

# Web に基づくプレゼンテーションのためのスライドスタックの実現

加藤 雄大<sup>†</sup> 白松 俊<sup>††</sup> 大園 忠親<sup>††</sup> 新谷 虎松<sup>††</sup>

名古屋工業大学工学部情報工学科<sup>†</sup> 名古屋工業大学大学院工学研究科情報工学専攻<sup>††</sup>

## 1 はじめに

プレゼンテーションのスライドを作成する際に主に Microsoft の PowerPoint が使用される。しかし、PowerPoint のスライドはファイルごとでまとまっているため、スライドを再利用するときは PowerPoint ファイルを開きスライドを選択する必要がある、スライドの再利用には向いていない。

本稿では、PowerPoint ファイルを HTML5 コンテンツであるスライドスタックに変換する手法について述べる。特に、PowerPoint におけるアニメーションを CSS Animation で再現するための変換および再生手法について述べる。PowerPoint ファイルを変換するには、PowerPoint ファイルが複数の XML ファイルの zip 圧縮により構成されているため、XML ファイルを解析し、スライドの構造やアニメーションを抽出する必要がある。HTML5 によるプレゼンテーションはブラウザを用いるためプレゼンテーションソフトウェアは必要なく、画像、音声、動画などの Web コンテンツを扱いやすいという利点がある。関連研究として Timesheets.js を用いたプレゼンテーションツールの研究 [1] や Carousel を利用したプレゼンテーションツールの研究 [2] が存在する。関連研究との差分は、本研究の目的をスライドの再利用としたことである。

## 2 スライドスタック

### 2.1 用語の定義

本研究におけるスライドスタックとは、複数のカードを積み重ねた HTML コンテンツである。カードは、PowerPoint におけるスライドに相当する SVG を含む Web コンテンツであり、テキスト、図形、画像などのオブジェクトを含む。

### 2.2 スライドスタックへの変換

本節では PowerPoint ファイルから SVG ファイルおよび HTML, JavaScript(以降, JS), CSS ファイルに変換するシステムについて説明する。生成される SVG ファイルはカードのファイルであり、HTML, JS, CSS はスライドショーを再生するためのファイルである。変換システムは PowerPoint ファイルを解析するモジュールと抽出された情報からファイルを変換するモジュールから構成されている。図 1 に変換システムのシステム構成図を示す。図では PowerPoint ファイルを PPTX とする。

変換の手順を以下に示す。解析モジュールは次のステップ 2, 3 の処理を行い、生成モジュールは次のステップ 4, 5, 7 の処理を行う。

1. PowerPoint ファイルをシステムに入力する
2. PowerPoint ファイルから次のスライドの XML を取得する
3. XML からアニメーションの情報を抽出する
4. XML から SVG ファイルを生成する
5. アニメーションの情報から CSS のコードを生成し SVG ファイルに追加する

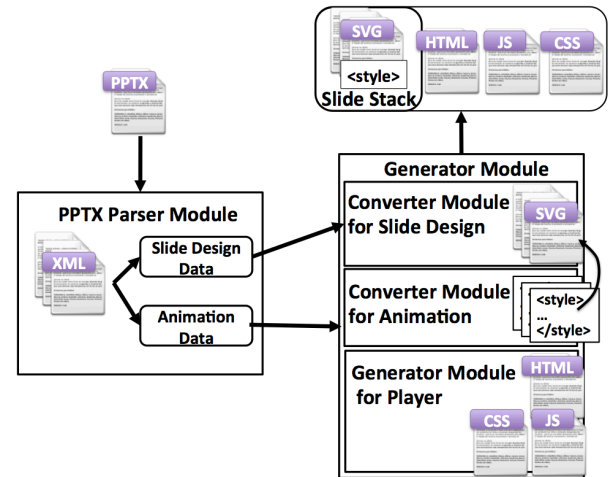


図 1: システム構成図

6. 未処理のスライドが存在する限り、2~5 を繰り返す
7. SVG ファイルと HTML, JS, CSS ファイルを出力する

ステップ 2, 4 の処理では PowerPoint ファイル解析ツールの Apache POI API を使用した。ステップ 2 で取得した XML ファイルの概略を図 2 に示し、ステップ 4 で生成された SVG ファイルの概略を図 3 に示す。スライド構造にはオブジェクトの位置や色などの情報が含まれている。図 2 のようにアニメーション情報は XML 形式で記述され、図 3 に示すように style タグ内のアニメーションは CSS Animation 形式で記述されている。CSS Animation とは CSS でアニメーションを記述する方法で、CSS Animation は 2 つの要素で構成されており、アニメーションの名前、時間などのアニメーション情報を記述したスタイルと時間ごとの動作を記述したキーフレームからなる。

```
<p:sld>
  <p:cSld> スライド構造 </p:cSld>
  <p:timing> アニメーション情報 </p:timing>
</p:sld>
```

図 2: XML ファイルの概略

```
<svg> スライド構造 </svg>
<style><![CDATA[ アニメーション ]]></style>
```

図 3: SVG ファイルの概略

ステップ 3 で抽出するアニメーション情報の具体例を図 4 に示す。XML から必要なアニメーション情報はアニメーション時間“dur”, アニメーション開始時間“delay”, 変化させる変数名“attrName”, (時間“tm”, 変数値“val”)である。図 4 の dur="500"はアニメーション時間が 500ms, delay="0"はアニメーションの開始が 0ms であることを示している。アニメーションにおいて変化させる変数は ppt.y であり、tm="0"のとき ppt.y=ppt.y(オブジェクトの y 座標), tm="100000"のとき ppt.y=1+ppt.h(オブジェクトの高さ)/2 となるように

