# XMLを使った北西セム語のタグ付け

竹内　茂夫（京都産業大学）

キーワード： XML、SGML、HTML、北西セム語

## 1　はじめに

今年度から始まった科学研究費補助金基盤研究 (C) の研究課題「前 2-1 千年紀における北西セム語の等語線の再画定：GIS による言語地理学的研究」(研究代表者：池田潤) では、データのタグ付けの言語として XML (Extensible Markup Language) を採用することとした。

本稿では、XML とは何かということについての概要を示し、なぜ XML を用いて北西セム語の文法タグ付けをするのかということと、その試みの実例を示す。

## 2　XML への道筋

XML は、SGML (Standard Generalized Markup Language) という文書の電子化のための規格の簡易版で、「SGML Lite」とも言える直系の子孫である。SGML から派生したもう 1 つの子孫が、HTML (HyperText Markup Language) である。XML がなぜ出てくることになったのかを見るために、これらについて簡単に見ていきたい[1]。

### 2.1　SGML

SGML は、1986 年に ISO (International Organization for Standardization 国際標準化機構) によって標準規格として定められた (ISO 8879:1986、JIS X

[1]第 2 章の記述は、竹内 (2000:242-47) に大幅に修正・加筆したものである。

4151:1992) [2]。SGML は、既に米国国防総省、EC の公式出版事務局、製造業ではボーイング社の電子マニュアル、日本でも厚生省への新薬申請のデータ形式として使われてはいた。しかしながら、仕様が複雑で処理系の開発が難しく、また SGML 文書の処理が重かったようである。

### 2.1.1 SGML の構造

SGML は、文書の見栄えではなくて論理構造を記述するもので、以下のものから成り立っている (Raggett, Le Hors and Jacob 2000 より)。

1. SGML 宣言。SGML 宣言は、アプリケーションに出現し得る文字と区切り子とを定める。
2. 文書型定義 (DTD)。 DTD は、マーク付け構成素のシンタクスを定義する。DTD には、文字実体参照などの追加的定義が含まれ得る。
3. マーク付けに反映されるべきセマンティクスについて説明する規定。この規定は、DTD 中では表現できないシンタクス上の制約をも科す。
4. データ (内容) とマーク付けとを含む、文書インスタンス。どのインスタンスも、解釈のために用いる DTD への参照を含む。

1 の「SGML 宣言文」は、SGML 文書がどのような基準に従って書かれているかを示すものである。後述の HTML も SGML から派生しているものであり、例えば HTML 4 も SGML 宣言文を持っている (参照：Raggett, Le Hors and Jacob 1999: http://www.w3.org/TR/html4/sgml/sgmldecl.html)。例として、その冒頭部分だけを示す。

```
<!SGML "ISO 8879:1986 (WWW)"
   --
     SGML Declaration for HyperText Markup Language version HTML 4

     With support for the first 17 planes of ISO 10646 and
     increased limits for tag and literal lengths etc.
   --
```

[2]元々は米 IBM で 1963 年に開発された GML にさかのぼるもので、開発者の Goldfarb, Mosher, Lorie の頭文字を取って名付けられた。「Generalized Markup Language」というのは後付けとのことである (倉元 1997)。

```
CHARSET
  BASESET "ISO Registration Number 177//CHARSET
           ISO/IEC 10646-1:1993 UCS-4 with
           implementation level 3//ESC 2/5 2/15 4/6"
```

次に、2 の「文書型定義」には、どのような文字集合を使用するか (記号など)、どのようなタグや属性が使用できるか (特殊文字の表し方も)、タグの省略機構を利用できるか (タグ名の長さ、大文字と小文字の区別など) が定義されている。3 の「規定」とは、マークアップに帰すべき意味を記述したスペックである。最後の 4 の「文書インスタンス」とは、SGML 宣言や文書型定義に従って実際に書かれる文書 (タグとデータそのもの) である。

