法政大学学術機関リポジトリ
HOSEI UNIVERSITY REPOSITORY

PDF issue: 2025-07-06

XMLを用いた複数言語に対応するWebサイトの自動合成システム

山本, 昌弘 / Yamamoto, Masahiro

(出版者 / Publisher)
法政大学国際文化学部
(雑誌名 / Journal or Publication Title)
異文化. 論文編 / 異文化. 論文編
(巻 / Volume)
7
(開始ページ / Start Page)
79
(終了ページ / End Page)
95
(発行年 / Year)
2006-04-01
(URL)
https://doi.org/10.15002/00002960

Hosei University Repository

XMLを用いた複数言語に対応するWebサイトの自動合成システム

Synthesis System for Multiple Languages Web Sites by XML

山本昌弘
YAMAMOTO Masahiro

要約

Webベースの応用が激増している。Webサイトの情報は、HTMLやホームページビルダで書かなければならない。eビジネス分野では、同じフォーマットで、しかし、内容が異なる複数個のサイトをしばしば作成する。また、1つのサイトが複数個の言語で作られ、そのときは、個々に各言語に別々に作成される。このため、これらのWebサイトは個々に作成されるので、同じ内容にもかかわらず、スタイルは異ってくる。
本論分では、頻繁に使用される形式について、複数個の言語を可能にするWebサイトを自動生成するシステムについて論じる。
キーワード
ソフトウエア自動合成、Webベース応用ｼｽﾃﾑ、多言語Webサイト、XML,XSLT,DOM

1　序

Webベース応用の要求は急速に増加している。Webサイトを作成するには、

Hosei University Repository

Web情報はHTMLや通常のホームページビルダー（HP）を用いて作成するする必要がある。HTMLやHPビルダーを使用するには、ITの専門知識が必要で、Webサイトを作成する作業はITの専門家に依頼するのが通常である。

更に、eビジネス分野では、同じフォーマットで異なる内容のWebサイトを作成することがしばしばある。この場合は、各WebサイトはHTMLやHPビルダーを用いて別々に作成する。結果として、時間と費用がかかる。

更に、Webサイトは頻繁に複数個の言語、たとえば日本語と英語のサイトが作製される。この場合、各サイトは個々に各サイトに対応して作製される。Webサイトの保守も重複して行われる。このため、時間とコスト([1])がかかることになる。また、Webスタイルは、１つひとつ作製されるために、同じ内容にもかかわらず異なることになる。

これらの問題を解決するために、複数個の言語に対応して頻繁に使用される形式のWebサイトを生成するシステムを提案する。本合成システムは英語と日本語を入力できる通常のホームページを生成できる。又、本システムは同じ形式で英語と日本語のWebサイトを生成できる。

2　自動生成の手法

XML ([2])を用いると、複数言語に対応するWebサイトのデータ表現したり、抽出できる。一方、XSLT(eXtensible Stylesheet Language Transformations) ([3])を使用すると、eコマースのなどの頻繁に使用される応用のためのWebサイトを表現できる。しかし、単にXSLTを用いてWebサイトを表現すると、形式は静的な表現になる。そこで、Dynamic HTML (DHTML)の中でDOM (Document Object Model) ([4])を使用することが可能で、この場合は、英語と日本語のWebサイトを表現するためにXML文書の中から英語と日本語のデータを抽出できる。結果、同時に英語と日本語のWebサイトに対応するHTML文書を生成できる([5])。

Hosei University Repository

3　システムの概要

合成ｼｽﾃﾑは３つの機能、初期システム画面、Webデータ入力およびWebサイト合成から構成される。

3．1　初期ｼｽﾃﾑ画面

システムが開始すると、初期ｼｽﾃﾑ画面が生成される。この画面上で、サイト合成ボタンとWebサイト表示ボタンが存在する。もし、新しいWebサイトを生成したいときは、Webサイト合成ボタンを押す。結果を見たいときは、Webサイト表示ボタンを押す。

3．2　Webデータ入力

Webサイトへのデータの入力は通常のWebサイトと同様にテキスト入力やイメージ入力によって行われる。ここでは、例として、英語と日本語の本販売Webサイトを考える。

Webデータの入力のために、本名、出版社、価格、要約、本のイメージなどを入力するWeb入力画面が生成される。この画面上で、英語と日本語に対応する入力項目が記述される。このプロセスは、さらに本の入力をする場合は続けられる。

