

Web 上におけるプレゼンテーション法

森光大輔 村尾裕一

電気通信大学 情報工学科

プレゼンテーションでは予め用意された内容を用いるのが一般的である。しかし、情報源の多様化に伴いそのような静的な情報だけではプレゼンテーションは不十分な場合がある。そこで我々は、多様な情報をプレゼンテーションに活かす試みとして Web ブラウザ上でのプレゼンテーション法を提案する。そのためのツールとして、Web ページのリストをブラウザ上で Carousel アニメーションを用いたスライドショーで提示するプログラムを JavaScript で実装した。その特徴は、予め用意された html の記述だけでなく、外部ページ自体を直接かつ一元的に利用可能とすることである。結果、一般的なプレゼンテーションに、デモ等の動的な内容も含めた、リアルタイムに発信される情報を容易に組み込み一元的に利用するためのツールを実現することができた。

A Method of Presentation on Web

MORIMITSU Daisuke, MURAO Hirokazu

Department of Computer Science, The University of Electro-Communications

We usually prepare all the contents of slides used for presentation in advance. However, with the diversification of information resources on the Web, presentation may be unsatisfactory only with such still prepared information. We propose a method to make presentaion on Web browsers, that will enables us to use the information spread over W^3 cooperatively. To examine its practical significance and realize, we implemented a presentation tool for this purpose in JavaScript, which performs slide-show on a list of Web pages in a carousel style of animation. Our method is characterized by the integrated use of various Web pages, regardless of being still or alive, and the tool enables us to incorporate various kinds of Web pages in presentation material as scenes of slide-show, such as pages of local HTML files, pages transmitted in real time via internet including Wiki and blog, and pages of program demonstration. Our experience shows such a browser tool can not only provide a sufficient ability for presentation but change the quality of presentation through live contents.

1. はじめに

近年 Web 上の技術発展は進み、個人レベルにおいても、Wiki や blog といったコミュニケーションツールを用いて情報を提供するようになってきている。Web 上の情報を講義などプレゼンテーションにおいて紹介するケースもある。その一方で、Web 上にはあまりにも多くの情報が提示されており、そのため、ある事柄についてそういった Web 上の情報を紹介する際に、紹介者が上手に纏めることができず理解しづらくなったり、また情報サイトを複数ウィンドウで開いた結果、ウィンドウ操作の為紹介に手間取ったりする場面が見受けられる。これは情報を紹介する際、複数のウィンドウを表示させることにより、比較したい情報が別のウィンドウに隠されるなどの理由で、情報の確認に支障がでるためであり、このような情報紹介方法では情報紹介者の狙いを達成することは困難である。

一方、現在プレゼンテーションにおいて Microsoft PowerPoint 等が広く用いられる。PowerPoint は発表内容はすべて事前に準備する必要があり、リアルタイムに発信されている情報を利用することができない。そこで、PowerPoint のようなスライドショーの機能と、ひとつの情報を提示する Web ページを一枚のスライド画面に対応させ、複数のページの切り替えをスライドの切り替えのように簡単に行える機能を組み合わせブラウザ上で実装し、新たなプレゼンテーション法を提案する。また、ひとつの Web ページを一枚のスライド画面に対応させることで、CGI などを Web 上で動作するスクリプトのデモンストレーションさえもスライ

ド中に実演することが可能になり、プレゼンテーションの枠が広げることが可能になる。今回、以上の考えのもと jSpow(jQuery Slideshow Presentation On Web) というツールを作成した。jSpow では、指定された HTML 要素中に含まれる特定の要素を一枚のスライドの記述とみなし、そのリストに対してスライドショーのアニメーションを行う。スライドショーに付随して必要となる細かな機能も自動的に付加して利便性を上げる。特に、ひとつの Web ページを一枚のスライドに対応させているため、リアルタイムに発信されている情報を含めることが可能なばかりか、Web 上でのデモなどの動的な内容を含むコンテンツも利用することが可能となる。

現在、Web 上におけるプレゼンテーション法としては、jPresen など幾つかのツールが存在する。表示する内容は、PowerPoint と同様に予め (HTML により記述し) 用意する必要がある。類似のものに Slidy, S5, S6 等があるが同様である。一方、JavaScript で記述されている既存のスライドショーツールはアニメーションに特化しているものである。アニメーションの対象指定はすべて設置者が細かく設定するようになっている。我々の jSpow は、この両端の手法をひとつにまとめる機構を用意し、現存する Web ページも活用することができるようにするためのツールである。

