

# Webを利用したスライドスタックのための スライドリポジトリの試作

Implementing a Slide Repository for Slide Stacks by Using Web

加藤雄大 井上良太 白松俊 大園忠親 新谷虎松  
Yudai Kato Ryota Inoue Shun Shiramatsu Tadachika Ozono Toramatsu Shintani

名古屋工業大学 大学院工学研究科 情報工学専攻

Dept. of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology

We implemented a slide repository for slide stacks. A slide conversion system in the slide repository converts slides into SVG files, slide pages and presentation document pages and pushes them to the slide stacks. A SVG file represents the appearance of slide and can play some animations. However the SVG file structure is not suitable for search engine optimization. We show the slide page structure how the slide page can be easily searched.

## 1. はじめに

本研究では、既存スライドの再利用を目的としたスライドリポジトリを開発している。ここでは、再利用に適したスライドの保管方法について明らかにすることを目指している。再利用に適した既存スライドを検索することにおいて、検索語との類似度に基づく検索手法では不十分である。例えば、スライドの再利用によって得られるユーザの利益を最大化することを考えると、目的にマッチし、かつ、作成コストの高いスライドの提示が有効である場合があると考えられ、この場合は、スライドの作成コストを検索結果のランキングに反映させる必要がある。

スライドリポジトリは、Web コンテンツ化されたプレゼンテーション資料に基づくスライド管理システムである。本稿では、プレゼンテーション資料をスライドの集まりとする。スライド管理システムとは、スライドのWeb コンテンツへの変換、スライドの共有、編集機能をもつシステムである。

スライドを再利用するために、PowerPoint でスライドを共有することは可能だが、専用サーバを利用する必要がある。本研究では、スライドをWeb コンテンツ化して、SVG ファイルとして扱う。スライドをWeb コンテンツ化することでWeb 上で共有や公開することが可能である [井上 13]。

ユーザがスライドを再利用する際、再利用するスライドを探す必要がある。スライドはSVG ファイルにWeb コンテンツ化され、Web 上で公開される。しかし、SVG はスライドの外観のみを表現しているため、検索エンジンによる検索にとって適した形式ではない。SVG ファイルをWeb 上に公開しても、検索でヒットしない可能性があるという課題がある。

スライドリポジトリは、PowerPoint ファイルを入力として受け取り、SVG ファイル、スライドページとプレゼンテーション資料ページに変換し、それらをスライドスタックに蓄積する。SVG ファイルは、元スライドと同じ外観を再現したファイルであり、PowerPoint のアニメーションの一部を再生することが可能である。スライドページは、SVG ファイルと別に、スライドの内容を表すWeb ページであり、プレゼンテーション資料ページは、1つのプレゼンテーション資料を表すWeb ペ

ジである。スライドスタックとは、カードを蓄積したものである。カードとは、HTML ファイル、画像、テキスト、PDF、Web コンテンツ化されたスライドなどのWeb コンテンツである。本研究ではプレゼンテーション資料としてPowerPoint ファイルを対象とするが、他の研究ではOpen Office Impressのプレゼンテーション資料を対象とする例もある [Fabien 11]。

本稿では、PowerPoint ファイルからSVG ファイル、スライドページとプレゼンテーション資料ページの変換について説明する。

## 2. スライドのWeb コンテンツ化

スライドリポジトリでは、PowerPoint ファイル形式のプレゼンテーション資料をWeb コンテンツ化して管理する。プレゼンテーション資料をWeb コンテンツ化することでいくつかの利点が得られる。

- プレゼンテーション資料の公開が可能である。例えば、企業または研究室で新規にプレゼンテーション資料を作成する場合、既存のプレゼンテーション資料を再利用することができれば、資料作成コストを少なくすることが可能である。他に、プレゼンテーション資料が公開されていれば、企業、研究室以外でも関連するプレゼンテーション資料を参考にすることができる。
- カードはテキスト形式で記述されているため、カードからテキストを抽出し、検索のためのインデックス作成に利用するのが容易である。
- 公開されているカードを閲覧するためにブラウザを用いればよく、カード閲覧用のソフトウェアを使用する必要がない。例えば、Microsoft SharePoint Server を使用すれば、PowerPoint ファイルの共有、公開をすることが可能であるが、専用サーバを設置する必要がある。Web コンテンツ化されたスライドの閲覧に関して、Web ページのリストをブラウザ上でCarouselを用いてスライドショーをする研究もされている [森光 09]。
- カードに画像、音声、動画などの様々なWeb コンテンツを貼付ける事が可能である。例えば、カードに画像を貼付ける場合に<img>を、音声を貼付ける場合に<audio>を、動画を貼付ける場合に<video>をカードに追加することで貼付けることが可能である。

