2009年度後期開講コンピュータ2D配布資料
SGML,XML,HTML

浅川伸一

2009年11月11日

1 SGML

SGMLとはStandard Generalized Markup Languageの略であり，国際規格 (ISO 8879) および日本工業規格 (JIS X 4151) として採用，制定されている。コンピューター間で文書を交換することを主な目的としている。

実際に文書を記述する際には，文書の構造を表すタグと，それに続く文を用いる。例えば，

<メール><件名>SGML</件名><著者>浅川伸一</著者><メールアドレス>asakawa@ieee.org</メールアドレス><今週の課題>SGMLとは何か調べよ</今週の課題><段落>SGML概説</段落>SGMLとはStandard Generalized Markup Languageの略であり...</メール>

などとすることで，

- メールの件名はSGMLであり
- 著者は浅川伸一
- メールアドレスはasakawa@ieee.org
- 今週の課題はSGMLとはなにか

といったことを伝達することになる。

もちろん，SGMLのデーターをやり取りする相互間で，メールは<メール>という開始タグを用いるということを取り決めておかねばならない。

タグの使用方法の合意ができれば，双方ともそのタグを使うことで受け取った文書の構造と内容は伝わることになる。

タグとは，

- 文の構造 (あるいは構成要素) を表す

- タグは文書を構成する各要素 (表題とか、章、節、注など) に対して定義される。
- タグには<要素名>の形の開始タグと</要素名>という/が追加された終了タグがある
- 文は開始タグと終了タグの間に挟まれる

といった規則がある。この辺は HTML と同様である。というより SGML の一変種が HTML なのである。

1.1 DTD

タグの定義は一般に DTD と呼ばれるファイルに記述される (正確には, DTD というファイルは存在せず, 前書き (prolog) と呼ばれるものの中に記述される)。基本的に、SGML で文書を作成する時は、

1. 作成する文書の DTD を選択する (無い場合は作成する)
2. 選択した DTD に定義されているタグを用いて文書作成する

という手順で作成する。
例えば

```
<!--メール DTD-- >
<!ELEMENT MAIL - - ((TO,FROM)?,DATE?,BODY) >
<!ELEMENT TO - O (#PCDATA) --宛先-->
<!ELEMENT FROM - O (#PCDATA) --発信-->
<!ELEMENT DATE - O (#PCDATA) --日付-->
<!ELEMENT BODY - - (P)* --本文-->
<!ELEMENT P - O (#PCDATA|Q)* --段落-->
<!ELEMENT Q - O (#PCDATA) --引用-- >
```

のように記述したとする。ここで DTD 内の –と–に挟まれた部分はコメントである。このとき、この DTD ではメールを以下のように定義したことになる。

1. <!ELEMENT MAIL - - ((TO,FROM)?,DATE?,BODY) >
   メールは要素 (ELEMENT)MAIL を持ち, その開始は<MAIL>, 終了には</MAIL>というタグを用いる
2. <MAIL>と</MEMO>は省略できない
3. <MEMO>と</MEMO>の間には, 要素として

- TO
- FROM
- DATE
- BODY

があり, TO と FROM は TO が先で FROM が来るという順で，1 つずつ，ペアで 0 回または 1 回のみ現れる (省略できる)

4. その次に DATE が現れる。DATE も 0 回または 1 回のみ現れる (省略できる)
5. 最後に BODY が現れる。BODY は 1 回のみ現れる (省略できない)

DTD に記述する宣言は、次の４つである。

1. 要素型宣言
2. 属性リスト宣言
3. エンティティ宣言
4. 記法宣言

ここでは，要素型宣言と属性リスト宣言について説明する。

1.2 要素型宣言

要素型宣言は，文書中で使用できる文書要素を定義する。要素型宣言では要素に関して，要素名，要素の親子関係，要素の内容モデル（子供の要素の出現の仕方）をそれぞれ指定する。要素型宣言は次のように書く。

`<!ELEMENT 要素名 内容モデル>`

例えばテキストを内容として持つ「名称」という要素は次のように宣言する。`<!ELEMENT 名称 (#PCDATA)>` ここで、`#PCDATA` はテキストを表すキーワードである。1 つの要素型宣言は，複数のレベルにわたる階層構造の 1 つのレベルを定める局所的なものである。これらを順に積み重ねることによって，データ全体の階層構造を作る。階層構造は親子関係で成り立っている。最上位の要素からはじめて，子供と孫，孫と曾孫といったように要素型宣言を重ねていく。内容モデルには、子どもの要素としてどのようなものの出現を許すかという指定を書く。このとき、子供として出現する要素に関する情報は、出現順序と出現回数の２つである。