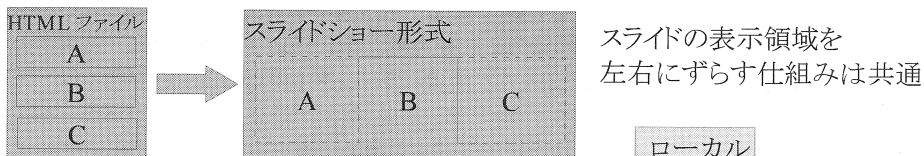
本稿では、この jSpow の実現法と記述法を説明し、併せて、我々の目指すところを明らかにする。次の 2 節では設計方針を述べ、利用する既存技術や基本技術について説明する。3 節では、実際に開発した内容を、主に利用者の観点からどのようなものであるかがわかるように説明する。最後の 4 節では、最近の研究開発との類似点や相違点に触れながら、今後の展望を検討する。

2. 設計方針と基本技術

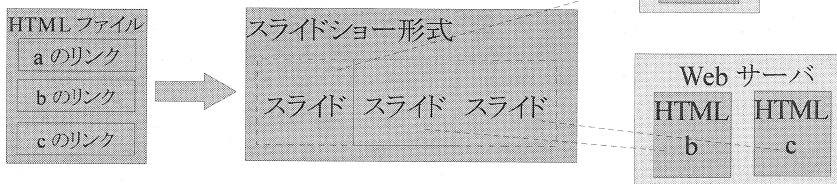
2.1 スライドデータの扱い

既存のプレゼンテーションツールでは特定のタグで 1 ページずつ記述するのが通例である。jPresen は JavaScript のライブラリである jQuery[1] を用いて、HTML の div タグで括られたエリアを 1 ページとして PowerPoint のようなスライドを提供する。他にも、PowerPoint のファイルを HTML に変換して Web 上に提示しページやスライドの内容は予め用意するという静的なものである。今回提案する方法は、動作中の Web ページを含めた様々な HTML ファイルをスライド形式として表示させ、動的な機構も活用可能とするものである。

既存する手法



今回提案する手法



2.2 注目した Web 技術

Web 上のツールの開発には色々な言語の利用(と形態)が考えられるが、我々のツールは様々な Web ページをブラウザ上で扱うことが主体となる為、ブラウザ上でスクリプト言語 JavaScript により開発する。JavaScript は、HTML の動的書き換え、スタイルの変更、情報通信の制御、入力フォームの自動補完など、Web ページの使用感向上を目的として利用されることが多い。代表的な応用例として GoogleMaps 等があり、近年、非同期通信を利用した Ajax などの開発が注目されている。また最近では、Mozilla Firefox の機能拡張のためのアドオンの記述に用いられるだけでなく、そのアドオンの一例である Greasemonkey(機能追加の汎用的な方法を与える)や Opera ブラウザでは利用者が用意したスクリプト(「User JavaScript」と呼ばれる)でブラウザや表示のカスタマイズを可能としている。

JavaScript によるライブラリも各種開発されている。ここでは jQuery [1]に注目する。その大きな特徴として処理が非常に軽いことが挙げられる。記述方法は JavaScript のライブラリとして有名な prototype.js と似るが、仕様は独自のものである。CSS やアニメーション操作等の機能が多く実装されており、それらを利用した多くの sample プラグインが開発されている。また、クロスブラウザ対策もある程度は行われている。詳細については公式ページ上のドキュメントを参照。

アニメーションには jCarouselLite[2]を利用する。jCarouselLite とは jQuery のプラグインのひとつで、一種の画像のスライドショーを目的として作られたものである。動作は、Carousel の名の通り、回転木馬を横から見たように画像が回転していく。クロスブラウザ対策は行われていない。今回、この画像のスライドショーの動作と PowerPoint の動作に注目した。jCarouselLite の動作においてスライドさせる対象を画像から HTML ファイルに変更することで、目的とするような機能が実現できるのではないかと考え、本研究の着想に至った。

2.3 基本設計方針

今回、次の4点を重視して開発を行うこととした。

- JavaScript の知識がない人でも扱えること、
- 簡単な初期設定だけで利用可能とすること、
- Web 上に公開されているページも表示対象として利用可能にすること、
- クロスブラウザ対策を行う。

