現在，`impress.js` の著作権から離れ，さらにスライドショーのための機能を追加するために，JavaScript ライブラリをゼロから試作し，その試作ライブラリを使いながら検討を進めている．JavaScript ライブラリを含め，DSL を使ったスライド記述のためのプログラム一式は，オープンソースとして無料で公開することを予定している．今後，日本国内外を問わず研究発表やオープンソース関連の会議にて，著者自身が講演者としてアピールすることで利用者獲得を目指しつつ，多様な意見を聞きながら改良を重ねていきたいと考えている．

参考文献

- [1] 5 star Open Data, <http://5stardata.info/> (accessed on October 29, 2014).
- [2] Upload, Share, and Discover Content on SlideShare, <http://www.slideshare.net/> (accessed on October 29, 2014).
- [3] Google Earth, <https://www.google.com/earth/> (accessed on October 29, 2014).
- [4] Prezi - Presentation Software, <http://www.prezi.com/> (accessed on October 29, 2014).
- [5] bartaz/impress.js - GitHub, <https://github.com/bartaz/impress.js/> (accessed on October 29, 2014).
- [6] パワポキラー Prezi がユーザ数 4000 万に – 急成長は今も続く, <http://jp.techcrunch.com/2014/04/10/20140409prezi-puts-on-10m-users-inside-5-months-hits-40m/> (accessed on October 29, 2014)
- [7] Martin Fowler, Domain-Specific Languages, Addison-Wesley, 2011.
- [8] マーチン・ファウラー, ドメイン特化言語, ピアソン桐原, 2012.
- [9] Jade - Template Engine, <http://jade-lang.com/> (accessed on Oct. 29, 2014).
- [10] Kazuaki Maeda, Ruby-Based Data Representation and the Performance of Java Programs, The 2nd International Conference on the Applications of Digital Information and Web Technologies, pp.8410–846, 2009.
- [11] Kazuaki Maeda, A Scala-Based Domain Specific Language for Structured Data Representation, 5th International Conference on Software and Data Technologies, pp.296–299, 2010.
- [12] Kazuaki Maeda, Executable Representation for Structured Data Using Ruby and Scala, 10th IEEE International Symposium on Communications and Information Technologies, pp.127–132, 2010.
- [13] Kazuaki Maeda, A Domain Specific Language to Represent Structured Data for Multiple Programming Languages, IADIS International Conferences Informatics 2011, pp.172–174, 2011.