デジタル・ドキュメント　27－4
（２００１．３．１４）

ＸＭＬ複合文書の実現方式に関する一考察

XHTML, SVG, MathML のボキャブラリ活用によるアプローチの有効性について

○野村直之*1　川口浩司*2　坂川浩二郎*3　小林亜令*4　高木康夫*5　小林茂*5　水野雅裕*6
蔵田憲彦*7　松井善郎*1　横田寧子*1　藤岡慎弥*8
*1 ジャストシステム　*2 セック　*3 日立製作所　*4 KDD 研究所　*5 日本ユニシス
*6 IBM システム・エンジニアリング　*7 CRC 総合研究所　*8 沖電気工業
＊全員 Java コンソーシアム XML 部会兼務

概要：　２１世紀の可搬データ表現、知識表現、Web 構築素材としてＸＭＬ基盤技術が注目を浴びている。本稿では、ＸＭＬによる文書処理の技術的可能性と普及可能性を占うため、XML ベースの文書記述言語 XHTML と SVG、MathML の組み合わせを中心に、可搬性、可用性の高い複合文書の可能性を考察する。また、数種類の実現手法を提案し、その一部については各種処理系による動作状況を確認する。また従来の特定プラットフォームに依存した複合文書実現方式との比較、得失の分析を踏まえ、今後、ＸＭＬ複合文書に求められる機能や、期待される応用用途について簡単に論じる。

A Note on XML Compound Document using the Vocabularies mainly from XHTML, SVG and MathML

○Naoyuki NOMURA*1　Koji KAWAGUCHI*2　Kojiro SAKAGAWA*3　Arei KOBAYASHI*4
Yasuo TAKAGI*5　Shigeru KOBAYASHI*5　Masahiro MIZUNO*6
Norihiko KURATA*7　Yoshiro MATSUI*1　Yasuko YOKOTA*1　Shinya FUJIOKA*8
*1 Justsystem Inc.　*2 SEC Corp.　*3 Hitachi Ltd.　*4 KDD Labs Ltd.　*5 Nihon Unisys　*6 IBM Systems Engineering　*7 CRC Research　*8 Oki Electric Ltd.

Abstract:　XML, the portable data description language is supposed to be able to enhance the capabilities of document exchange, knowledge representation and WWW services.　This paper proposes several possible design methods to implement XML Compound Document, and discusses the pros and cons of each method.

1 はじめに

次世代 Web は、意味表現を備え、"Semantic Web" と呼ばれる世界知識庫になる、と規格標準化団体W3C(World Wide Web Consortium)が表明している[Lee2000]。ここでいう「意味」は、（1）情報をその生来の構造に即して必要なだけ構造化して表現する、（2）リポジトリによって世界知識体系上で絶対的もしくは相対的に位置付ける、の２通りの仕組みで与えられる[野村 2000]。（1）を担うのがＸＭＬファミリー言語の規格群であり、（2）を担うのが URI(Universal Resource Identifier)や RDF(Resource Description Format)などの、Web 上での書誌情報（メタ情報）記述の仕組みである。Lee2000 によれば、さらに上位の知識表現記述の枠組みや、安全で効率的な知識流通のためのセキュリティ関連の規格等が、任意に取捨選択、組み合わせて使えるように、モジュールの組み合わせとして提示されている。

このように、ビジネスや生活の基盤となる知識インフラ、情報通信インフラの根幹が XML ファミリー規格に基づいたものになろうとしている中で、Word 文書や一太郎文書形式が交換、編集目的で今世紀中使われ続けるとは考え難い。特定のアプリ専用に複雑・大規模化した XML ベースの専用言語も同様の運命を辿るかもしれない。

文書処理、特に文書交換、それも全地球規模で、何のアプリを使っているかもわからない相手と共同執筆したり知識の共同創成を日常的に行えるような、効率良い、且つ多彩な表現能力を備えた文書フォーマットはどうあるべきであろうか。本稿では、OLE Compound Document [OLE2000]等が備える、異なる表現メディア（'オブジェクト'）を枠の中に編集可能な形で埋め込めるという機能性や、実際のビジネス現場での文書運用のニーズと対比しながら、XML ファミリー言語規格を組み合わせて XML 複合文書を構成するガイドライン構築の議論の一端緒を開いてみたい。