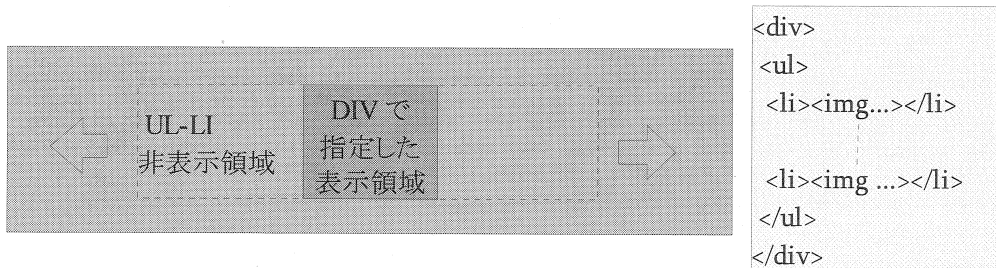
設計のベースとして、上述の jCarouselLite のスライドショーのアニメーション機能を利用し HTML ファイル自体をスライドさせるという動作を考えた。ここで、操作の対象は HTML ファイルだが、HTML に関して詳しい知識がない人でも直感的に使えるよう、利用するタグはページのリンクとしての「a(アンカー)」もしくは該当リンクを直接ページに表示する「iframe(インラインフレーム)」に限定して考える。利用者の負担を少なくするために HTML の記述を簡略化することと、スライドアニメーション時のページ遷移をスムーズにすることを目的として、「iframe」を用いることとした。iframe で指定された HTML ファイルは、その記述を含むファイルのロード時に読み込まれるため、最初にすべての HTML ファイルが読み込まれた結果、ページ遷移時に処理が重くなったり通信のために処理が中断することが避けられる。また、Ajax を用いて HTML ファイルを非同期に読み込む方法も考えられるが、同様の理由により考慮から外した。

他に、スライドに対する各種機能を自動的に付与することが望まれる。これは利用者が HTML の記述などの知識が乏しい場合でも、ページ番号の付与等あって当然といった機能を提供する。最後のクロスブラウザ対策では、実際設計していく過程でブラウザに拠る様々な動作や装飾の違いが現れてくることが考え

られるが、その都度適切な対策を行っていくこととする。

2.4 Carousel アニメーションの構造

Carousel アニメーションの機能は `jCarouselLite` に基づいて開発を行う。その実現法を説明する。スライドショーの構成は下図のとおりである(アニメーション方向が水平の場合)。



スライドショーに変換する対象の全体は `div` タグで括られ、その階層下に対象物のリストを表す `ul` タグ `li` タグが置かれる。`li` タグの各項目をスライド一枚に対応させ、`li` タグの項目リストを、通常の表示は行わずに内部的に `ul` タグ内に横に連続して並べる。各項目の `li` タグのスタイルの位置情報を書き換えることにより項目全体は左右にスライドするが、`div` タグで指定した表示領域内に位置づけられた項目だけが表示され、それ以外は DOM ツリーオブジェクトとして内部では存在するが表示はされない。例えば、`div` タグが映写機で `ul-li` タグがフィルムのような役割になっている。また、スライド方向は横向きだけでなく縦向きにも行うことができる。変換対象の領域に関しては `div` タグの ID やクラスを利用し JavaScript で対象を指定し、スライドショーに変換する形式とする。

アニメーションは、専用のアニメーション関数で構成される。あらかじめ1スライド当りのサイズを保存しておき、指定されたアニメーション方向に対して、`li` タグのフォーカスの変更とスライドの移動のアニメーションを表示する。アニメーションは Carousel (回転木馬) 式であるためスライドをループして表示することもできるが、スライドショーにおいてループは不要なのでここでは用いない。特に、`jCarouselLite` におけるループは項目リストを2回繰り返すことで擬似的なループを実現しており、そのまま我々の用途に用いると HTML ファイルの読み込みが倍増するので適さない。

3. 開発

3.1 最小限の機能

先ず、当初の目的を実現する最小限の機能として、Carousel スライドショーを行う機構の開発を行った。`jCarouselLite` 本来の機能に加え、次の機能を実装した。

(ア) スライドと `iframe` タグとの対応付け

(イ) `jCarouselLite` が提供するいくつかの機能を利用するかの `on/off` による指定

(on 指定時には機能の自動追加)

(ウ) キーボードによるスライドの操作機能の追加

Carousel アニメーションによるスライドショーを行う部分の記述法については §2.4 で説明したが、(ア) では `iframe` で指定された Web ページをひとつの項目としてリストの処理を行う。つまり、実際の HTML の記

述では `iframe` によるスライドを複数用意し、変換する範囲を `div` タグで括り JavaScript によって指定する。これはページ全体をスライドショーに変えてしまうわけではなく、ページの一部としてスライドショーを組み込む為である。