Webデータの入力後、入力データを含んだXMLファイルが生成される。このXMLファイルは各項目に対応する英語の部分と日本語の部分から構成される。

3．3　Webサイト合成

１つのWebサイトの表現スタイルはXSLTで規定される。１つのWebサイトのデータはXML文書で表現される。Webサイトの合成と表示はDHTMLで記述することができる。DHTMLの表現の中で、合成プロセスはDOM (Document Object Model)で記述される。すなわち、XSLTに基づいて、DOMドキュメント関数がXML文書からHTML文書へ変換する。

Hosei University Repository

4　合成システムのソフトウエア構成

合成システムは、図1に示すように、5つの機能モジュール、初期ｼｽﾃﾑ画面、Webデータ入力, WebサイトXML生成, Webサイト合成および Webサイト表示から構成される。

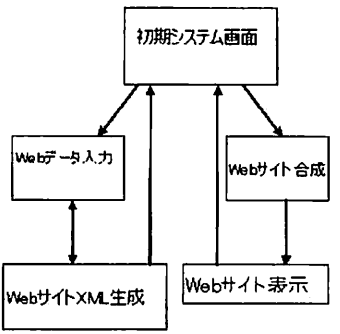

図１　合成システムのソフトウエア構成

4．1　初期ｼｽﾃﾑ画面

システムが起動すると、初期システム画面が図2のように生成される。画面上で、Ｗｅｂサイト合成ボタンおよびWeb サイト表示ボタンが生成される。

図2　初期システム画面

Hosei University Repository

4．2　Webデータ入力

Webサイト合成ボタンが初期ｼｽﾃﾑ画面で押されるとWebデータ入力画面が図3のように生成される。

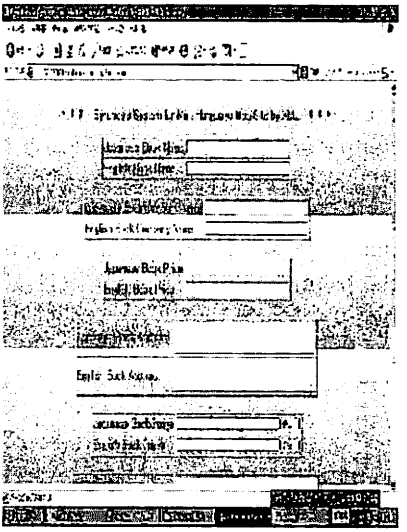

図3　Web データ入力

英語と日本語で本のいくつかの入力項目について入力できる。１つの本の入力が終了すると、継続ボタンまたは終了ボタンが押される。継続ボタンは、更に本の入力がある場合は押される。もしもう入力がない場合は、終了ボタンが押される。いずれの場合にもWebサイトXML生成が行われる。継続ボタンが押され、WebサイトXML生成が終了するとWebデータ入力が繰り返される。終了ボタンが押されると、初期ｼｽﾃﾑ画面へ戻る。

4．3　WebサイトXML生成

Webサイト生成は各本の入力ごとに行われる。制御変数ファイルが読みこまれる。そのデータから最初の本の入力かどうか決められる。もし、最初の本の入力の場合は図4に示す構造のXML文書の概要が作成され、<books> <book>...</book> </books>になる。入力データは図４のように、<book> と

Hosei University Repository

</book>の間にいれられる。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<books>
 <book>
  <name>
   <p xml:lang="ja">Perl and XML </p>
   <p xml:lang="en">Perl and XML </p>
  </name>
  <publisher>
   <p xml:lang="ja">オライリージャパン </p>
   <p xml:lang="en">O'REILLY </p>
  </publisher>
  <price>
   <p xml:lang="ja">￥3,360 </p>
   <p xml:lang="en">$35.00 </p>
  </price>
  <abstract>
  <p xml:lang="ja">プログラミング言語「Perl」と
XML（ExtensibleMarkupLanguage）データを組み合
わせて利用する 方法を解説 </p>
  <p xml:lang="en">Introduction of methodogy to
integrate a programming language Perl and XML
（ExtensibleMarkupLanguage）data. </p>
  </abstract>
  <image>
  <p xml:lang="ja">C:\Program Files\Apache
Group\Apache\htdocs\xmlpro\perlxml-j.bmp </p>
  <p xml:lang="en">C:\Program Files\Apache
Group\Apache\htdocs\xmlpro\perlxml-e.bmp </p>
  </image>
  <comments>
  <p xml:lang="ja">入門的な内容からやや専門的な技
術を紹介し、教科書として有用である。 </p>
  <p xml:lang="en">Introduce from elementary tech-
```

Hosei University Repository

