

SVG、XHTML中心の複合文書 異なるDOM空間の操作

XMLコンソーシアム基盤技術部会
ナレッジ-文書ワーキンググループ
株式会社フジミック 大坂哲司

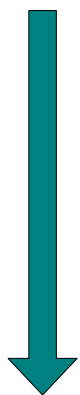
本日の内容

- 1 . SVGについて
- 2 . 複合文書について
- 3 . XHTML-DOM操作によるSVGファイルのエンベッド処理
- 4 . HTML、XHTML、XML、SVGのDOM操作
- 5 . XHTML文中のSVG要素のエンベッド処理

添付資料 SVGレファレンス

S V G (Scalable Vector Graphics) とは

W 3 C は2001年9月、
2次元のベクターデータ / ラスタデータを記述するための
XML を、S V G として勧告しました。



S V G は、新しい画像フォーマットではありません。

S V G は、HTML にはできないコンテンツの表現ができます。
特に名前の通り、2 D グラフィック表現は大変豊かです。

S V Gの特徴

ベクターグラフィックス機能

斜め線の描画

曲線の描画(円、楕円、2次曲線、3次曲線)

ベクター文字の表示(任意の回転に対応)

豊かな表現

アニメーション(CSS属性: style情報の操作として色変更など、
XML属性: 移動や変形などの操作が容易)

透過効果

各種フィルタ処理(リアルな表現)

インタラクティブ

DOM操作が、XML - DOMに準拠している。

ダイナミック

テキストであるため(サーバー)プログラムによる生成が可能
(Flashは専用のアプリケーションが必要)

SVGの基本的なボキャブラリ

コンテナー機能

<code>svg</code>	SVGドキュメントの宣言
<code>title</code>	タイトル記述
<code>desc</code>	コメント記述
<code>g</code>	表示要素のグループ化
<code>defs</code>	コンテナー宣言
<code>symbol</code>	コンテナー中でのグループ化
<code>use</code>	<code>defs</code> 中のコンテナー参照
<code>tref</code>	<code>defs</code> 中のテキストコンテナー参照

アニメーション機能

<code>animate</code>	XMLやCSSの属性を変化
<code>set</code>	XMLやCSSの属性を変化
<code>animateMotion</code>	指定したpath上での移動
<code>animateColor</code>	CSSの色属性を変化
<code>animateTransform</code>	XMLのtransform属性を変化

基本機能

<code>rect</code>	四角形の描画
<code>circle</code>	円の描画
<code>ellipse</code>	楕円の描画
<code>line</code>	線の描画
<code>polyline</code>	折れ線の描画
<code>polygon</code>	多角形の描画
<code>path</code>	汎用曲線描画
<code>text</code>	テキストの表示
<code>tspan</code>	テキストへの部分修飾
<code>textPath</code>	path上へのテキスト表示
<code>image</code>	イメージの表示
<code>a</code>	リンク設定

SVGレファレンス参照

複合文書

特定のアプリケーション(群)専用にXMLベースの言語を設計するのではなく、**文書交換目的の汎用性、可搬性、可読性を維持しやすいXML複合文書**はどうあるべきだろうか。

極力、XMLファミリー言語として標準化されたシンタックス、標準のボキャブラリを組み合わせで多重の埋め込みオブジェクトを**2次元の文書イメージ上にレイアウトできるような手法**が1つの解であろう。

(野村、川口ら 2001)