連絡先: 加藤雄大, 名古屋工業大学 大学院工学研究科 情報工学専攻, 〒466-8555 愛知県名古屋市昭和区御器所町, Tel:052-733-6550, Fax:052-735-5584, E-Mail:kyudai@toralab.org

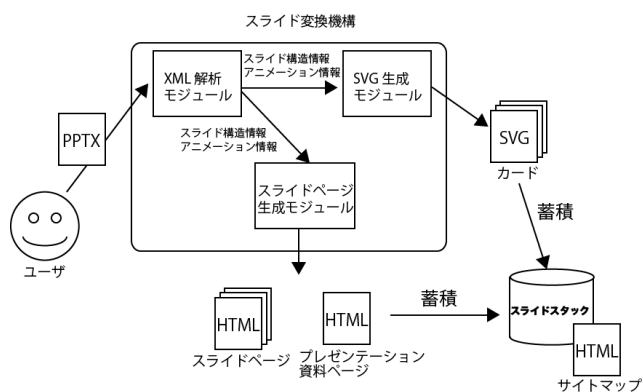


図 1: スライドリポジトリのシステム構成

以上の理由でスライドの Web コンテンツ化は必要であると考える。

本研究では、Web コンテンツ化されたスライドとして SVG ファイルを用いる。SVG の利点として、スライド内にはテキスト、図形、画像などのオブジェクトが含まれており、SVG はタグでこれらのオブジェクトを記述することが可能である。SVG はオブジェクトの位置、回転、拡大縮小などの操作も可能である。SVG はアニメーションをする機能を持ち、`<animate>` を使用してアニメーションを定義することが可能である。

### 3. スライドリポジトリ

スライドリポジトリのシステム構成図を図 1 に示す。スライドリポジトリは、スライド変換機構をもつ。

スライド変換機構は、PowerPoint ファイルのスライドから SVG ファイルへの変換と、検索エンジンによる検索のためのスライドページとプレゼンテーション資料ページを生成する。1つの PowerPoint ファイルから生成される SVG ファイルとスライドページはスライドの枚数分である。

スライドスタック内には、サイトマップ、スライドページ、プレゼンテーション資料ページ、SVG ファイルが蓄積されている。サイトマップは、プレゼンテーション資料ページへのリンクをもつ Web ページである。PowerPoint ファイルの変換が行われるたびに、カード、プレゼンテーション資料ページ、スライドページが生成され、スライドスタックに蓄積されていく。

スライド変換機構は、図 1 に示すように、XML 解析モジュール、SVG 生成モジュール、スライドページ生成モジュールで構成されている。XML 解析モジュールと SVG 生成モジュールの一部には Microsoft ドキュメントのための Java API である、Apache POI を利用した、XML 解析モジュールは、XML ファイルからスライド構造とアニメーションの情報を抽出し、SVG 生成モジュールに渡す。SVG 生成モジュールは、SVG ファイルを生成し、抽出されたスライド構造とアニメーションの情報から SVG 形式でファイルに書き込む。スライドページ生成モジュールは、XML 解析モジュールから抽出されたスライド構造とアニメーションの情報から、HTML 形式のスライドページを生成する。

#### 3.1 XML 解析モジュール

XML モジュールについて説明する。XML 解析モジュールは、PowerPoint ファイルから XML ファイルを取得し、XML ファイルからスライド構造とアニメーションの情報を抽出す

る。PowerPoint ファイルは複数の XML ファイルとテーマ、レイアウトなどのファイルが圧縮されて構成されている。1枚のスライドは 1つの XML ファイルで記述され、その XML ファイルはスライドの構造、アニメーション、画面遷移の情報を持つ。XML ファイルの内容を図 2 に示す。p:sld タグ内の p:cSld タグ内にスライド構造、p:transition タグ内に画面遷移、p:timing タグ内にアニメーションについて XML で記述されている。スライド構造には、オブジェクトと呼ぶ、テキスト、図形、画像などが含まれ、オブジェクトの位置、大きさ、色などが定義されている。

```
<p:sld>
  <p:cSld>スライド構造</p:cSld>
  <p:transition>画面遷移</p:transition>
  <p:timing>アニメーション</p:timing>
</p:sld>
```