以下、まず第２節では、XML 複合文書の備えるべき要件として、文書記述の標準言語 XHTML と、ベクトル図形を中心に単独でも複合文書を視野に入れた SVG、そして数式表現の領域(オブジェクト)やその数値計算結果を他の図形、表、文章表現中に連動表示させることを視野に入れて、MathML とを組み合わせた複合文書実現法について考察する。そして、第３節で、これらの考察に基づいて試作した文書ソースを、両言語を表示できる規範的な処理系に表示させてみた実例を示し、その結果を比較、一覧する。その際に、文書の表現力の観点から、各要素に修飾を施す CSS(Cascading Style Sheet)が３つの言語とその組み合わせにどう効いているかについても調査した結果をまとめる。第４節では、複合文書の備えるべき機能として、複数表現形式の複合という基本機能に加え、機能の連動やデータ間の連携、そして、編集、改版、流通、流用の便宜のための書誌情報や履歴の管理、といった観点で、今後の言語拡張や高次言語の可能性、そして、次世代の処理系や次世代インターネットに期待される要求仕様について考察する。

2 XML 複合文書の備えるべき要件と３つの実現法　～XHTML, SVG, MathML による複合文書の提案

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て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。

そこで、ベースとなる個別言語として、文書レイアウト論理記述言語の XHTML1.1 [XHTML2000]とベクトル図形記述言語の SVG [SVG2000] 、さらに、MathML [MathML2000]とを採用した。XHTML と SVG は共にモジュール化されていて汎用性・拡張性に富み、広く普及した標準ボキャブラリを備える。両者

共にハイパーリンク・モジュールを備え、単独でも、他リソースとのインタフェースを備えた複合文書コンテンツ表現言語となり得る位の充実した仕様である。

MathML はＸＨＴＭＬ、ＳＶＧに比べれば、言語仕様自体の拡張性は乏しいようである。複合文書という目的に即して言い換えれば、他の言語を埋め込めるコンテナ言語としての設計はなされていないようである。また、早期に仕様が策定されたせいか、はたまたソースを直接手入力する前提で設計されたせいか、２文字程度の要素名が多く、可読性、ひいては文書交換性の観点で今後 XHTML、SVG なみに広範に利用されるかどうか疑問な点もある。これらの事情から、今後の Web ブラウザは、ＸＨＴＭＬ、ＳＶＧには標準で対応すべきだが、MathML はあくまで他の無数の個別言語と同様に専用のプラグインが作れれば良い、という仕様の進化のガイドラインをもつべきかもしれない。

しかし、とりあえず数学的表現の標準ボキャブラリであること、そして、数式表現言語の付加的機能として、ＸＨＴＭＬから受け取った数値を計算し、それをＳＶＧで読みとって、グラフ描画するようなＸＳＬＴによる、XML 複合文書間の変換が容易に作れそうである、という見通しの下、これら３つの標準言語を選択し、意味のある混在が可能であることを確認することにした。

まず、XHTML と SVG とを組み合わせた複合文書として次の３種の実現方式を考えた：
(1)XHTML1.1 の新モジュールとして、SVG(の一部) を'ドライバ'[XHTML2000] で定義
(2)SVG に一部の XHTML1.1 要素の定義を埋め込み DTD で記述するなどして個別拡張
(3)SVG と XHTML1.1 を名前空間でマージ；要素を互いに多重に入れ子で埋め込む
方法(1)は XHTML1.1 を主言語とし、それに従属するサブモジュールとして SVG を利用するものであり、逆に SVG を主言語としてその拡張部分に XHTML を埋め込むのが方法(2)である。いずれの場合も主従関係が存在し、モジュール定義の部分で独自拡張が必要となる。これに対して、方法(3)は、名前空間の語彙使用の構造パタン記述ガイドラインを設けるだけで、独自拡張部分無しという利点がある。規格としては最も弱いものになるが、原則的に双方に主従関係無い点、および既存の処理系で早期に実用になる点から今回実験の対象として採用した。また、(2)についても、foreignObject という、SVG 拡張の作法に従って表示可能な処理系が存在したため実験の対象に加えた。

また、多種の言語を共存させていく試金石の意味で、第三の言語として MathML を調査対象に加えた。MathML については、本来の数式表現言語としての役割に加えて、XSLT で変換した際に、XHTML の文章中から数値を取得して数値計算を行い、その結果を SVG でグラフにする、などの、複合文書としての新たな機能性を期待することができる。さらに、本来の文書の表現力の観点から、各要素に修飾を施す CSS(Cascading Style Sheet)が３つの言語とその組み合わせにどう効いているかについても調査した。