以下は、SGML を使って William Blake の詩をマークアップした文書インスタンスの例である (Sperberg-McQueen and Burnard 1994 http://www.isgmlug.org/sgmlhelp/g-sg13.htm より空行を削除した)。

```
<anthology>
   <poem><title>The SICK ROSE</title>
     <stanza>
       <line>O Rose thou art sick.</line>
       <line>The invisible worm,</line>
       <line>That flies in the night</line>
       <line>In the howling storm:</line>
     </stanza>
     <stanza>
       <line>Has found out thy bed</line>
       <line>Of crimson joy:</line>
       <line>And his dark secret love</line>
       <line>Does thy life destroy.</line>
     </stanza>
   </poem>
       <!-- more poems go here -->
 </anthology>
```

なお、単純化のために、終了タグ (</line>など) は、省略することができる (引用元のサイトでも、終了タグが省略された例が挙げられている)。

### 2.2 HTML

WWW で用いられている HTML は SGML から派生し「SGML アプリケーション」と呼ばれているが、こちらは今では SGML からは遠く離れてかなり独自の道を歩んでしまっている。HTML は、決められた要素 (それを定義する「タグ」) しか使うことができず、独自の要素を定義することはできない。反面、比較的簡単で覚えやすく、WWW ページが作りやすいために爆発的に普及して、今日の WWW ブームが築かれることとなった。

#### 2.2.1 HTML 略史

HTML の歴史を大まかに述べると (H-Hash 1999-2005, Raggett, Le Hors and Jacobs 2000)、元々スイスにある欧州原子核研究機構 (CERN) の Tim Berners-Lee が 1989 年に WWW を提案し、1993 年に NCSA (米国立スーパーコンピュータ応用センター) で Marc Andreessen らによって開発された画像も表示できるブラウザの Mosaic が解釈する HTML の仕様をまとめていった。そして、インターネットで利用される技術の標準化を策定する組織である IETF (Internet Engineering Task Force) のあるワーキング・グループから提出されたドラフトが、「HTML 1.0」と呼ばれている。HTML 1.0 では、テキスト、リンク、画像などを含む文書の構造化が規定されている。

1995 年、「HTML 2.0」が IETF の HTML ワーキング・グループによって RFC (Request for Comments) 1866 として発表された。このヴァージョン以降、文書の冒頭に 2.2.2 で述べる文書型 (DOCTYPE) 宣言が入っている。また、現在の HTML の基礎と言える文書構造 (例：`html` 要素)、ブロック構造の要素 (例：`p` 要素)、リスト (例：`ul` 要素)、句のマークアップ (例：`em` 要素)、ハイパーリンク (例：`a` 要素)、インラインイメージ (例：`img` 要素)、物理的指定を示すもの (例：`b` 要素)、フォームが定義された。

ただし、規格の上では利用できる符号化文字集合 (いわゆる「文字セット」で、後述の 2.4 参照) が ISO-8859-1 (Latin-1) すなわち 1 バイト英数字に限られており、日本語などの 2 バイト文字が利用できなかった。そうした文字が利用できるように定められたものが 1997 年に出された RFC 2070 と言われるもので、そのように国際化された HTML 2.0 が「HTML 2.x」あるいは「HTML i18n (= internationalization)」である。

時代は前後するが、1994 年に W3C (World Wide Web Consortium) という WWW で使われる各種技術の標準化を推進するために設立された団体が発足し、次の HTML 3.2 勧告以降は IETF に代わって標準化を行っている。

HTML 3.2 勧告は、1997 年に IBM, Microsoft, Netscape Communications Corporation などとともに開発したものであるが、規格上は文字集合として ISO-8859-1 (Latin-1) しか利用できなかった。さらに、色の指定 (例：`bgcolor="white"`)、背景の指定 (例：`background="back.gif"`)、位置の指定 (例：`align="center"`)、サイズの指定 (例：`size="5"`)、テーブルの採用 (例：`<table>`)、フレームの採用などの視覚的情報に関する要素が加わったので、一般的にはよく使われるようになったものの、本来の文書化構造という主旨から外れることとなった。