図 2: XML ファイルの概略

XML 解析モジュールは、XML ファイルの p:cSld から位置、種類などのオブジェクトの情報を抽出し、p:timing タグからアニメーションの情報を抽出する。

本稿では、(アニメーション対象のオブジェクト ID)、(変化させる変数)、(開始タイミング)、(開始時間)、(時間)、(キーフレーム)をまとめて動作情報と呼ぶ。動作情報はオブジェクトの属性の値の変化を定義している。例えば、オブジェクトの x 座標を変化させることやオブジェクトを非表示から表示に変えることがある。一般的に、p:timing タグ内には複数の動作情報が存在しており、その複数の動作情報をまとめてアニメーション情報と呼ぶ。次に、抽出したスライド構造とアニメーション情報を SVG 生成モジュールに渡す。

動作は大別して、表示、移動、エフェクト、拡大縮小、回転、色、モーションに分けられる。PowerPoint のアニメーションにおける動作の割合を図 3 に示す。本研究で対応した動作は表示と移動の動作であり、グラフにおいて 66% を占めている。

具体例を用いて、アニメーション情報について説明する。図 4 に XML 形式のバウンドというアニメーションの一部を示し、タグと属性の説明を表 1 に示す。図 4 の p:cTn タグ内に p:set タグ、p:anim タグが存在し、それぞれのタグは 1つの動作を定義している。p:cTn タグの nodeType 属性"clickEffect"はアニメーションをクリック時に開始することを定義している。p:set タグは開始 0s から 1ms の間にアニメーション対象のオブジェクトを表示させる動作である。p:anim タグはアニメーション対象のオブジェクトの y 座標の値を、`#ppt_y-sin(pi*$/3`

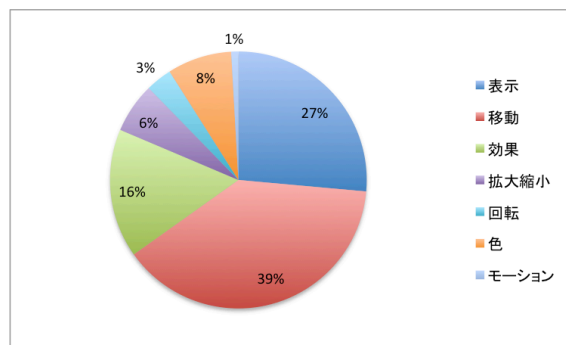


図 3: アニメーションにおける動作の割合

表 1: タグと属性の説明

タグまたは属性	説明
p:cTn の nodeType	アニメーションの開始トリガー
p:spTgt の spid	アニメーションするオブジェクトの ID
p:cTn の dur	アニメーションの時間. 単位は ms
p:cond の delay	アニメーションの開始時間. 単位は ms
p:attrName	値を変化させる属性
p:tav	キーフレーム

にする動作である. この数式の変数 $\$$ は 0s から 500ms の間に, 0.5 から 1 に変化する.

XML 解析モジュールは, p:set タグ, p:anim タグのそれぞれから 1 つの動作情報として抽出し, アニメーション情報にまとめる.

```
<p:cTn ...nodeType="clickEffect">
  ...
  <p:set>
    <p:cTn id="6" dur="1" fill="hold">
      <p:cond delay="0"/>
    </p:cTn>
    <p:spTgt spid="3"></p:spTgt>
    <p:attrName>style.visibility</p:attrName>
    <p:to>
      <p:strVal val="visible"/>
    </p:to>
  </p:set>
  <p:anim>
    <p:cTn id="8" dur="500" fill="hold"/>
    <p:attrName>ppt_y</p:attrName>
    <p:tavLst>
      <p:tav tm="0" fmla="#ppt_y-sin(pi*$/3">
        <p:val>
          <p:fltVal val="0.5"/>
        ...
      <p:tav tm="100000">
        <p:val>
          <p:fltVal val="1"/>
        ...
      <p:animRot by="2160000">
        <p:attrNameLst>
          <p:attrName>r</p:attrName>
        </p:attrNameLst>
      ...
    </p:cTn>
```

図 4: バウンドアニメーションの XML 形式の一部

### 3.2 SVG 生成モジュール

SVG 生成モジュールについて説明する. SVG 生成モジュールは XML 解析モジュールから渡されるスライド構造とアニメーション情報に基づいて SVG ファイルを生成する. スライド構造の情報から同じ外観となるような SVG 形式のテキストを生成する.