3 既存のブラウザ類による表示実験とその結果

まず、XHTML と SVG とを、名前空間で混在させ、互いに多重に入れ子にしたソースを作成し、既存のブラウザ、プラグインによる表示実験を行った[野村 2001b]。SVG の中に XHTML を記述する際には、[SVG2000]で標準の拡張手法とされている<foreignObject>を使用し、XHTML 文書を挿入する。XHTML の中に SVG を記述する際には、XHTML 中で<svg>を使用し、現在広く使われている処理系(Web ブラウザ)に対し、矩形領域を確保してインライン画像を表示させる方針をとった。

図１(a)に、<foreignObject>により SVG 中に XHTML 要素を埋め込んだソース例を示し、図２(a)に、XHTML 中で<svg>を使用したソース例を示す。また、図１(b)に、図１(a)のソースから期待される表示イメージを示し、図２(b)に、図２(a)のソースから期待される表示イメージを示す。

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20001102//EN"
  "http://www.w3.org/TR/2000/CR-SVG-
      20001102/DTD/svg-20001102.dtd">
<svg xmlns:html="http://www.w3.org/1999/xhtml">
  <g>
    <polygon points="50 0 0 50 100 50"/>
    <polyline  points="50  100  0  150  100  150"
      style="fill:none; stroke: red;"/>
  </g>
  <foreignObject   x="200"   y="200"   width="200"
      height="200">
    <html:h1>Header 1</html:h1>
    <html:p>
       This is a test, HTML page, in SVG.
       <html:br/>
       <html:img
      src="http://www.mozilla.org/projects/svg/imag
      es/svgmoz.png"/>
       <html:br/>
       PNG Image Here.
    </html:p>
  </foreignObject>
</svg>
```

図1 (a)　SVG 中に XHTML 要素を埋め込んだソース

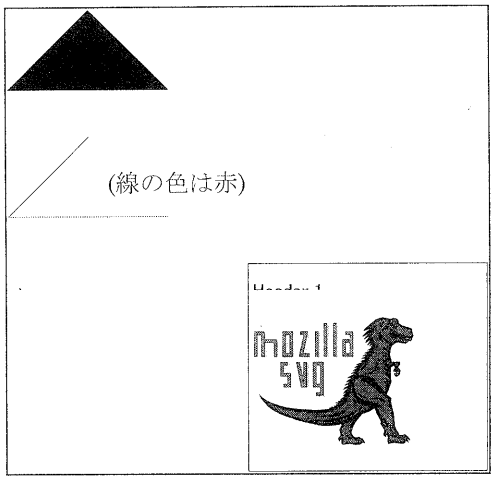

図1 (b) 'SVG 中に XHTML' の表示イメージ

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE  html  PUBLIC  "-//W3C//DTD  XHTML  1.0
      Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
      transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
     xmlns:svg="http://www.w3.org/2000/svg">
<h1>Header 1</h1>
<p>
This is a test, HTML page, in SVG.
<br/>
<img src="svgmoz.png"/>
<br/>
PNG Image Here.
<br/>
<svg:svg>
  <svg:g>
    <svg:polygon points="50 0 0 50 100 50"/>
    <svg:polyline  points="50  100  0  150  100  150"
      style="fill:none; stroke:red;"/>
  </svg:g>
</svg:svg>
</p>
</html>
```

図2 (a)　XHTML 中で<svg>を使用したソース

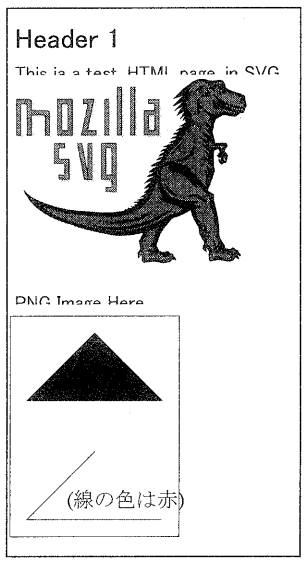

図2 (b) 'XHTML 中の<svg>' の表示イメージ

これらのソースを、W3C 自ら開発、公開している規範的な Web ブラウザ Amaya、それから、全世界的な公開の議論に基づいて仕様拡充、開発が進められている Mozilla という処理系にかけ、名前空間の指定法、スタイルの指定法を様々に切り替えて表示を調べた。さらに、SVG がコンテナの場合は、Adobe SVG Viewer 、IBM SVGView 、Apache Batik SVG という公開ツールで、また、XHTML がコンテナの場合は、Netscape Communicator 、Opera 、Internet Explorer という公開ツールで、表示を調べた。