そうしたことから、新たな規格として 1997 年に HTML 4.0 勧告が、1999 年にはそれを修正した HTML 4.01 勧告が出された。現在「HTML」といえば、この 4.01 を指す。総称して HTML 4 とするが、HTML 4 では、1) 国際化が図られて日本語も利用できる符号化文字集合 (Unicode を含む ISO/IEC 10646-1:1993 UCS-4) を採用し、2) 画像以外にも音声や動画などのデータを統一的に扱うために `object` 要素を定め、3) スタイルシート (CSS として定められる) によって視覚的情報を定義して HTML 本体は文書の構造化に絞り、4) 一般的なブラウザ以外のユーザにもアクセスしやすいドキュメントをサポートした。HTML 4 では、3.2 からの移行を踏まえ、視覚的情報を含む/含まないなどの用途に応じた 3 つのスキーマ、すなわち厳密型 (Strict)、移行型 (Transitional)、フレーム設定型 (Frameset) が存在し、文書冒頭の文書型宣言の中に明示されなければならない (2.2.2 参照)。

こうして発展していった HTML だが、マークアップする規則が緩いために、要素を書く際開始タグはあっても終了タグがなかったり (例えば `<p>` のみで `</p>` がない)、不規則な入れ子関係 (`<p>`here is an emphasized `<em>`paragraph.`</p></em>`) で書かれることも多かった。そうした HTML 文書であってもブラウザが適当に解釈して表示できるようになっているが、コンピュータで処理をさせる場合には都合が良くなかった。

そこで、HTML 4.01 を本主題の XML で再定義したものとして、XHTML (Extensible HyperText Markup Language) が登場した。XHTML は、2000 年

1 月に 1.0 勧告が、2002 年 8 月に改訂されて 1.0 第 2 版が出された。2001 年 5 月には 1.1 勧告が出されて、要素と属性を関連したグループ毎に分割してモジュール化し、それまで 3 つあったスキーマを厳密型のみにした。現在、2.0 が 7 月 26 日付けの第 8 版公開ワーキング・ドラフトすなわち議論の途上にある。1.0 勧告が出された同年、携帯電話などの小さな端末向けに必要最小限のモジュールだけで構成した XHTML Basic も出された。

HTML との違いは 2.2.2 でも例示するが、XHTML ではタグは全て小文字で書かなければならない (例えば<HTML>は認められない)、終了タグは必須 (例えば<p>だけは認められず、単独で現れる空要素は<br />のように書かなければならない) などがある (Pemberton 2000 参照)。

HTML が SGML の直系である XML によって XHTML として再定義されたことによって、SGML から遠く離れていた HTML が SGML の方に回帰してきたと言えよう。

### 2.2.2 HTML の構造

HTML は次の 2 つから成る。

1. 文書型宣言
2. 文書インスタンス

文書型宣言とは、当該文書の文書型定義の名前を示すもので、上述のように HTML 2.0 から入るようになった。HTML 4.01 では 3 つの文書型定義すなわち 2.2.1 の HTML 4.01 のところで述べた厳密型、移行型、フレーム設定型を規定しており、HTML 文書を作る際にはそのいずれかを含めて文書型宣言をしなければならない。

文書インスタンスは、タグとデータそのものである。

これらを含め HTML 4.01 の厳密型文書型宣言を用いた短い文書の例は、次の通りである (Raggett, Le Hors and Jacobs 1999 http://www.w3.org/TR/html4/intro/sgmltut.html より)。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
```

```
    <HEAD>
        <TITLE>My first HTML document</TITLE>
    </HEAD>
    <BODY>
        <P>Hello world!
    </BODY>
</HTML>
```

これを XHTML (1.0) の厳密型で書き直すと次の通りになる。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja"
lang="ja">
    <head>
        <title>My first HTML document</title>
    </head>
    <body>
        <P>Hello world!</p>
    </body>
</html>
```