具体的には

現在利用できる環境における複合文書の検討

XHTML、SVG、XMLのDOMの操作

XHTML、SVGのCSSの操作

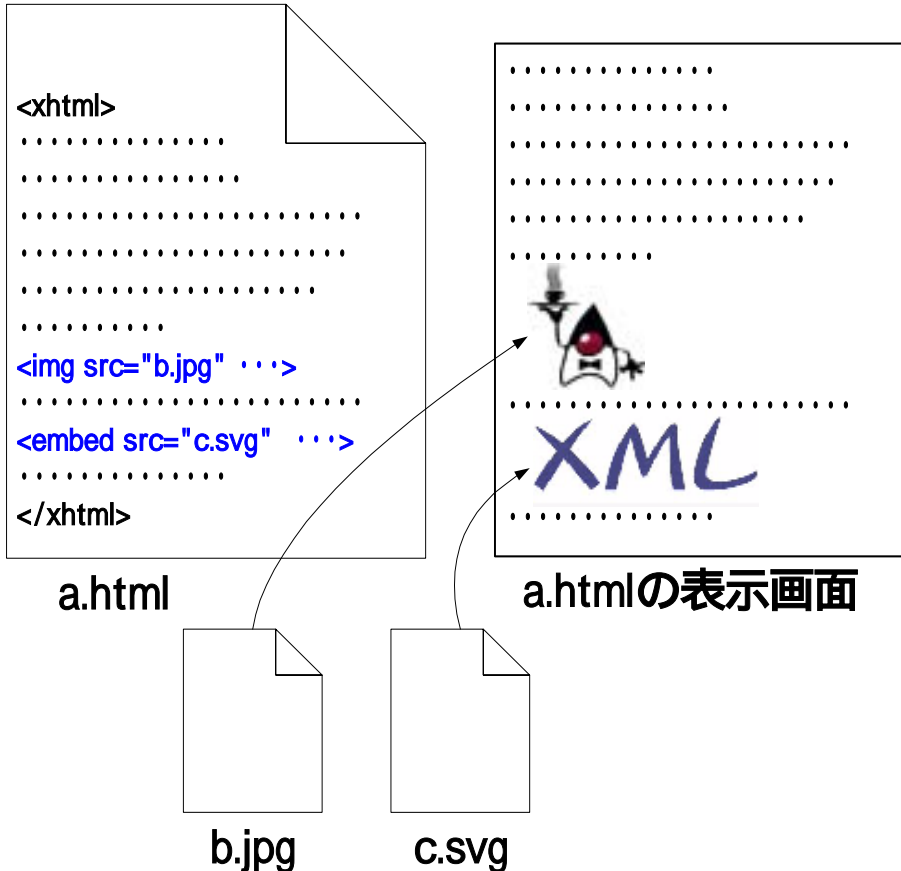
インタラクティブでマルチモーダルな複合文書を実現

ここで言う複合文書とは、XHTML文書中にSVG文書を埋め込み、一つの文書とした形態。

インタラクティブな複合文書とは、必要に応じてSVG文書を挿入、入替、削除を行うことのできる文書。

前提：ブラウザとしてマイクロソフトのIE 5.0以上、アドビシステムズのSVG Viewer 2.0以上を使用。

HTMLでの埋め込み処理



埋め込み記述の中で、オブジェクトの表示サイズを指定した場合、そのサイズでオブジェクトを表示する。指定した表示サイズのアスペクト比(縦横比)とオブジェクトのアスペクト比が異なると、歪なオブジェクト表示になる。

DynamicHTMLを用いて、これらのオブジェクトを表示したり、非表示にしたりすることができるが、オブジェクトの表示を望まないとき(非表示状態)においてもオブジェクトはブラウザにロードされるため、不要な通信トラフィックが生じる。

埋め込み記述の基本形

```

```

```
<embed src="c.svg" width="400" height="200" />
```

```
<object type="image/svg+xml" data="a.svg" style="width:400; height:200" />
```