```
nique to advanced method.It is useful as a textbook.
</p>
  </comments>
 </book>
 <book>
  <name>
   <p xml:lang="ja">Java and XML </p>
   <p xml:lang="en">Java and XML </p>
  </name>
  <publisher>
   <p xml:lang="ja">オライリージャパン </p>
   <p xml:lang="en">O'REILLY </p>
  </publisher>
  <price>
   <p xml:lang="ja">\4,830  </p>
   <p xml:lang="en">$34.95 </p>
  </price>
  <abstract>
  <p xml:lang="ja">ribune、Cocoon、EJBossなど
JavaとXMLの連携事例の多くに携わる著者による名著
の待望の翻訳版です。 </p>
   <p xml:lang="en">The book offers complete docu-
mentation of all features in both the Sun
Microsystems JAXB API and popular open source
alternative implementations  </p>
   </abstract>
   <image>
   <p xml:lang="ja">C:\Program Files\Apache
Group\Apache\htdocs\xmlpro\javaxml-j.bmp </p>
   <p xml:lang="en">C:\Program Files\Apache
Group\Apache\htdocs\xmlpro\javaxml-e.bmp </p>
   </image>
   <comments>
   <p xml:lang="ja">JavaとXMLそれぞれの解説を行う
のではなく、XMLを使ってWebアプリケーションの開
発を行うJavaプログラマを対象にしています。 </p>
```

Hosei University Repository

```
<p xml:lang="en">This new title provides an in-
depth technical look at XML Data Binding. The book
offers complete documentation of all features in
both the Sun Microsystems JAXB API and popular
open source alternative implementations  </p>
  </comments>
 </book>
</books>
```

図4　XML ファイの構成

もし、最初でない場合は、<book> と </book>間のデータが、<books> と </books>の間に挿入される。Webデータ入力で継続ボタンが押され、WebサイトXML生成が終了するとシステムはWebサイト生成へ戻り.新しいXMLデータが入力される。Webデータ入力で終了ボタンが押され、WebサイトXMLが終了するとシステムは初期システム画面へ戻る。

4．4　Webサイト合成

初期システム画面でWebサイト表示ボタンを押すとWebサイト合成が開始する。システムは最初に図5に示すWebサイト形式を規定するXSLTデータを読み込む。XSLTでは、英語か日本語かを定義する変数がある。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
<xsl:output method="html" encoding="Shift_JIS" />
<xsl:variable name="lang">ja</xsl:variable>
<xsl:attribute-set name="imageSize">
  <xsl:attribute name="width">120</xsl:attribute>
```

Hosei University Repository

```
  <xsl:attribute name="height">160</xsl:attribute>
</xsl:attribute-set>
  <xsl:template match="books">
<html>
<head>
<title><xsl:value-of
select="book/name/p[lang($lang)]" /></title>
</head>
<body bgcolor="yellow">
   <xsl:choose>
     <xsl:when test="$lang = 'ja'">
          <input type="button" value="英語を標準にす
る"
onclick="document.location.href='transfer.html?lang
=en'" />
     </xsl:when>
    <xsl:when test="$lang = 'en'">
          <input type="button" value="Japanese
Default"
onclick="document.location.href='transfer.html?lang
=ja'" />
    </xsl:when>
   </xsl:choose>
<hr />
<table border="1" >
<tr>
<th>
   <xsl:choose>
     <xsl:when test="$lang = 'ja'">
          <center><xsl:text>  日本語書名
</xsl:text></center>
     </xsl:when>
     <xsl:when test="$lang = 'en'">
          <center><xsl:text> English Book Name
          </xsl:text></center>
          </xsl:when>
```

Hosei University Repository

```
    </xsl:choose>
</th>
<th>
    <xsl:choose>
      <xsl:when test="$lang = 'ja'">
        <center><xsl:text> 出版社 </xsl:text></cen-
ter>
      </xsl:when>
      <xsl:when test="$lang = 'en'">
        <center><xsl:text> Publisher
</xsl:text></center>
      </xsl:when>
    </xsl:choose>
</th>
<th>
    <xsl:choose>
      <xsl:when test="$lang = 'ja'">
        <center><xsl:text> 価格 </xsl:text></cen-
ter>
      </xsl:when>
      <xsl:when test="$lang = 'en'">
        <center><xsl:text> Price </xsl:text></cen-
ter>
      </xsl:when>
    </xsl:choose>
</th>
<th>
    <xsl:choose>
        <xsl:when test="$lang = 'ja'">
        <center><xsl:text> 概要 </xsl:text></cen-
ter>
        </xsl:when>
        <xsl:when test="$lang = 'en'">
        <center><xsl:text> Abstract
</xsl:text></center>
        </xsl:when>
```