HTML と XHTML の違いは、2.2.1 でも述べたように、<html>以降のタグは小文字であり、 タグは で閉じられている。また、1 行目には後の 2.4 で述べる XML 宣言が入り、2 行目の html 要素の中に属性として xmlns=といった名前空間 (2.4 参照) と xml:lang のように言語の指定がされていることで、XML で定義されていることがわかる。

### 2.3 XML

本稿の主題であり、HTML を XHTML に再定義する際に用いられた XML は、WWW のための新しい文書記述言語である。XML は、1996 年にワーキング・ドラフトが出され、1998 年 2 月 10 日に W3C によって 1.0 勧告として制定された。その後、2000 年 10 月に 1.0 第 2 版勧告が、2004 年 2 月に 1.0 第 3 版勧告が、2006 年 8 月 16 日に 1.0 第 4 版勧告が出され、これが現在のところ最新で国際標準ともなっている (Bray, Paoli, Sperberg-McQueen, Maler and Yergeau 2006)。1.1 は 2001 年 12 月にワーキング・ドラフトとし

て発表され、2004 年 2 月に 1.1 勧告となった。その後、1.1 第 2 版勧告が 2006 年 8 月に発表され、これが今のところ最新版である。

XML の利用例として、Apple 社の OS である Mac OS X では初期設定ファイル (.plist ファイル) が XML で書かれており、10.4 「Tiger」においては XML 形式で保存された MS Word 文書 (WordML) をテキストエディットというアプリケーションに取り込むことができる。Microsoft 社の Office では、XP のヴァージョンから XML 形式への対応を始め、2003 で XML スキーマ (後述) がサポートされ、ヴァージョン 2007 ではデフォルトの保存方式が XML となるとのことである (Microsoft Office Open XML Formats)。また、MS Office と互換性のあるオープン・ソースの OpenOffice は、当初より XML ベースの保存形式を標準としている。

XML の設計目標は次の通りである (Bray, Paoli and Sperberg-McQueen 1998)。

1. XML は、インターネット上でそのまま使用できる。
2. XML は、広範囲のアプリケーションを支援する。
3. XML は、SGML と互換性をもつ。
4. XML 文書を処理するプログラムは容易に書ける。
5. XML では、オプションの機能はできるだけ少なくし、理想的には一つも存在しない。
6. XML 文書は、人間にとって読みやすく、十分に理解しやすい。
7. XML の設計は、すみやかに行う。
8. XML の設計は、厳密で、しかも簡潔なものとする。
9. XML 文書は、容易に作成できる。
10. XML では、マーク付けの数を減らすことは重要ではない。

本プロジェクトで XML を採用したのも、HTML と違ってタグを自由に設定できるということの他に、特に 1[3]、2、4、6、9 の設計目標があるために他ならない。

### 2.4 XML の構造

XML は次の要素から構成されている。

1. XML 宣言
2. XML インスタンス
3. 文書型定義

XML 宣言はあった方が望ましいもので、XML 文書の先頭に記述されて XML であることを宣言し、使用する XML のヴァージョン、使用する文字符号化処理 (方式) などを宣言する。

```
<?xml version="1.0" encoding="UTF-8"?>
```

この宣言では、XML のヴァージョンが 1.0 であり、文字符号化処理として UTF-8 を採用することが宣言されている。

注意が必要なのは、この UTF-8 とよく耳にするようになった Unicode というのは等価ではないということである。Unicode とは符号化文字集合 (Character Set いわゆる「文字セット」) であり、コンピュータ上で多言語の文字を 1 つの文字符号化処理 (後述) で取り扱うために提唱された文字の集合のことである。他に、日本語で使われる文字を網羅した JIS X 0208:1997 も符号化文字集合である[4]。一方、UTF-8 とは文字符号化処理 (Character Encoding Scheme いわゆる「文字コード」) であって、符号化文字集合中のある文字 (例えば「A」、「あ」) を文字コード (16 進法表示で「0x41」や