図1 (a)のソースを、Mozilla Milestone 18 にかけた結果を図3に示す。SVG と HTML は同時に表示されたが、style の指定は無視されている。<foreignObject>があっても無くても同じ表示となること、また、SVG

の基本図形のうち、polygon と polyline しか対応していない。これらの事実から、Mozilla の開発者達には、SVG をコンテナ言語として XHTML をところどころで使用したソースはそこそこ適切に表示させようという、SVG 主体の複合文書推進の意図があった可能性がある。が、図１(a)で明確に意図したほどには、SVG の標準の拡張仕様に準拠して、埋め込み部分を明確に周囲と区別してレイアウトし、表示しようとまで、この種の XML 複合文書の仕様、ガイドラインを想定してはいないようである。

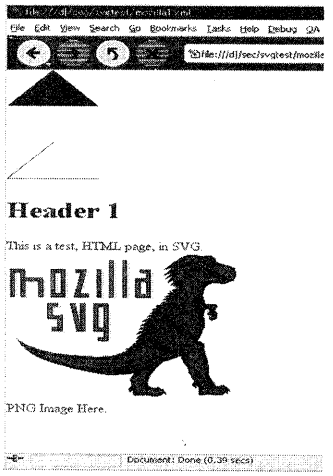

図３ Mozilla で図１(a)を表示

W3Cの規範的ブラウザAmayaでは、SVGは表示されるが、HTMLタグが「Unknown XML Element」エラーになっている。HTMLブラウザに SVG 単独表示を独立的に付加した実装方式になっているようである。少なくとも SVG が一番外側になっている XML 文書については、複合文書を考慮した設計に至っていないといえる。

SVG 用の処理系とされる３つのツールについても同様の結果が予想され、実際、SVG のみ表示され、HTML 部分は無視された。これらの一部は、図１(a)の時点の SVG の仕様よりも古い仕様にのみ対応しているらしきエラーメッセージを出した。例えば、Apache Batik SVG Tookit 1.0beta 付属の SVG Viewer (Release: 2000/12/01) では、以下のエラーメッセージが表示される：

The current document is unable to create an element of requested type (namespace: http://www.w3.org/2000/svg, name: foreignObject)

そこで、「foreignObject」を取り除くと、SVG のみ表示され（XHTML は無視）他のツールと同様の画面となった。但しこれらの SVG 処理系は SVG の最新仕様が規定する範囲に忠実な動作をしているとみなすこともできる。本来、foreignObject 要素の典型的な利用法は、SVG 中に、別の namespace に属する、他の user agent が描いた graphical content を含めることにある、とされているからである。

次に、XHTML がコンテナで SVG を埋め込んだ図２(b)のソースを、Mozilla Milestone 18 にかけた結果を図４に示す。図のように、HTML と SVG が同時に表示されたが、SVG の線上の赤色の style 指定は無視された。なお、図４の表示を得るためには、<html>を以下のように指定する必要がある：

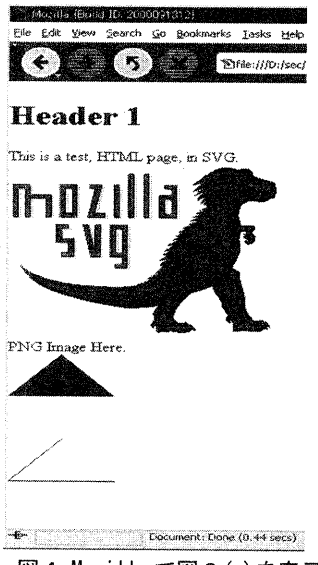

図４ Mozilla で図２(a)を表示

```
<html   xmlns=http://www.w3.org/1999/xhtml
      xmlns:svg="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.svg">
```

Amaya でも、ほぼ図４と同様に、HTML と SVG が同時に表示された。図４との相違は、多少、行間が詰まって見えているだけである。XHTML がコンテナ言語となっているだけあってか、エラー表示などは出ていない。