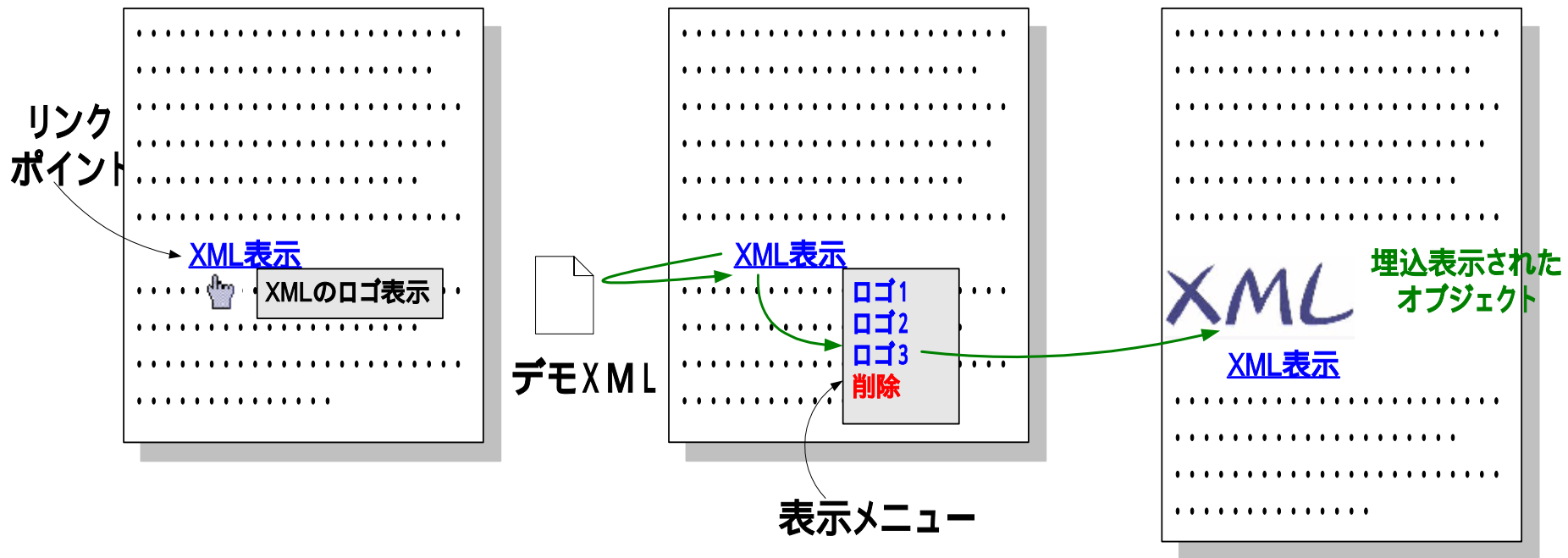

SVGの埋め込み

ケース1 . SVGファイルの埋め込み表示

文書中のリンクポイントから複数のSVGファイルの埋め込み表示

ケース2 . XHTML (HTML) 中のSVGデータの埋め込み表示

リンクメニューからの埋め込み表示



マウスカursorがリンクポイントに入ると、リンクに関する吹き出しが表示される。

リンクポイントでマウスをクリックすると、リンクメニューが表示される。

リンクメニュー中の項目を選ぶと、対応するファイルが埋め込み表示される。

リンクポイントの記述

1. アンカー要素

```
<a id="ap1" onmouseover="focus_txt()" onmouseout="blur_txt()"
  onclick="dispMenu()">SVGの基本サンプル</a>
```

2. テキストの一部をリンクポイントに(p、span、divなど)

```
<p id="ap1" onmouseover="focus_txt()" onmouseout="blur_txt()"
  onclick="dispMenu()"
  style="color:blue;text-decoration:underline">SVGの基本サンプル</p>
```

3. イメージボタン

```
<div id="ap1" onmouseover="focus_btn()" onmouseout="blur_btn()"
  onclick="dispMenu()">
  
  
</div>
```

4. フォームボタン

```
<form id="ap1">
  <input id="ap1_i" type="button" onclick="dispMenu();" value="表示" style="color:blue"
    onmouseover="focus_txt()" onmouseout="blur_txt()" />
</form>
```