[3]ただし、XML 文書をブラウザでそのまま表示してもあまり有用ではないので、実際には XSL (Extensible Stylesheet Language 拡張可能なスタイルシート言語) を用いて、より理解しやすい形式に変換したり組版を行ったりする。

[4]符号化文字集合のうち、Windows 符号化文字集合と Macintosh 符号化文字集合は、JIS X 0201 と JIS X 0208 を共通の文字集合として持っているが、それに追加されている文字の配置が若干異なるために文字化けの原因となる。

「0xA4 0xA2」) に対応させるもので、UTF-8 は Unicode という符号化文字集合で使えるものである。他に、主に電子メールで使われる ISO-2022-JP (いわゆる「JIS コード」) も文字符号化処理の 1 つである。

次に、XML インスタンスは、XML 文書の本体、すなわちタグでマークアップされたデータ群である。一例として、国文学研究資料館によってテキストデータとして電子化された古今和歌集に、簡単な XML のタグ付けを行った文書が公開されているので (FXIS 1997-2006: http://www.fxis.co.jp/xmlcafe/link/jirei/sample_kokin.html)、そこから序文と本文の一部を若干整形して引用する。必要に応じたタグが自由に定義されていることが、よくわかる。

```
<?xml version="1.0" encoding="UTF-8"?>
<集>
  <集名>古今和歌集</集名>
  <序文>
  やまとうたは
  [中略]
  その所にやいろの雲のたつを見てよみ給へるなり、
    <歌への参照 番号="N01112"></歌への参照>
  [中略]
  </序文>
  <巻>
    <巻名>古今和歌集巻第一</巻名>
      <部立て>
        <部立て名>春歌上</部立て名>
      <歌 番号="N00001">
        <詞書>ふるとしに春たちける日よめる</詞書>
        <詞書の読み>ふるとしにはるたちけるひよめる</詞書の読み>
        <作者>在原元方</作者>
        <作者標準名>元方</作者標準名>
        <和歌>
          <区>年の内に</区><区>春はきにけり</区><区>一とせを</区>
          <区>こそとやいはん</区><区>ことしとやいはん</区>
        </和歌>
        <和歌の読み>
          <区>としのうちに</区><区>はるはきにけり</区>
          <区>ひととせを</区>
          <区>こそとやいはむ</区><区>ことしとやいはむ</区>
        </和歌の読み>
        <メモ>
          <新編国歌大観></新編国歌大観>
```

```
            <旧版国歌大観></旧版国歌大観>
          </メモ>
        </歌>
    [中略]
      </部立て>
    </巻>
    [中略]
</集>
```

このように、XML 文書の形式に従っている文書を「整形式 XML 文書 (well-formed XML document)」と呼ぶ。一方、整形式であることに加え、文書型定義のようなスキーマ言語 (後述) によって定義された文書構造に準じているかどうか、文法的に正しいかどうかを検証し表示する XML パーサと呼ばれるソフトウェアによって検証された文書を「妥当な XML 文書 (valid XML document)」と呼ぶ。

最後に、2.1.1 の SGML のところでも扱った文書型定義は、XML では必須ではない。文書型定義には次の欠点があるとされる (H-Hash 1999-2005)。

1. XML 名前空間 (後述) をサポートしていない。すなわち、複数の言語で同じ名前の要素があると、どの言語に属する要素なのかわからなくなる。

2. 文書型定義はデータ型を定義できない。すなわち、記号以外の文字しか指定できないので (後述)、独特なフォーマットを定義することができない。

3. 文書型定義は XML の書式になっていない。

これらの問題を解決するために、使用する XML の定義を XML そのもので行う「XML スキーマ」が W3C によって 2001 年 5 月に 1.0 として勧告され、多くの誤植を修正した 1.0 第 2 版が 2004 年 10 月に出された。しかしながら、開発に Microsoft 社や CommerceOne 社など多くの企業が関わっていることもあり、多くの機能を盛り込みすぎて仕様が複雑になっていると言われている。現在、1.1 の策定に向けて作業中のようである。