Netscape Communicator 4.76、および、Opera 5.01 では、HTML 部分のみが表示された。SVG 図形は表示されなかったものの、特にエラー表示などは出ていない。

Internet Explorer 5.5 では、ブラウザ画面に図２(a)のソースが、XML ファイル一般の色づけ、インデント形式で表示された。ファイル拡張子や MIME 指定などを変更すれば、図４もしくは HTML 部分だけの表示となるのかもしれないが、他の処理系と条件が異なってくるため今回は実施していない。XML 文書であれば、原則その論理情報、すなわちソースそのものを木構造の形で汎用的に表

すべきであり、個別言語の仕様、セマンティクスに対応したレンダリングは個別のプラグインが正式に仕様に即して作成されてから行うべきだ、とするInternet Explorer 5.5の設計思想を伺うことができる。

```
<?xml version="1.0" standalone="no"?>
<?xml-stylesheet href="svg-css.css" type="text/css"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20000802//EN"
  "http://www.w3.org/TR/2000/CR-SVG-20000802/DTD/svg-20000802.dtd">
<svg width="15cm" height="15cm">
  <g>
    <text class="title" lang="en" x="10" y="30">
      *****SVG shapes*****
    </text>
    <circle cx="70" cy="100" r="50" />
    <rect x="150" y="50" width="135" height="100" />
    <line x1="325" y1="150" x2="375" y2="50" />
    <line x1="375" y1="50" x2="425" y2="150" />
    <polyline points="50,250,75,350,100,250,125,350,150,250,175,350" />
    <polygon points="250,250,297,284,279,340,220,340,202,284,250,250" />
    <ellipse cx="400" cy="300" rx="72" ry="50" />
    <text class="t1" lang="en" x="50" y="175">Circle</text>
    <text class="t2" lang="en" x="175" y="175">Rectangle</text>
    <text class="t3" lang="en" x="355" y="175">Lines</text>
    <text class="t4" lang="en" x="50" y="375">Polyline</text>
    <text class="t5" lang="en" x="225" y="375">Polygon</text>
    <text class="t6" lang="en" x="375" y="375">Ellipse</text>
  </g>
</svg>
```

図5 (a) SVGの中でCSSを利用

```
text.title          {font-family:
palatino;  font-weight:  bold;
font-size: 32; color: tomato}
text.t1  {font-family:  times;
font-size: 24; color: blue}
text.t2  {font-family:  times;
font-size: 24; color: green}
text.t3  {font-family:  times;
font-size: 24; color: black}
text.t4  {font-family:  times;
font-size: 24; color: pink}
text.t5  {font-family:  times;
font-size: 24; color: brown}
text.t6  {font-family:  times;
font-size: 24; color: magenta}
rect {fill: yellow; stroke: blue}
circle {fill: red; stroke: blue}
line {fill: green; stroke: blue}
polygon  {fill:  blue;  stroke:
blue}
polyline  {fill:  pink;  stroke:
blue}
```

図5 (b) CSSファイル

次に、CSSによるスタイル指定を、SVGのソース中で、あるいは、別ファイル(拡張子.css)で与えられることを、図5のソースを用いて確認した。効果の確認にはカラーが必要なため、結果の概要を文章で記すと、Mozillaではpolygonとpolylineの基本図形は表示されたが、スタイルの指定は無視、Adobe SVG Viewer、Apache Batikでは、正常に表示、Amaya、IBM SVGViewでは図形は表示されたがスタイルの指定は無視、となった。

```
<?xml version="1.0" encoding="iso-8859-1"
    standalone="yes"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
    Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
    transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<ul>
<li>An embedded HTML fragment,
    <code>foreignObject</code>
  in SVG,
  <svg xmlns="http://www.w3.org/2000/svg"
    width="6cm" height="45px">
    <rect y="0" x="27px" width="110px"
     height="45px"
      style="fill: #C1FFFF"/>
    <foreignObject width="120px" y="0" x="20px">
      <div xmlns="http://www.w3.org/1999/xhtml">
        <ul>
          ..
        </ul>
      </div>
    </foreignObject>
    <foreignObject width="120px" y="50px"
        x="20px">
      <math xmlns="http://www.w3.org/1998/Math/
          MathML" style="color: red; font-size:
          24.000000pt; font-family: times">
        <mrow>
          ..
        </mrow>
      </math>
    </foreignObject>
  </svg>
</li>
<li>
  <math xmlns="http://www.w3.org/1998/Math/Math
      ML" style="color: blue;  font-size: 24.
      000000pt; font-family: times">
    <mrow>
      ..
    </mrow>
  </math>
</li>
</ul>
</html>
```