Hosei University Repository

```
    </xsl:choose>
</th>
<th>
    <xsl:choose>
        <xsl:when test="$lang = 'ja'">
          <center>        <xsl:text> 本の表紙
</xsl:text> </center>
        </xsl:when>
        <xsl:when test="$lang = 'en'">
          <center>        <xsl:text> Picture of Book
              </xsl:text></center>
        </xsl:when>
    </xsl:choose>
</th>
<th>
    <xsl:choose>
        <xsl:when test="$lang = 'ja'">
        <center><xsl:text> コメント
</xsl:text></center>
        </xsl:when>
        <xsl:when test="$lang = 'en'">
        <center><xsl:text> Comments
</xsl:text></center>
        </xsl:when>
    </xsl:choose>
</th>
</tr>
<xsl:for-each select="book">
  <tr>
  <td><xsl:value-of select="name/p[lang($lang)]"
/></td>
  <td><xsl:value-of select="publisher/p[lang($lang)]"
/></td>
  <td><xsl:value-of select="price/p[lang($lang)]"
/></td>
  <td>  <xsl:value-of select="abstract/p[lang($lang)]"
```

Hosei University Repository

```
/></td>
  <td> <img xsl:use-attribute-sets="imageSize">
   <xsl:attribute name="src">
    <xsl:choose>
          <xsl:when test="$lang = 'ja'">
         <xsl:value-of select="image/p[lang($lang)]" />
          </xsl:when>
          <xsl:when test="$lang = 'en'">
           <xsl:value-of select="image/p[lang($lang)]"
/>
        </xsl:when>
    </xsl:choose>
  </xsl:attribute>
  </img>  </td>
  <td>  <xsl:value-of
select="comments/p[lang($lang)]" /></td>
  </tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

図5　XSLTの構成

Hosei University Repository

また、WebサイトXML生成で作成されたXML文書および図6に示すWebサイトの基礎となるDHTMLを読み込む。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="ja" lang="ja">
<head>
<meta http-equiv="Content-Type"
content="text/html;charset=Shift_JIS" />
</head>
<body>
<div id="disp"></div>
<script language="JavaScript">
var objDoc=new ActiveXObject("Microsoft.XML-
DOM");
objDoc.async=false;
objDoc.load("xmlgen.xml");
var objStl=new ActiveXObject("Microsoft.XML-
DOM");
objStl.async=false;
if(location.search==""){
        var strCookie=document.cookie + ";";
        var intStart=strCookie.indexOf("lang=");
        if(intStart!=-1){

intEnd=strCookie.indexOf(";",intStart);
                strLang=unescape(strCookie.sub-
string(intStart + 5,intEnd));
        }else{
                strLang="ja";
        }
```

Hosei University Repository

```
}else{
        aryLang=location.search.split("=");
        strLang=aryLang[1];
}
var objToday=new Date();
var objExpires=new
Date(objToday.getFullYear()+5,objToday.getMonth(),
objToday.getDate());
document.cookie="lang=" + escape(strLang) +
";expires=" + objExpires.toGMTString();
objStl.load("ja_en.xsl");
var nodVar=objStl.selectSingleNode("//xsl:variable");
nodVar.text=strLang;
document.write(objDoc.transformNode(objStl));
</script>
```

図6　DHTMLの構成

最初、DHTMLが実行され、Webサイトから送られたクッキーによって英語か日本語が選ばれる。最初に、英語のWebサイトが表示される。クッキーから取られた情報はXSLT文書のタグ変数に保存される。DOM変換命令を用いてXML文書はXSLT文書に従ってHTML文書に変換される。英語のHTMLまたは日本語のHTMLがXSLT文書のタグ変数に従って生成される。

Hosei University Repository

4．5　Webサイト表示

Webサイト表示はWebサイト合成のあと実行される。Webサイト表示では、合成されたHTML文書が実行され、Webサイト表示は、図7aと図7bに示されるようにそれぞれ英語と日本語で表示される。英語と日本語のサイトをWebサイト表示画面の選択ボタンを変えることでみることができる。