XML スキーマの簡単な例とその説明を引用して示す (小野 2003: http://www.atmarkit.co.jp/fxml/rensai2/schema01/schema01.html)。次のような文書インスタンスがあるとする。

```
<greeting>Hello World!!</greeting>
```

この文書インスタンスから、次の構造がわかる。

- ルート要素は、greeting。
- greeting 要素は属性を持たない。
- greeting 要素の内容は文字列。
- greeting 要素は子供要素を持たない。

これを XML スキーマで表すと (もしくは変換すると) 次のようになる。

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="greeting" type="xsd:string"/>
</xsd:schema>
```

1 行目は、XML スキーマは先に述べたように XML 文書なので、XML 宣言が現れる (ここでは文字符号化処理方式は表示されていない)。2 行目はルート要素の表示で、`xsd:schema` とは XML スキーマの schema 要素であることを示すもので、`xmlns:xsd=`とは要素名の最初に xsd と付いたら=以降の URI[5](ここでは http://www.w3.org/2001/XMLSchema) で定義された要素であるという名前空間宣言である (文書型定義には上述のようにこれがない)。これは 4 行目の終了タグで閉じる。3 行目が XML の構造を表すもので、`xsd:element` とは element 要素も XML スキーマ要素であること、`name="greeting"`は「greeting」という要素を持つ name 属性であること、`type="xsd:string"`はその要素の内容が文字列であって子供要素や属性がないことを示す。なお、これは空要素なので、タグそのもの

[5]URI (Uniform Resource Identifier) はインターネットに限らずリソース一般を指す概念で、一般に使われておりネットワーク上のリソースを指す URL (Uniform Resource Locator) よりも広く、W3C では URI を用いている。

が/>で終結していなければならない。詳しくは、小野 (2003) などを参照されたい。

これらを踏まえた上で、北西セム語を XML でマークアップする試みについて、次章で述べたい。

## 3 北西セム語を XML でマークアップする

北西セム語の資料を大きく分けると、前 2 千年紀の楔形文字資料と前 1 千年紀および少数の前 2 千年紀のアルファベット文字資料がある。北西セム語の前 2 千年期の楔形文字資料ではアマルナ書簡が主な資料となるため、必要な情報としては、文書名、出土地、年代、王の名前、言語名、ジャンル、書記名の他、ローマ字転写、音価、文字記号の転写、形態素情報であろう。前 1 千年紀のアルファベット資料では、文書名、出土地、年代、言語、ジャンルの他、ローマ字転写、形態素情報であろう。

この両者の言語資料を XML でマークアップするために、現時点では次の要素と属性を立てて作業を進めている。まず、ヘッダに入れる要素は次の通りである。(<header>から</header>の間)。

1. 文書名 (title 要素)
2. 出土地 (from 要素。ただし、書簡の場合別に発信地という項目を立てる必要があるかもしれない)
3. 年代 (date 要素)
4. 言語 (lang 要素)
5. 送信者または王名 (by 要素)
6. ジャンル (genre 要素)
7. 書記名 (scribe 要素)

次に、本体に入れる要素である (<body>から</body>の間)。

8. 何行目か (line_no 要素)

9. ローマ字転写 (tl 要素)
10. 楔形文字記号 (sign 要素。アルファベット文字ならローマ字転写と基本的に同じ)
11. 形態素 (morph 要素。属性として、品詞、性、数、人称など)

これらを XML でマークアップした文書インスタンスとして、ヘブライ語で書かれた前 10 世紀後半のゲゼル農事暦の最初の語を示す。

```
<corpus>
  <text>
    <header>
      <title>Gezer Calender</title>
      <from>Gezer</from>
      <date>the second half of the 10th century B.C.E.</date>
      <lang>Hebrew</lang>
      <by>anonymous</by>
      <genre>calender</genre>
      <scribe></scribe>
    </header>
    <body>
      <line>
        <line_no>1</line_no>
        <word>
          <tl>yrḥ-w</tl>
          <sign>yrḥ-w</sign>
          <morph class="N">yrḥ</morph>
          <morph class="PRON" person="3" number="SG"
          gender="M">-w</morph>
        </word>
      </line>
    </body>
  </text>
</corpus>
```

このテキストの文書型定義は、次の通りである (ADOS XML Studio 2.0.1 にて、文書インスタンスから変換)。