SVGファイルの埋め込み方法

HTMLでの表記

```
<embed src="c.svg" width="400" height="200" />
```

スクリプトによる表記(ダイナミックな埋め込み処理が可能となる)

```
function embedSVG(src, id, w, h, node, document){  
    var oNode=addTextNode("embed", "src", src, "", node, document);  
    oNode.setAttribute("id", id);  
    oNode.setAttribute("width", w);  
    oNode.setAttribute("height", h);  
    oNode.setAttribute("type", "image/svg+xml");  
}
```

ノード生成関数

// タグ tagName で、attrName で attrValue の属性を持つ text の要素ノードを生成し、oNode に追加する。
// docs は対象とする DOM を指定する。

```
function addTextNode(tagName, attrName, attrValue, text, oNode, docs) {  
  // tagName の指定があれば、その要素を生成し oNode に追加する。  
  if(tagName){  
    var oNewElement=docs.createElement(tagName);  
    // attrName の指定があれば、attrValue の属性を設定する。  
    if(attrName)  
      oNewElement.setAttribute(attrName,attrValue);  
    // テキストの指定があれば、oNewElement にテキストオブジェクトを生成し、追加する。  
    if(text){  
      var oNewText=docs.createTextNode(text);  
      oNewElement.appendChild(oNewText);  
    }  
    oNode.appendChild(oNewElement);  
  }  
  return oNewElement;  
}
```

この関数はHTML、XHTML、XML、SVGなどのDOM操作に利用できる汎用関数である。

XLinkによるリンク設定

```
<links xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="extended">
<resources>
  <resource xlink:type="locator" xlink:label="A" xlink:href="index.html" xlink:title="SVGのページ" />
  <resource xlink:type="locator" xlink:label="B" xlink:href="SVGViewer.svg" xlink:title="アドビSVGゾーン" />
  <resource xlink:type="locator" xlink:label="C" xlink:href="t.svg" xlink:title="SVGいろいろ" />
  <resource xlink:type="locator" xlink:label="D" xlink:href="arcs02.svg" xlink:title="アーク" />
  <resource xlink:type="locator" xlink:label="E" xlink:href="script01.svg" xlink:title="インタラクティブSVG" />
</resources>

<arcs>
  <arc xlink:type="arc" xlink:from="A" xlink:to="" xlink:show="embed" />
</arcs>
</links>
```