図6 XHTMLの中にSVGとMathMLを記述しSTYLE属性を指定

最後に、XHTMLの中にSVGとMathMLを記述し、さらに個々の要素の属性としてCSSのSTYLEを指定するという複合ソース（図6）を作成し、Mozilla、Amayaで表示を確認した。その結果を図7に示す。Mozillaでは、予想通りXHTML、SVG、MathMLとも表示されたが、スタイルの指定は無視された。が、

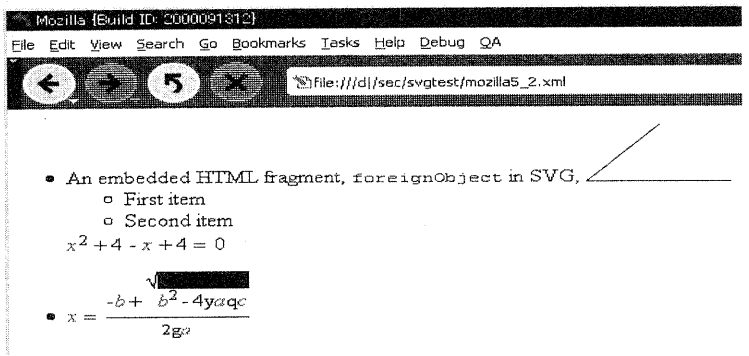

図7　3言語のスタイル付きソース(図6)の Mozilla による表示(文字の一部が赤色)

<mi> と </mi> で囲んだ中身が、斜体かつ赤く表示され、<msqrt> で指定したルート（√）記号が、黒塗りになって１行上に浮くなど、一部意図的、一部不具合ととれる表示となっている。Amaya でも３言語とも表示されたが埋め込み図形が "First item" "Second item" などと、開発途上の「工事中」を思わせる表示となった。

以上の結果を表１、表２にまとめる。表１は、コンテナ、すなわち外側言語がＳＶＧで、スタイルや、XHTML の埋め込み等の拡張を加えていったものと、処理系の対応状況のまとめである。表２は、コンテナが XHTML の場合に各種拡張を加えていったものと、処理系の対応状況のまとめである。後者については、より本格的な複合文書の可能性として、上記でとりあげた３種、２重までの言語の埋め込みだけでなく、３重、４重、５重、６重の深さまで多重に埋め込まれた複合文書を作成し、対応を調査した。

表１ コンテナがＳＶＧの各種複合文書と、処理系の対応状況

	Mozilla Milestone 18	W3C Amaya 4.1	Adobe SVG Viewer 2.0 b1	IBM SVG View 0.4a	Apache Batik SVG Toolkit 1.0b
外側の SVG の表示	○	○	○	○	○*1
SVG の style 有効	×	○	○	○	○
ForeignObject（○:認識される、△:無視、×:エラー）	△	○*2	△	△	×
内側の HTML の表示	○	×*3	×	×	×
外部の CSS 有効	×	×	○	×	○
その他	*4		*5		

表 2 コンテナが XHTML の各種複合文書と、処理系の対応状況

	Mozilla	Amaya	Adobe	IBM	Apache
外側の HTML の表示	○	○	−*0	−	−
内側の SVG の表示	○	○	−	−	−
SVG の style 有効	×	○	−	−	−
3 重以上の複合文書	○	○	−	−	−
HTML 内 MathML 表示	○ *6	○	−	−	−
SVG 内 MathML 表示	○ *6	○	−	−	−
HTML,SVG,MathML 共存	○	○	−	−	−
MathML の style 有効	×	○	−	−	−
MathML の CSS 有効 *7	×	×	−	−	−
その他	*4		−	−	−

注：　*0「−」は最初から処理系の仕様外の意味。 *1　foreignObjet タグを削除した場合。

*2　foreignObject タグ無しでも同じ動作。　*3　SVG が別の HTML に含まれた形なら表示ＯＫ。

*4　SVG に前空間「http://www.mozilla.org/keymaster/gatekeeper/there.is.only.svg」を指定する必要有り。