```
<!ELEMENT corpus (text)*>
<!ELEMENT text (header|body)*>
<!ELEMENT header (title|from|date|lang|by|genre|scribe)*>
```

```
<!ELEMENT title (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT lang (#PCDATA)>
<!ELEMENT by (#PCDATA)>
<!ELEMENT genre (#PCDATA)>
<!ELEMENT scribe EMPTY>
<!ELEMENT body (line)*>
<!ELEMENT line (line_no|word)*>
<!ELEMENT line_no (#PCDATA)>
<!ELEMENT word (tl|sign|morph)*>
<!ELEMENT tl (#PCDATA)>
<!ELEMENT sign (#PCDATA)>
<!ELEMENT morph EMPTY>
<!ATTLIST morph class CDATA #IMPLIED number CDATA #IMPLIED
person CDATA #IMPLIED>
```

この中で、#PCDATA (Parsed Character Data 構文解析対象文字データ) と記されている要素 (ELEMENT と記されている行) が、2.4 において「文書型定義が全ての文字しか指定できない」部分であって、記号は記号として解釈され、例えば「&」は「&」と書かないと表示されないし、マークアップに使われる「<」や「>」は使用できない。また、最後に現れる CDDATA (Character Data 文字データ) は記号も文字として扱う[6]。

次に、同じ文書インスタンスを上述 (2.4) の XML スキーマに変換したものを示しておく (拡張子は.xsd)。文書型定義に比べると、文書そのものが XML になっており、細かな定義がなされていることがわかる。非常に長いので、冒頭の header 要素の部分と、末尾の形態素を要素と属性によって定義している部分のみ示す。

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <xsd:annotation>
     <xsd:documentation>
     XML Studio Model generator ver 0.1
     </xsd:documentation>
```

[6] この他に、マークアップに使う「<」と「>」以外の記号は文字として扱う RCDATA (Replaceable Character Data 置換可能文字データ) がある。

```
</xsd:annotation>
<xsd:element name="corpus">
  <xsd:complexType>
    <xsd:sequence maxOccurs="unbounded" minOccurs="0">
      <xsd:choice maxOccurs="unbounded" minOccurs="0">
        <xsd:element ref="text"></xsd:element>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="text">
  <xsd:complexType>
    <xsd:sequence maxOccurs="unbounded" minOccurs="0">
      <xsd:choice maxOccurs="unbounded" minOccurs="0">
        <xsd:element ref="header"></xsd:element>
        <xsd:element ref="body"></xsd:element>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="header">
  <xsd:complexType>
    <xsd:sequence maxOccurs="unbounded" minOccurs="0">
      <xsd:choice maxOccurs="unbounded" minOccurs="0">
        <xsd:element ref="title"></xsd:element>
        <xsd:element ref="from"></xsd:element>
        <xsd:element ref="date"></xsd:element>
        <xsd:element ref="lang"></xsd:element>
        <xsd:element ref="by"></xsd:element>
        <xsd:element ref="genre"></xsd:element>
        <xsd:element ref="scribe"></xsd:element>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

ここまでが冒頭のヘッダ要素を定義した部分である。次が末尾の形態素の属性を定義している部分である。

```
<xsd:element name="morph">
  <xsd:complexType>
    <xsd:sequence maxOccurs="unbounded" minOccurs="0">
      <xsd:choice maxOccurs="unbounded" minOccurs="0">
```

```
        </xsd:choice>
      </xsd:sequence>
      <xsd:attribute name="class"></xsd:attribute>
      <xsd:attribute name="number"></xsd:attribute>
      <xsd:attribute name="person"></xsd:attribute>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