(イ)は、JavaScript の知識がない利用者を想定し、各種の機能設定を簡単な操作で行えるようにすることが目的である。具体的には、スライドアニメーションの実行を振り分けたボタンの自動設置機能を用意した場合、`on(true)`を指定することで自動的に設置されるといった仕組みが用意されている。

(ウ)は一般的なスライドショーと同様の簡単なキーボード操作を導入することにより、違和感なく操作できるようにすることが目的である。既存の機能としては、スライドのヘッダー部分やページ自体をクリックすることでスライドしたり、プルダウン式のメニューによってページを選ぶといった方法がある。また、「S5」、「S6」といったプレゼンテーションツールのように、キーボード操作を導入している例もある。キーボード操作は Web ブラウザ上での機能の衝突も考えられるので、今回はカーソル Key と EnterKey や SpaceKey に対してのみ割り当てをすることとした。

以上により最低限の機能は実装される。しかし、これだけではスライドショーの機能としては実用上は不十分なので拡張機能の追加を行った。その拡張については次節で述べる。

3.2 拡張機能

拡張機能として実装した内容を簡単に紹介する。

1. `iframe` 以外のタグによるスライド表現の導入
2. メニュー表示機能
3. スライドアニメーションの操作に関する機能
4. 画面の分割表示機能
5. 目次表示機能
6. 背景色の変更機能

1.は、一枚のスライドに変換する対象を拡張し、`img` タグを用いた画像による表現と `span` タグを用いた HTML の直接記述による表現を用意したことである。`img` タグによる画像の表現は HTML が書けない人がスライドを画像で代替する利用法である。`span` タグによる直接記述による表現は HTML を書ける人向けの方式である。前述した `iframe` だけのスライドでは、既存の Web ページ以外に利用者独自のスライドを組み込むには専用の HTML ファイルを個別に用意せねばならず面倒だが、`span` タグで直接記述すればこの点は解消される。より詳しくは後述する。簡単な HTML 記述形式の例を以下に示す。

```
<div>
<iframe src="..."></iframe>

<span class="slide">...</span>
</div>
```

実際の記述形式



```
<div>
<ul>
<li><iframe src="..."></iframe></li>
<li></li>
<li><span class="slide">...</span></li>
</ul>
</div>
```

jSpow 内部での形式

2.のメニュー表示機能は、スライドタイトルの一覧表示や、特定のスライドまでジャンプする為の機能である。jSpow を用いて通常の Web ページをスライドアニメーションで表現するという使い方も考慮した機能で

もある。メニューのタイトルはスライド一枚の指定を HTML で記述している際に、title 属性値として設定したものを取得し利用している。メニューは数種類用意しており、好みのメニュータイプを選んで設置する。一部のメニューはブラウザ毎に表示サイズが異なってしまうが、ブラウザ毎に表示するサイズを変更することで解決している。将来の二次元方向へのアニメーションの実装を想定した場合にメニューをどのように表示するかを考える必要があるが、このことに関連して、現状 Internet Explorer では別の箇所が再レンダリングされない限り変更が反映されないという問題と z-index を調整してもスライド表示領域の下に隠れてしまうという問題が発覚している。この問題を回避するために、利用できないブラウザにおいては似たタイプのメニューに自動的に置き換えている。

3.では、設計方針で述べたキーボードによるアニメーション操作に加え、スライドアニメーションを行わせるボタンを自動的にページに埋め込む機能も用意した。どちらも利用せずにメニューだけをを用いてページ遷移を行わせるという使い方も考えられるが、設置者が適宜機能を選別して使えばよい。キーボードへの割り当てに関しては、左右のカーソル Key と EnterKey、SpaceKey、BackspaceKey をそれぞれページの遷移に割り当てている。EndKey には PowerPoint では最後のページに移動する機能が割り当てられているが、jSpow ではブラウザの元々の機能としてページの一番下まで移動する機能が割り振られていて重複する為割り振っていない。

4.の画面の分割表示機能は、今回外部の動的な Web ページも参照できるということで複数ページを見比べてみる、といった使い道も考えられる。その為、span タグで解説したスライド内に二つのページを表示させる画面分割表示の機能を用意した。また、同時に次のページのサムネイルのように表示する機能や、タイトルのみ表示させる機能を用意した。ページのタイトルの取得に関しては、メニューと同様の形式で取得している。