*5　対応規格が 2000 年 8 月 2 日版。DOCTYPE に最新の DTD を指定すると警告メッセージが表示される。

*6　MathML の一部の要素が正常に表示されない場合がある。　*7 言語仕様外のはず。

4 考察と今後の展望

幅広い文書交換、共同執筆・編集用途を意識して、XHTMLとSVGの組み合わせ方を中心に3種類のXML複合文書の実現方式を提案し、その内2つについて、既存の最新のブラウザやプラグインにより、複合文書らしいレイアウトで表示可能であることを確認した。特にXHTMLをコンテナとする方法(3)では、4重、5重と、XHTML、SVGが互いに交代する多重の入れ子についても正しい表示を確認できた。このことはXMLらしい、比較的緩い規約による複合文書が軽い開発コストで早期に実現できることを確認したことを意味する。

調査の過程で修正履歴をとるためにAMAYAが持つアノテーション機能が使えるかの検討を行った。ここから想起されたのは、今後、グローバルに共同編集を行う際に、また、undo機能を実現するために、埋め込み文書ごとに履歴を管理する必要があるが、それを分散・局所管理するべきか、それともどこかに履歴サーバを置いて一括管理すべきか、等の議論である。おそらく、自動管理してundoを実現する目的では局所管理を、執筆者や編集人が見て構想を練り議論するためにはグローバル・リポジトリを、といった使い分けで、両方必要になるのではないだろうか。履歴やリソースの相関関係を管理するインフラの充実によって、OLE Compound Document [OLE2000]等の旧知のバイナリ複合文書規格を超える機能性が予感される。

スタイルシートをはじめとする付随的なリソースの管理についても適切なガイドラインが必要とされると考えられる。XHTMLに対するCSSでは音声の修飾もできる位であるから、近い将来、テキスト、音声、そしてベクトル図形の各オブジェクトに対して、詳細な配置や色指定等の修飾、各種効果で共通化できるものは共通化できるのが望ましいだろう。

今後は、上記、更新履歴記述スタックを含む書誌情報の埋め込み拡張や、Xlinkを駆使した複合文書内部リンクや外部との双方向リンク、さらに複合文書の構造を活かした文書検索のための拡張等を検討していきたい。また、本稿で提案した、XHTMLのドライバを作成することによる本格的なモジュール化を行う方法(1)や、XML Schemaを駆使するなどによる他の複合文書構成方法の考案と検証を行い、応用分野ごとの適不適を検証し、これらを踏まえて、さらなる得失の分析・比較を行い、複合文書規格の標準化を視野に入れていきたい。

参考文献：

[Lee2000] Tim Berners Lee: "Semantic Web," presentation at an XML Conference, Sept. 2000

[OLE2000] "OLE Documents," http://msdn.microsoft.com/library/books/inole/s12a2.htm ,2000

[SVG2000] "Scalable Vector Graphics (SVG) 1.0 Specification," http://www.w3.org/TR/SVG/ ,2000.11

[SVGimpl2000] http://www.w3.org/Graphics/SVG/SVG-Implementations, 2000.08

[XHTML2000] "XHTML™ 1.1 - Module-based XHTML," http://www.w3.org/TR/xhtml11/ , 2000.1

[野村 2000] 野村直之：「知の創出における XML の役割」,第2回広島セミナー, http://ph2.ed.hiroshima-u.ac.jp/xml/seminar/2-000925/index.html, 2000.9

[野村 2001] 野村直之：「ナレッジマネジメントツールの配備、実践動向と次世代技術」, 人工知能学会誌 2001.1, pp.39-41

調査ツール：

Mozilla Milestone 18 (Release: 2000/10/12) mozilla-win32-M18-mathml-svg-xslt.exe :
http://www.mozilla.org/projects/svg/

W3C Amaya 4.1 (Release: 2000/11/23) : http://www.w3.org/Amaya/

Adobe SVG Viewer 2.0 beta 1 (Release: 2000/10) : http://www.adobe.com/svg/

IBM SVGView 0.4a (Release: 2000/04/26) : http://www.alphaworks.ibm.com/tech/svgview

Apache Batik SVG Tookit 1.0beta 付属の SVG Viewer (Release: 2000/12/01) : http://xml.apache.org/batik/