## 4 終わりに

本稿では、XML が登場するまでの流れを、その親でありフルセットの SGML、そして SGML をごく限られた要素に限定しながら独自の拡張を施してきた (X)HTML と関連させながら見てきた。

北西セム語をマークアップする際に XML を採用したのは、インターネット上でそのまま使用でき、広範囲のアプリケーションを支援し、XML 文書を処理するプログラムは容易に書け、XML 文書は容易に作成できる、などの設計目標があるからに他ならない。

(X)HTML もインターネット上でそのまま使用できるという点では同じであるが、要素や属性を自由に定義できないために、データベースとして扱ったりその後のデータ加工の点で難点がある。一方、XML と (X)HTML の親である SGML では、タグは自由に定義できるが、仕様が複雑な上にインターネットに対応していないという問題がある。

XML で北西セム語をマークアップする作業を行っていく間に、現時点では予測できない細かな調整や課題 (例えば要素や属性の増減や名前の修正) が発生するかもしれないが、一旦 XML でマークアップされたデータは様々な応用が可能なので、今後予定している GIS との連動も視野に入れながら作業を続けていく予定である。

## 【参照文献】

倉元靖史 (1997)『SGML & XML ガイド』日刊工業新聞社.

竹内茂夫 (2000)「はじめての HTML: WWW ページを作ろう」京都産業大学計算機センター教育研究システム課 (編)『コンピュータガイドーインターネット編ー』214-49. 京都産業大学.

(ウェブサイト)

Bray, Tim, Jean Paoli and C. M. Sperberg-McQueen (1998)『拡張可能なマーク付け言語 (XML) 1.0 W3C 勧告 1998 年 2 月 10 日』 http://www.fxis.co.jp/xmlcafe/tmp/rec-xml.html

Bray, Tim, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler and Francois Yergeau (2006) *Extensible Markup Language (XML) 1.0 (Fourth Edition).* http://www.w3.org/TR/2006/REC-xml-20060816

FXIS (1997-2006)『XML Café』http://www.fxis.co.jp/xmlcafe/

H-Hash (1999-2005) 『Studying HTTP. HTML & XML』 http://www.studyinghttp.net/markup

小野彩子 (2003)『連載 SE のための XML Schema 入門 (1) 簡単な XML Schema から始めよう』 http://www.atmarkit.co.jp/fxml/rensai2/schema01/schema01.html

Pemberton, Steven *et al*. (2000)『XHTML 1.0: 拡張可能ハイパーテキストマークアップ言語』http://www.doraneko.org/webauth/xhtml10/20000126/Overview.html

Raggett, Dave, Arnaud Le Hors and Ian Jacobs (1999) *HTML 4.01 Specification. W3C Recommendation 24 December 1999.* http://www.w3.org/TR/html4/

Raggett, Dave, Arnaud Le Hors and Ian Jacobs (2000) 『HTML 4.01 仕様書 1999 年 12 月 24 日付 W3C 勧告』http://www.asahi-net.or.jp/%7Esd5a-ucd/rec-html401j/

Sperberg-McQueen, C. M. and Lou Burnard (1994) *A Gentle Introduction to SGML*. http://www.isgmlug.org/sgmlhelp/g-index.htm

# Marking up Northwest Semitic Languages with XML

Shigeo TAKEUCHI

The aim of this paper is to show an example how to mark up some Northwest Semitic languages with XML (Extensible Markup Language). The XML is a W3C-recommended general-purpose markup language (among whose versions 1.0 Fourth Edition and XML 1.1 Second Edition are considered as current versions). It is a simplified subset of SGML (Standard Generalized Markup Language). In addition to allowing us to create any tags needed, the following design goals of XML will be helpful for us: XML will be usable on the Web as (X)HTML; it supports a wide variety of applications; and XML documents will be easy to create. That is why we have adopted XML when marking up Northwest Semitic languages.

*Faculty of Cultural Studies*
*Kyoto Sangyo University*
*Motoyama, Kamigamo, Kyoto 603-8555, Japan*
*E-mail: atake@cc.kyoto-su.ac.jp*