XLinkの拡張リンクにより、1対Nのリンクを実現

埋め込み表示の留意点

ファイルの埋め込み処理では、次のような留意点がある。

エンベッド空間とSVG空間の大きさの差異

エンベッド空間とSVG空間のアスペクト比(縦横比)の不一致

エンベッド空間の定義

```
<embed src="c.svg" width="400" height="200"/>>
```

SVG空間の定義

```
<svg width="400" height="200" >
```

ファイルの埋め込み表示ではSVGの制御を必要としていない。

HTML、XHTML、XML、SVGのDOM操作

HTML - DOM
XHTML - DOM

SVG - DOM



XML - DOM

各DOM相違点

rootDocumentの取得

style属性の操作

XSL、XSLTの利用

JavaScriptとJavaの相違点

rootDocumentの取得

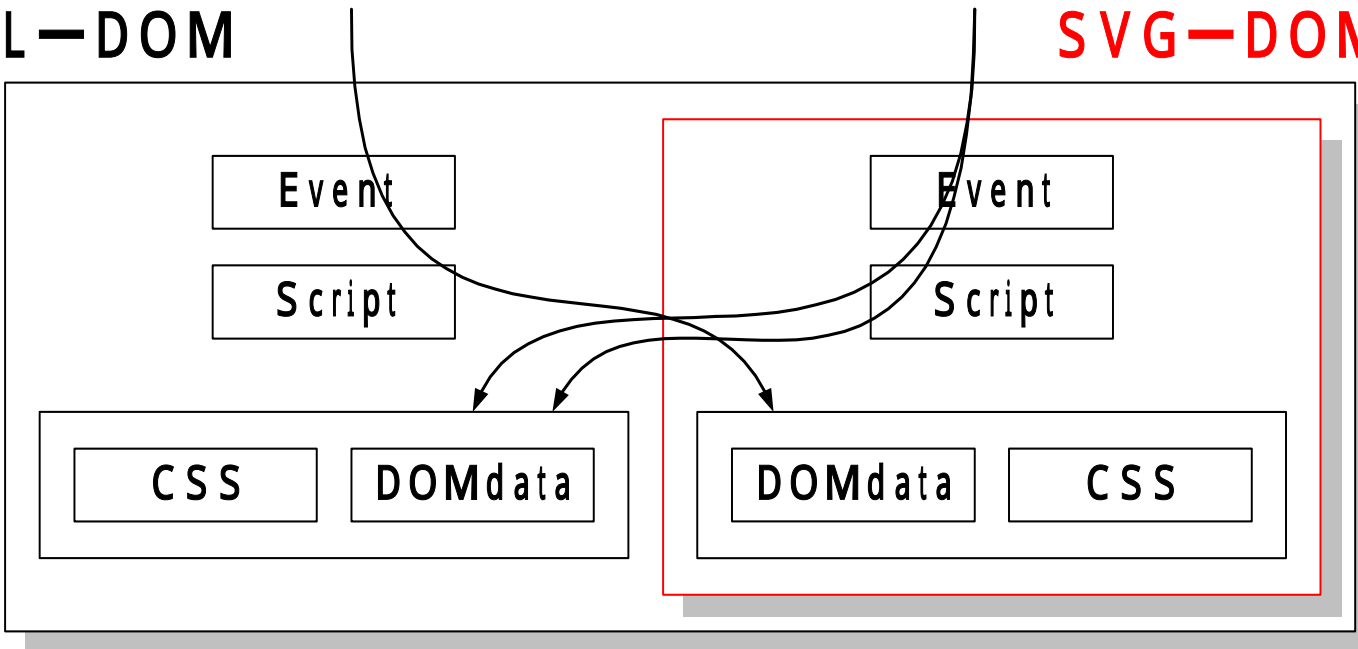
HTMLレンダリング

コレクションの利用

HTMLからSVG、SVGからHTML

HTML-DOM

SVG-DOM



```
var node=event.srcElement;  
var id=node.getAttribute("id");  
var text=node.firstChild.data;
```

```
var node=event.getTarget();  
var id=node.getAttribute("id");  
var text=node.firstChild.data;
```