スライド構造から SVG 形式のテキストへの変換には Apache POI を使用した. 図 5 にスライド構造から変換された SVG 形式のテキストの記述例を示す.

```
<g>
  <text x="263" y="243">Slide Title</text>
</g>
<g>
  <text x="287" y="395">SubTitle</text>
</g>
```

図 5: スライド構造の SVG 形式の記述例

次に, アニメーション情報から SVG 形式のテキストへの変換について具体例を用いて説明する. 図 6 にバウンドアニメーションの XML 形式を SVG 形式に変換したものを示している. 1 つ目の animate タグは, id が target2 の SVG オブジェクトの visibility 属性を, 1ms で hidden から visible に変化させる. 2 つ目の animate タグは, id が animation0 の animate タグで定義されているアニメーションと同時に開始され, id が target2 の SVG オブジェクトの y 座標の値を, 500ms で keyTimes に指定された時間に values で指定した値となるように変化させる.

アニメーション情報から SVG アニメーションテキストの生成について説明する. SVG 生成モジュールはアニメーション情報内の動作情報 1 つに対して, 1 つの animate タグを生成する. animate タグの xlink:href 属性の値を "#target(アニメーション対象のオブジェクト ID)" とする. attributeName 属性は (変化させる変数) が style.visibility ならば visibility に, ppt\_x ならば x などにする. dur 属性は (アニメーション時間)ms とする. PowerPoint ファイルのキーフレーム処理は動的に値を変化させられるが, SVG は静的に値を決定しなければならない. したがって, 図 4 のバウンドアニメーションのキーフレームに対応する SVG のキーフレームを生成するにはあらかじめ値を決定しておく必要がある. そこで, キーフレームの時間をある一定間隔でとり, その時間に対する式  $\#ppt\_y - \sin(\pi * \$) / 3$  の値を計算する. 複数の時刻に対する値を計算し, keyTimes と values 属性を追加する.

```
<animate id="animation0" xlink:href="#target2"
  attributeName="visibility" begin="
  indefinite" dur="1ms" keyTimes="0;1"
  values="hidden;visible"/>
<animate xlink:href="#target2" attributeName="
  y" begin="animation0.begin+0ms" dur="500ms
  " keyTimes="0.0;0.07;0.2;0.467;1.0" values
  ="-180;-178;-171;-133;0"/>
```

図 6: バウンドアニメーションの SVG 形式の一部

### 3.3 スライドページ生成モジュール

スライドページ生成モジュールは, スライド構造とアニメーションの情報から, HTML 形式のスライドページとプレゼンテーション資料ページを生成し, スライドスタックに蓄積させていく.

まず, サイトマップ, プレゼンテーション資料ページ, スライドページの構造について説明する. スライドスタック内に存在するサイトマップとプレゼンテーション資料ページ, スライドページの構造を図 7 に示す. 矢印はページからページへのリンクを表している.

サイトマップは, 全てのプレゼンテーション資料ページへのリンクをもつページである. アンカーテキストはそれぞれのプレゼンテーション資料のタイトルである.

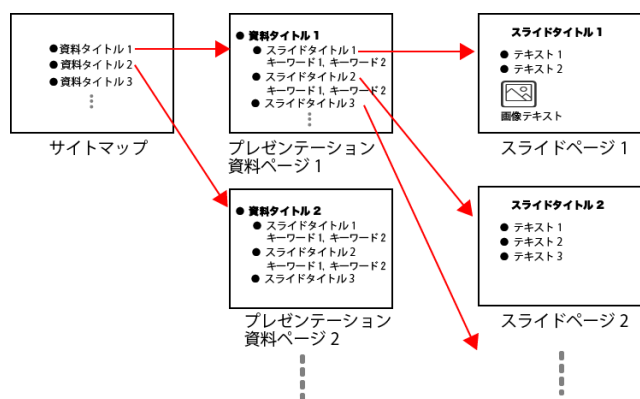


図 7: サイトマップ, プレゼンテーション資料ページ, スライドページの構造

プレゼンテーション資料ページは、スライドページへのリンクをもつページである。SEO 対策のために、プレゼンテーション資料ページの<title>内をプレゼンテーション資料のタイトルとしている。そして、箇条書き形式でスライドページへのリンクをもつ。また、リンクの下部にスライドページから抽出したスライドの重要語を付け加えている。これにより、プレゼンテーション資料ページがどのような内容のプレゼンテーション資料なのか示すことができる。