5.の目次表示機能は、目次としてスライド全てをサムネイル形式で表示する機能である。span タグで表現されたスライドに関しては、表示サイズを一律に小さく指定することでサムネイル画像のように表現することによりうまく表示することができた。iframe を利用した外部ページに関しては、サムネイル作成 API を利用してサムネイル画像を表示するという仕組みとした。外部サイトに関しては利用者が好みのサイトを利用することができる。また、サムネイルをクリックすることで該当ページにスライドアニメーションを行う。

6.は、設置者の好みでデザインを変える機能の一環であり、背景色の変更を可能にする。現在変更できる範囲はスライドの背景とメニューの背景に限られる。

1.の span タグによる直接記述による表現について、分割表示機能との関連も含めて、より詳しく説明する。span タグは本来括った範囲を装飾する目的で利用するタグであるが、我々は括られた範囲の記述を一枚のスライドに対応付けるという目的で用いる。この方法は、Web 上のプレゼンテーションツールでよく用いられる。本来の目的で span タグを利用した場合入れ子構造となるが、見分けをつけるために一枚のスライドに対応する span タグにはクラスとして「slide」を指定することとした。また、現状 img タグと iframe タグは自動的に1枚のスライドに変換することとしているが、この span タグの内部にこれらのタグの記述が存在する場合(一枚のスライド中に複数の Web ページを表示する場合)には誤った処理をしかねない。これを回避するために、そのような場合は内部の img タグや iframe タグには「notslide」クラスを指定することとした。これは記述を煩わしくするというデメリットがあるが、そのような記述は見栄えを悪くすることが多いので多用すべきでないという我々の考えを反映している。今回のスライドショーの表示サイズが Web ブラウザの表示サイズに左右されるよう設計した為、ひとつのスライド中に様々な画像などを詰め込むと非常に見栄えが悪くなるからである。複数ページの同時表示は2ページに限定し、分割表示の機能を用いるべきである。

分割表示の機能では、2つのスライド領域を「twin」クラスを付与した span タグで括ることとし、各スライドをひとつのスライド上に幅を半分にして左右に独立して表示する。ここで、twin クラスの span タグの内側では、一枚のスライド記述に対応する span タグには「slide」クラスを付与する必要はない。一方、内側での

指定に `img` や `iframe` タグを指定すると、元々のスライド単体として使う場合と同一の記述になってしまう問題点がでてきた。これを解決するために、「`twin`」クラスを親に持つ `img,iframe` タグには自動的に「`notslide`」クラスを自動的に付与することとした(利用者が指定する必要はない)。この画面の二分割して表示する機能の用途として、ふたつのページを見比べたり、片方でデモンストレーションを行うといった利用形態が考えられる。以下に `span` タグによる記述例を示す。

```
<span class="twin">
<span>...</span>
<iframe src="..."></iframe>
</span>
```

画面の二分割の記述例

```
<span class="slide">
...<span style="font-color:red">...</span>...

</span>
```

`span` タグ内でスライド変換対象を利用する記述例

`iframe` による表現では、指定された Web ページ内で記述された JavaScript の動作と本ツールとの干渉が問題となる。Firefox のアドオンとしてユーザ定義の機能追加を行う Greasemonkey については、正しく動作することを確認しており、Greasemonkey の動作解説のデモといった動的な内容を含めることも可能である。もちろん通常の Web 上のスクリプトデモ等も行うことも可能である。ただし、ページによっては自動的にページの中の入力フォームにフォーカスをあてる JavaScript 等が組み込まれている場合がある。この場合最初のスライドの構成の後に `iframe` 内でページのレンダリングが行われ JavaScript が実行される為に、フォーカスが自動的に対象ページにずらされてしまい、表示が乱れてしまう。Yahoo や Google のような検索ページにおいては自動フォーカスが組み込まれているに注意が必要である。

4. 展望と考察

4.1 今後の展望

次に今後の展望について幾つか述べていく。

二次元的なスライドの実現に関して、次のようなことが考えられる。jCarouselLite の元々の機能であらかじめ指定した方向として縦もしくは横向きにスライドするアニメーション関数がある。これは、元々指定していた方向のスライド一枚あたりの縦、もしくは横の長さを取得しておき、その方向に向かいスライドさせる仕組みになっている。それをアニメーションを呼び出す際に、毎回スライドする方向を指定することで、両方向のスライドの長さを確保しておきさえすれば二次元的なアニメーション自体は理論的には実現できる。ただし、そのアニメーションを実現するためのリスト構造の処理が複雑化することは避けられない。今回は、HTML などの知識が余りない人向けにも使い易いものを開発するというのを主眼としており、HTML の記述も複雑になるかねない二次元表示の機能実装は先送りした。今後実装する場合には導入をし易くする方法を検討していきたい。