SVGの埋め込み

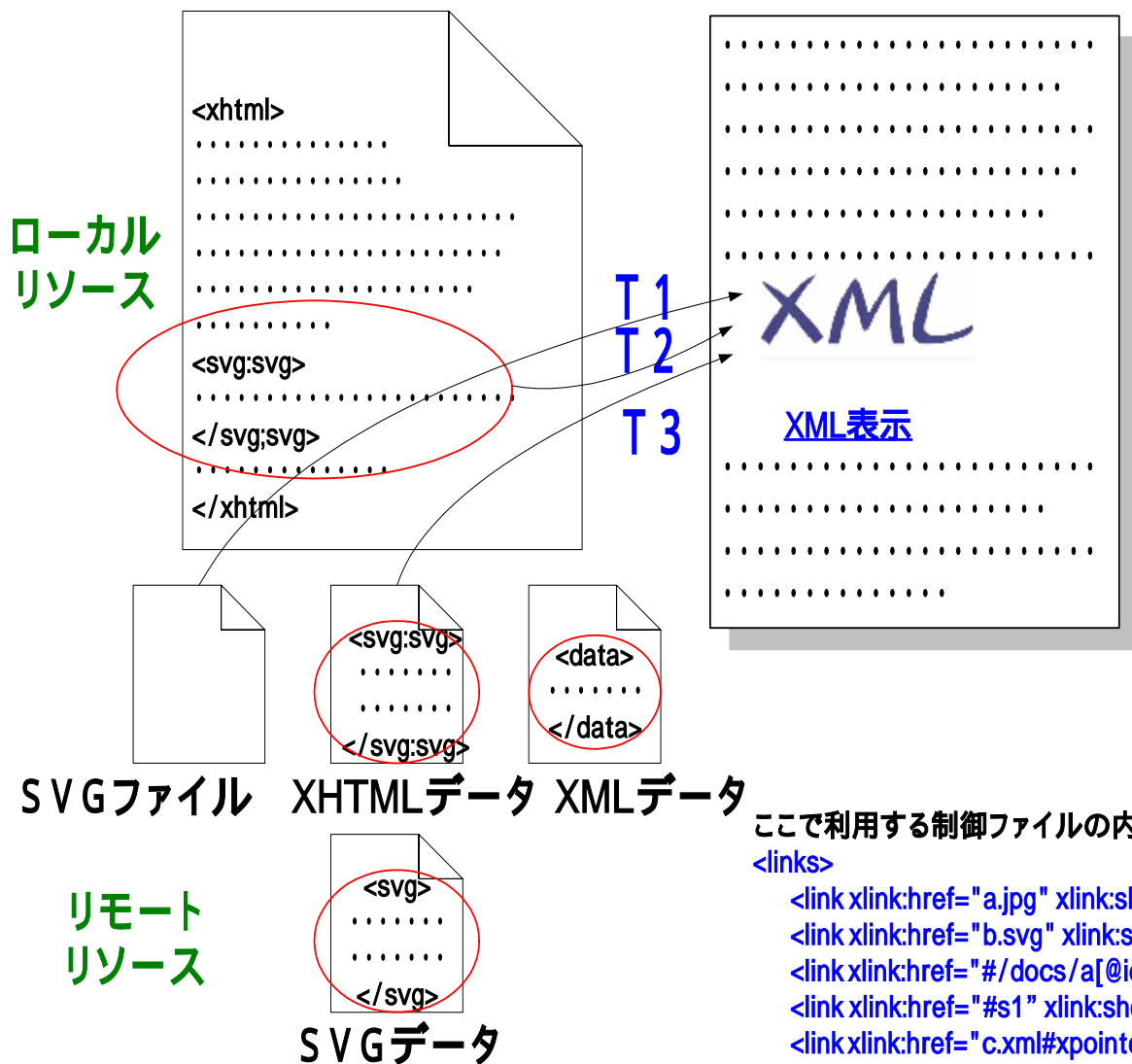
ケース1 . SVGファイルの埋め込み表示

文書中のリンクポイントから複数のSVGファイルの埋め込み表示

ケース2 . XHTML (HTML) 中のSVGデータの埋め込み表示

多様な埋め込み表示形態への対応

埋め込みオブジェクトのリソース形態



埋め込み表示オブジェクトのリソース形態は次の通りである。

T1 . 最も一般的な形態で、リモートリソース(当該HTML内にはないリソース)で、フォルダにあるファイルやネットワーク上にあるファイルを埋め込む表示を行う形態である。制御ファイル中で `xlink:show = "embed"` の指定を行う。

T2 . 当該HTML中にある指定のデータを抽出し、そのデータを埋め込み表示を行う形態である。制御ファイル中で `xlink:show = "inline"` の指定を行い、`href` は # で始まるデータ名を指定する。

T3 . リモートリソース中の指定のデータを抽出し、そのデータを埋め込み表示する形態である。制御ファイル中で `xlink:show = "inline"` の指定を行い、`href` 対象ファイル名の後に # を置きデータ名を指定する。