Realizing Slide Stacks for Presentation Using a Web Web  
Yudai KATO, Shun SHIRAMATSU, Tadachika OZONO, and Toramatsu SHINTANI

<sup>†</sup>Dept. of Computer Science and Engineering, Nagoya Institute of Technology. <sup>††</sup>Dept. of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology.

ppt\_y を変化させる。ここで、#ppt\_y や 1+#ppt\_h/2 の値はスライドの高さを 1 としたときの相対的な値となる。

```
<p:anim calcmode="lin" valueType="num">
...
<p:cTn id="7" dur="500"/>...
<p:cond delay="0"/>...
<p:attrName>ppt_y</p:attrName>...
<p:tav tm="0">
<p:strVal val="#ppt_y"/>...
<p:tav tm="100000">...
<p:strVal val="1+#ppt_h/2"/>...
</p:anim>
```

図 4: XML のアニメーション情報の具体例

次にステップ 5 で生成される CSS のコードの具体例を図 5 に示す。図 5 のコードは図 4 のアニメーションを CSS Animation 形式に直して記述したコードである。“animation:”以下はアニメーションの定義名であり、keyframes1 の動作を 0ms から 500ms 間アニメーションするという定義である。linear は keyframes で定義した値が線形的に変化するということである。0% {top: 136px;}、100% {top: 554px;} はアニメーションの対象オブジェクトが 0ms で top: 136px、500ms で top: 554px となるように移動することを表す。ステップ 5

```
.animation0.build {
  animation: keyframes1 500ms linear 0ms
;
}
@keyframes keyframes1 {
  0% { top: 136px; }
  100% { top: 554px; }
}
```

図 5: CSS のアニメーションの具体例

では生成された CSS のコードを図 3 の style タグに挿入する。ステップ 7 では生成された複数の SVG ファイルとスライドショーを再生するための HTML、JS、CSS ファイルを出力する。

本システムでスライド数 3、アニメーション数 1、168KB の画像 1 枚を含む 225KB の PowerPoint ファイルを変換すると変換後の SVG ファイルは 229KB であり、再生用の HTML ファイルなどを含めて 279KB であった。

### 3 アニメーション管理

本節ではスライドのアニメーション管理について説明する。本システムではアニメーションはタグごとに適用できる。本システムでは、アニメーションするタグは class 属性に animation 名、data-animationorder プロパティ、data-animationtiming プロパティの 3 つのプロパティを持つ。animation 名は“animation”と任意の文字列を連結した文字列である。data-animationorder はアニメーションの実行の順番を表し、値は数値である。data-animationtiming はアニメーションの実行タイミングを表し、値は click, sametime, after の 3 種類がある。

次にアニメーションの順序の管理について説明する。1 つのスライドに複数のアニメーションがあった場合、それらのアニメーションには実行する順番が存在する。スライド遷移後の処理を以下に示す。スライドの遷移とは前後のスライドに移動することとする。

- class 属性に“animation”を接頭辞としてもつタグを全て取得する
- 取得したタグの DOM 要素、data-animationorder、data-animationtiming からアニメーションオブジェクトを生成する
- アニメーション配列に全てのアニメーションオブジェクトを追加する
- data-animationorder について昇べきの順になるようにアニメーション配列をソートする

ステップ b で DOM 要素、data-animationorder、data-animationtiming を要素として持つアニメーションオブジェクトを生成する。アニメーション配列とはアニメーションオブジェクトを要素としてもつ配列である。

次にアニメーションの実行管理について説明する。本システムでは右キーを押すとスライドを遷移させるか、アニメーションを実行する。右キーが押されたときの処理を以下に示す。

- アニメーションカウンタと同じ値の data-animationorder のアニメーションオブジェクトをアニメーション配列から取得する
- もし、取得したオブジェクトが存在すればステップ 3 の処理を行い、存在しなければ次のスライドに遷移する
- 取得したアニメーションオブジェクトのタグの class 属性に build を追加する
- アニメーションカウンタをインクリメントする

アニメーションカウンタの値が実行中のアニメーションの順番を表している。アニメーションの実行はタグの class 属性に build を追加することで実現している。例えば、<div class="animation0" data-animationorder="1" data-animationtiming="click"></div>というタグが存在しアニメーションカウンタの値が 1 の時、class 属性に build が追加され、図 5 の .animation0.build セレクタのアニメーションが実行される。このようにしてアニメーションの実行管理を行う。

### 4 おわりに

本稿では、PowerPoint ファイルを HTML5 コンテンツであるスライドスタックに変換する手法について述べた。特に、PowerPoint におけるアニメーションを CSS Animation で再現するための変換および再生手法について述べた。スライドの再利用性を高めるために、各スライドを独立して再生可能な SVG ファイルとして実現した。本研究の先行研究では Web 上での同期編集技術を実現しており、本技術をスライドスタックに適用することで、遠隔地でのプレゼンテーション、および、スライドの協調編集も実現可能である。さらに、Web RTC 技術を利用することで、Web ブラウザを利用したビデオチャットも実現可能であり、新たな Web ブラウザの高度利用が可能になる。また、スライドスタックに関連して、複数のスライドスタックのマッシュアップを支援するためのスライド共有基盤を開発中であり [3]、スライドスタックに基づく情報再利用および共有の促進が期待できる。

### 参考文献

- [1] F. Cazenave, V. Quint, and C. Roisin. “Timesheets.js: Tools for Web Multimedia” Proceedings of the 19th ACM international conference on Multimedia. pp.699-702. 2011.
- [2] 森光大輔, 村尾祐一. “Web 上におけるプレゼンテーション法” 情報処理学会研究報告. pp.45-52. 2009.
- [3] 井上他. “スライドスタックを用いたプレゼンテーション Web プラットフォームの開発” 情報処理学会第 75 回全国大会. 2013.