サムネイル画像を利用してページを並べ直すという機能もあると便利であろう。具体的には目次表示を利用してサムネイル画像を表示し、それぞれのサムネイル画像をドラッグして並べ直しを行うという GUI の機構である。現在の実装ではメニューのクラス名が動作に関連しているため、現在の実装を拡張してこの機構を実現するには、並べ直しの際にクラス名を変えたりメニューのスライド名自体も入れ替える必要がでてくる。より簡単な機構を考案し実装することは将来の課題とする。

クロスブラウザ対策に関しては、現在ブラウザ毎に表示が異なってしまう箇所において、ブラウザ毎に異なる出力を行っている。しかし、表示サイズといった限定された部分にのみ問題が表れるため、ブラウザ毎の表示サイズの比率がわかっしまえば、一括して表示サイズを変更する仕組みを用意できると考えられる。その為、表示サイズの比率等今後調査していく必要があるだろう。

4.2 類似研究との比較

Web 上でのプレゼンテーションで動的なコンテンツも扱えるという観点で類似する研究として、デモ画面を連続する Web ページとして表示するもの[3]や、Web を用いた参加型プレゼンテーション[4]に注目する。

前者は、デモ画面を Web ページとして記録し、そのページ遷移を JavaScript で自動的に行うという機構を用意するもので、動画を用いるよりも軽量にデモを行うことができる。この方法は Web 上で公開するという用途では jSpow よりも優れるが、発表者が操作をしながらプレゼンテーションを行うという場面では我々の jSpow の方が柔軟な対応が可能であり自由度も高いと言えるだろう。状況に応じて使い分けていくのがよいだろう。

後者は公演者用と視聴者用のスライドショーを表示するシステムで、公演者のスライドに対して視聴者のスライドの同期を取る仕組みである。同期を取ると同時に全体でコメントを共有できリアルタイムの検討が行えるという仕組みである。一般的なリアルタイムのコメント共有の手段としてチャットがあるが、Web ページにチャットと jSpow を同時に導入することで、類似する機能も簡易的に提供できるだろう。

4.3 おわりに

今回、本研究で `iframe` を使い、Web 上における情報の流用をスライドショーという形で行った。外部公開しての評価などは行っていないものの利用の仕方によっては高い利便性があると考えている。また、§4.1 でふれた2次元的なスライドショーを Flash で行うものなども出現しているが、外部 URL をそのまま利用することはまだ行われていないようで、HP のリンク一覧を2次元配列のアニメーションで行うものであった。情報を2次元的に提示する手法はウィンドウ管理にも取り入れられるようになってきており、今後の発展が期待されるので、我々の jSpow でも逸早く取り組みたい。

また別の観点では、プレゼンテーションツールとしての Web ブラウザにはフォントや表示を変更したり、ブラウザ上にマーカーを加えるなど、プレゼンテーション自体をわかり易く、見易くするための飾り付けや追加機構の導入も考えられる。特に Mozilla Firefox 等ではページ上にマーカーをつけるアドオンも存在し、こうした機能を併用することにより、より効果的なプレゼンテーションが行えると考える。

さらに、問題点として HTML 知識をもっている人でも記述が複雑で利用することが面倒に感じてしまうことが挙げられる。これは、専用の編集ツールを作成することで解決するであろう。また、HTML の記述ができるできないに関係せず、専用のエディタがあれば jSpow の利用の補助になりうるので、利用者層を拡大できるだろう。今後は専用エディタの開発も検討していきたい。

参考文献

- [1] jQuery Documentation, <http://docs.jquery.com/>.
- [2] jCarouselLite Documentation, <http://www.gmarwaha.com/jquery/jcarousellite/#doc>.
- [3] Aoki, Y., Ando, F., Nakajima, A., Creating Web-based Presentations by Demonstration. Special Issue on Multimedia Network System, *IPSS Journal*, Vol. 42, No. 2, pp.155-165 (2001).
- [4] 黒河優介, 藤枝崇史, 福本佳史, 関戸亮介, 服部達也, 萩野達, Web を用いた参加型プレゼンテーション. 第 16 回インタラクティブシステムとソフトウェアに関するワークショップ WISS2008, <http://www.tom.sfc.keio.ac.jp/~fjedi/presen.pdf>.