尚、T2、T3でデータ名を検索する記述方法としてXPath記述、XPointer記述もサポートする。

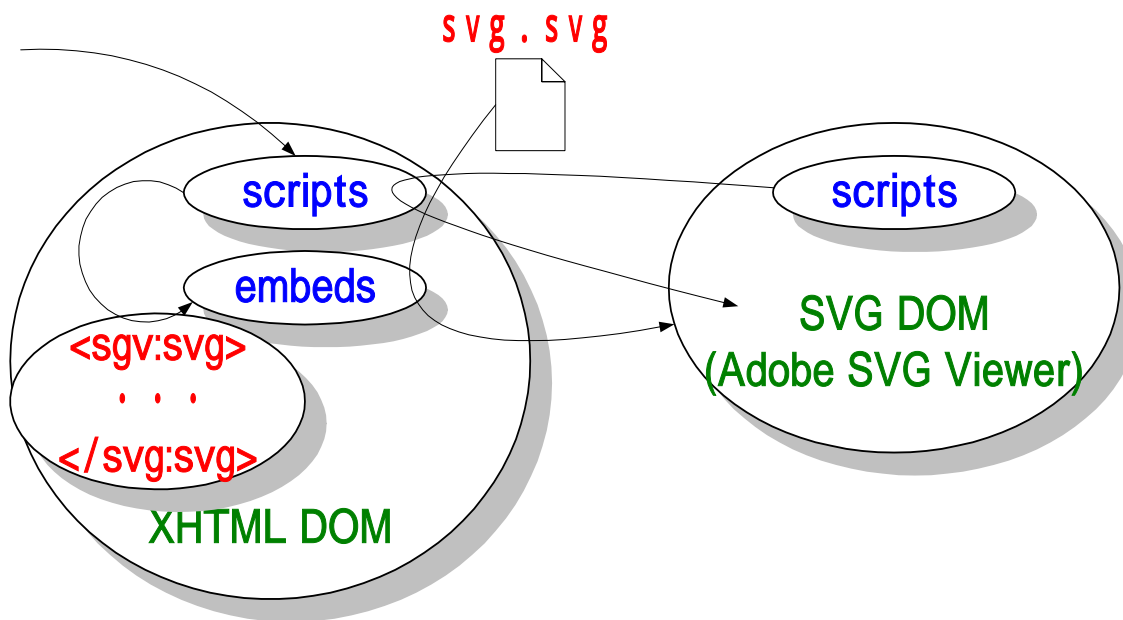
ここで利用する制御ファイルの内容

`<links>`

```
<link xlink:href="a.jpg" xlink:show="embed">ロゴ1</link>
<link xlink:href="b.svg" xlink:show="embed">ロゴ2</link>
<link xlink:href="#/docs/a[@id='s1']" xlink:show="inline">SVGロゴ1</link>
<link xlink:href="#s1" xlink:show="inline">SVGロゴ2</link>
<link xlink:href="c.xml#xpointer(/docs/a)" xlink:show="inline">ロゴ説明</link>
<link xlink:href="javascript:delete()" xlink:show="delete">削除</link>
```

`</links>`

SVGオブジェクト埋め込み表示の機構



画面上のリンクポイントをクリックすると、SVGを埋め込む処理が実行される。イベントが生じたポイントのid情報を取得する。

埋め込みスクリプトは、取得したidに基づいて、XHTML - DOMから対応するSVGデータを読み込む。このデータ中からsvg要素にあるSVG空間の大きさを取得し、その大きさに従ったエンベッド空間を生成する。

エンベッド空間にテンプレートとなるsvgファイルを読み込む。このファイルはSVGレンダラーを起動するためのもので、svg要素以外に何も無いSVGファイルである。但し、svg要素の属性として、onLoad関数があり、SVGファイル読み込み完了後に実行する関数を記述する。この関数はXHTML内の関数である。

onLoad関数は、関数内のSVGデータから、SVG空間の大きさ、その他の属性をsvg要素に追加する。次にSVGデータを読み込み、それらの属性をsvg要素に追加することで、XHTML内にSVGが埋め込み表示を実現する。

svg.svgの内容

```
<?xml version="1.0" standalone="no"?>

<!DOCTYPE svg PUBLIC "-//W3C//DTD
SVG 20001102//EN" "http://www.w3.org/
TR/2000/CR-SVG-20001102/DTD/svg-
20001102.dtd">

<svg id="svg" onload="dispSVG(evt);"
xmlns:xlink="http://www.w3.org/1999/
xlink">

</svg>
```