スライドページは、SVG ファイルへの 1 つのリンクをもつ。スライドページが検索されれば、スライドページから SVG ファイルに移動することが可能となる。ほとんどのスライドのレイアウトは、“タイトルとコンテンツ”というレイアウトが使用されている。タイトルは、スライドの内容を表す最も重要なテキストであるため、スライドページの<title>内のテキストとし、スライドページの上部にタイトルを配置する。また、コンテンツが画像ではなくテキストである場合、テキストは箇条書きで記述されるため、箇条書きの内容を<ul><li>を用いて、タイトルの下に配置する。

次に、スライドページ生成モジュールについて説明する。スライドページ生成モジュールは、スライドページ、プレゼンテーション資料ページの順で生成する。初めに、スライドページ生成について説明する。最初にスライドページとなる HTML ファイルを生成する。この HTML ファイルは、<title>, <body>内のテキストが存在しないファイルである。スライドのレイアウトが“タイトルスライド”ならば中央のタイトル，“タイトルとコンテンツ”ならばタイトル部分を抽出し、HTML ファイルの<title>内に追加する。スライドのレイアウトが“タイトルとコンテンツ”であり、かつコンテンツが箇条書きで記述されているならば、<ul><li>を用いて箇条書き形式にし、HTML に追加する。また、スライド内に箇条書きとは別に独立したテキストがあるならば、そのテキストを<div>でくくり HTML ファイルに追加する。文字が大きい、色が他の文字と違う、アニメーションが付加されているテキストは、強調されているため HTML ファイルでも強調する必要がある。そのため、スライドのテキストを HTML ファイルに追加する際に、強調されているテキストを<strong>でくくる。例えば、“これは<strong>重要語</strong>である”とする。スライドに画像が含まれるならば、HTML に<img>で画像を追加する。ここで、画像の内容をテキストで記述することが望ましく、画像を表すテキストは画像内や画像周辺に存在する 경우가多い。よっ

て、画像内からテキストを抽出、または、画像周辺の独立したテキストを抽出し、<img>の alt 属性の値に設定する。例えば、とする。スライド内にアニメーションが存在するならば、HTML ファイルに、<div>～アニメーション</div>を追加する。例えば、スライド内にスライドインのアニメーションが含まれていれば、アニメーションの動作を解析し、動作を表す文字列を調べ、<div>上に行くアニメーション<div>というように追加する。最後に、スライドに対応する SVG ファイルへのリンクを HTML ファイルに追加する。

次に、プレゼンテーション資料ページ生成について説明する。スライドページ生成時と同様に、HTML ファイルを生成する。PowerPoint ファイルの 1 枚目のスライドからプレゼンテーション資料のタイトルを抽出し、HTML ファイルの<title>内のテキストとする。また、タイトルのテキストをプレゼンテーション資料ページの上部に配置する。

次に、スライドページへのリンクを<ul><li>で箇条書きの形式で追加する。アンカーテキストをそれぞれのスライドのタイトルとして、スライドページへのリンクを追加する。スライドページからコサイン尺度などを用いて、スライド内のいくつかの重要語を抽出し、アンカーテキストの下部に配置する。

プレゼンテーション資料ページの生成されてから、サイトマップからプレゼンテーション資料ページへのリンクの更新を行う。サイトマップ内の箇条書きの部分に、新しく生成されたプレゼンテーション資料ページへのリンクを追加する。

## 4. おわりに

本稿では、スライドリポジトリと呼ぶ、Web コンテンツ化されたプレゼンテーション資料に基づくスライド管理システムを試作した。スライドのコンテンツ化として、PowerPoint ファイルからアニメーション機能を有する SVG ファイルに変換した。PowerPoint のアニメーションの中で、移動、表示の動作に対応した。しかし、スライドを Web コンテンツ化したカードは SVG ファイルであり、SVG ファイルはスライドの外観のみを表現しているため、スライドの検索に対して適した形式ではないという課題があった。本稿では、SVG ファイルとは別に、検索エンジンによる検索のためのスライドページを生成した。例えば、<title>内のテキストをスライドのタイトルにすることや、強調されているテキストを<strong>でくくることで、スライドの内容を表し、かつ検索されやすいスライドページを生成した。

## 参考文献

- [井上 13] 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.
- [Fabien 11] Fabien Cazenave, Vincent Quint, and Cecile Roisin. “Timesheets.js: Tools for Web Multimedia” Proceedings of the 19th ACM international conference on Multimedia. pp.699-702. 2011.
- [森光 09] 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.