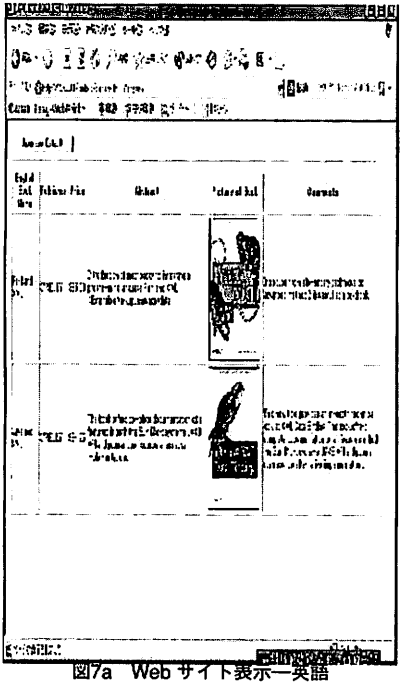

図7a　Web サイト表示―英語

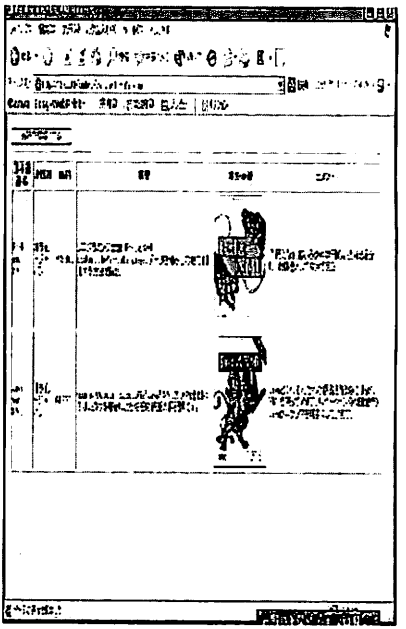

図7a　Web サイト表示―英語

Hosei University Repository

5　システム構成

合成システムは、図8に示すように、ブラウザ, WWWサーバ、合成ソフトウエア、データベースおよびファイルシステムから構成される。

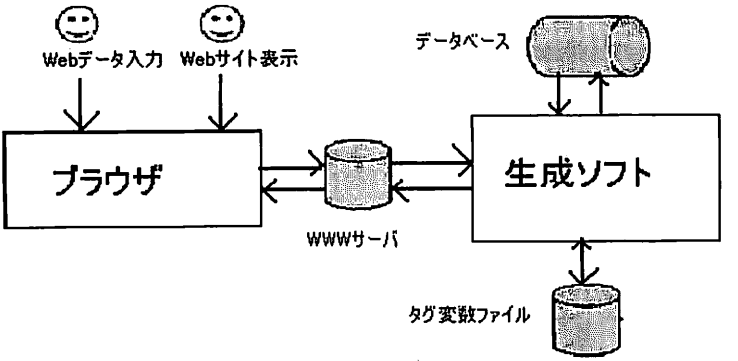

図8　システム構成

ブラウザ：ブラウザは、初期システム画面、Webデータ入力、Webサイト表示を表示するために使用される。通常のブラウザで、たとえば、インターネットエクスプローラである。

WWWサーバ：WWWサーバは、合成ソフトウエア、入力データのデータベースを保存する。ここではApache Active Serverが使用される。

合成ソフトウエア：合成ソフトウエアは、システムの中核で、すでに述べた5つの機能モジュールから構成される。これは、Script言語であるPerl([6])で書かれている。

データベース：データベースは、Webサイトの入力データを保存するもので、たとえば、MySQLのような関係データベースが使用されるファイルシステム：ファイルシステムは、英語か日本かを規定するタグ変数を保存する。通常のテキストファイルが使用される。

Hosei University Repository

6　結論と将来課題

たとえば書籍販売のようなeコマースのサイトはこのシステムを用いて容易に作成できる。複数言語、たとえば英語や日本語、の同じ形式のWebサイトを同時に生成される。結果、実現時間や実現費用が劇的に減少する。本論分では、例題として、英語と日本語の本販売Webサイトが示された。しかし、他のWebサイトや他の言語に拡張できる。

このシステムは、プロトタイプシステムである。ユーザーインタフェースは実用化のためには改善する必要がある。

参考文献

[1] S.B. Essanaa & N. Lammari, RetroWeb:A Web Site Reverse Engineering Approach. Proc. 4 th International Conference of Web Engineering, Munich, 2004, 306-310.

[2] M. Riehl, XML and Perl (Indianapolis, IN: New Riders Pub, 2002).

[3] M. Key, XSLT Programmer's References, 2nd Edition (Hoboken, NJ: Wiley Publishing Inc. , 2003).

[4] J.Marini,Document Object Model;Processing Structured Documents (New York, NY: McGraw-Hill, 2002).

[5] B. Thalheim, K.D. Schewe, I. Romalis, T. Raak, & G.Fiedler, Website Modeling and Website Generation. Proc. 4 th International Conference of Web Engineering , Munich, 2004, 577-578.

[6] C. Ferm, Intro to Perl for CGI (Sebastpol, CA: O'Reilly & Associates , 1998).