通常embed要素によってSVGコンテンツが埋め込まれる。ここで、エンベッド空間とSVG空間の大きさに相違がある場合、適切な表示を得ることができない。この方法によると、それぞれの空間の大きさが一致させることができる。

埋め込み表示の確認状況

- ・描画要素の描画
- ・アニメーション
- ・xlink:href属性を利用した要素の描画

IEの制約で、複数のSVGオブジェクトを一括して表示すると並びに表示された複数のSVGを印刷することはできない。

まとめ

現在利用できる環境で、複合文書を実現するためには、DOM、CSSの操作が必須です。

これらを操作するスクリプトによって、ダイナミックで、インタラクティブでマルチモーダルな複合文書が実現できます。

複合文書の生成にXLinkの概念が今後必要となるでしょう。

SVGは、携帯端末向けのTiny SVGなどファミリを増殖中です。今後、ますます目を離せない技術となるでしょう。

参考資料

- ・[AdobeSVGViewer2.0] "Adobe SVG Viewer Download Area", <http://www.adobe.com/svg/viewer/install/old.html>, 4/2001
- ・[KevLinDev2001] "advanced SVG Tutorial", <http://www.kevlindev.com/tutorials/>
- ・[Macromedia]<http://www.macromedia.com/jp/software/flash/>
- ・[Microsoft] "About the W3C Document Object Model", <http://msdn.microsoft.com/workshop/author/dom/domoverview.asp>
- ・[SVG1.0a] "Scalable Vector Graphics (SVG) 1.0 Specification", <http://www.w3.org/TR/2001/REC-SVG-20010904/>, 9/2001
- ・[SVG1.0b] "Scalable Vector Graphics (SVG) 1.0 の公開について (W3C 勧告)", <http://www.w3.org/2001/09/svg1-pressrelease.html>, 9/2001
- ・[SVGZone] "SVGゾーン", <http://www.adobe.co.jp/svg/>
- ・[XHTML1.0] "XHTML 1.0: The Extensible HyperText Markup Language", <http://www.w3.org/TR/xhtml1/>, 6/2000
- ・[XLink1.0] "XML Linking Language (XLink) Version 1.0", <http://www.w3.org/TR/2001/REC-xlink-20010627/>, 6/2001
- ・[ZDNN2000] 「.NET」のポイントは“妥協のないユーザー体験”, <http://www.zdnet.co.jp/news/0006/23/ms.html>, 6/2000
- ・[大坂2000a] 大坂: "XMLを用いたカタログ作成", ON DEMAND Printing Japan 2000, 4/2000
- ・[大坂2000b] 大坂: "XML / PDFを活用したソリューション", PDFConference2000, 7/2000
- ・[大坂2001] 大坂、野村: "SVG-DOMによる豊かなクライアントXML表現力", XMLコンソーシアムDay, 11/2001
- ・[大坂2002] 大坂: "XLinkについて 付箋紙をXLinkで動かしてみる", XMLコンソーシアムWeek, 6/2002
- ・[大坂・野村] 大坂、野村: "SVG-DOMによるアニメーションとXHTML中心複合文書の可能性", 情報処理学会デジタルドキュメント研究会予稿集, 3/2002
- ・[野村] 野村、川口ら: "XML複合文書の実現方式に関する一考察 XHTML, SVG, MathMLのポキャブラリ活用によるアプローチの有効性について", 情報処理学会デジタルドキュメント研究会予稿集, 3/2001

実装並びに作動環境

OS: Windows2000, WindowsNT, WindowsXP

Browser: IE 5.0, IE 5.5, IE 6.0

SVGViewer: Adobe SVG Viewer 2.0 並びに 3.0

作成ツール: テキストエディタ(秀丸)

ご質問の受付 フジミック 大坂まで
(osaka@fujimic.com)