1.3 レイアウト

タグを付けることによって文書の構造は伝わるが，文書を作成する場合にはレイアウトを指定したい場合もあるだろう。しかし，SGML 文書では

- レイアウト情報は、同じ分類の文書でも異なる場合がある
- 印刷時のレイアウト情報などはアプリケーション固有のものである

などの理由のために，特定アプリケーション固有の情報は文書内には含めないことが基本となる。このあたりは，HTML 文書では基本的に文章の内容と構造のみを記述し，レイアウトに関する情報はカスケードスタイルシート css として分離しておくことに似ている。

2 XML

XML(eXtensible Markup Language) は SGML のサブセットであり，1998 年 2 月に発表されたインターネット上で扱うデータを記述するためのデータフォーマットである。

近年，SGML の応用言語である HTML を利用した Web が普及し，文書閲覧のみならず，電子商取引という形で利用されるにいたった。しかし同時に HTML の限界が認識されるようになった。すなわち，HTML ではタグ名が固定されており，ユーザが自由にタグを拡張して使用することはできない。このようなことから SGML を Web 上で使えるようにしようという議論が始まった。

この授業で HTML を学習した際に，HTML を書く場合には，テンプレートフィアルをコピーして使うように指示した。このテンプレートファイルの先頭部分を見ると

```
<?xml version="1.0" encoding="iso-2022-jp"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/x
html1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja">
```

などと書いてある。つまりこのテンプレートファイルの一行目ではこの文書が XML の規格に従うことを宣言していたのである。

2.1 XML の長所

SGML は特定のソフトウェアに依存しないデータ形式である。SGML と HTML から多く点を引き継いでいる。

HTML からは

- 読みやすく簡単なデータ形式
- ハイパーリンク
- Web 技術（URL、MIME、HTTP など）に対応
- プログラムからの操作が容易

一方 SGML からは次のような長所を引き継いでいる

- タグを自由に定義することができる
- 強力なスタイルシートの使用が可能
- 厳密な文法チェックが可能であり、プログラムからの文書処理が容易

2.2 拡張可能と意味情報

先にも述べたとおり XML は HTML と同じく SGML の流れをくむデータ記述言語である。XML は HTML と比較すると、「拡張可能」であり、「意味情報」を含めることができる。例えば以下のような XML ファイルがあるとする

```
<?xml version = "1.0" encoding = "iso-2022-jp"?>
<documentinfo type = "XML">
<title>XML の簡単な解説</title>
<doctype>メール</doctype>
<author>浅川伸一</author>
</documentinfo>
```

HTML ではタグ名は決められたものしか用いることはできない。HTML 文書を作成するものはその内容をブラウザに表示させることによって確認し，さらにこの授業では Opera を用いて文法のチェックを行っていた。つまり，HTML のタグ名は表現情報を表しており、タグによってマークアップされている情報の意味は人間が実際にブラウザで見て初めて確認できるのである。

一方、同じ情報を XML でマークアップすることにすると，XML ではタグ名を自由に決めることができるため，目的に応じたタグを定義して使用することができるようになる。たとえば，<doctype>メール</doctype> のような場合”メール”という文字列の意味をタグ名が与えている。つまり”メール”という情報についての意味情報をタグ名が与えているのである。この行を見れば人間は”メール”という情報が文書のタイプを表すものであることを，タグ名から知ることができるのである。

このような場合「タグが意味情報をもつ」と考える。このように XML を使用してマークアップすることで，タグ名を HTML のように表現情報として利用するのではなく，意味情報として利用することになる。

この違いはプログラムからデータを処理する際に大きな違いとなって現れる。データを XML でマークアップしておけば、個々の情報に意味情報のタグがついているのでプログラムから正確にデータ処理をすることが可能になるのである。

2.3 文書における「内容」「構造」「体裁」の分離

文書を構成する「内容」「構造」「体裁」という 3 つの要素は，従来の紙媒体ではひとつとなって決定されている。XML ではこれら 3 つの要素を分離して扱う。「内容」は XML 文書内でマークアップされるが、その XML 文書の「構造」は DTD（Document Type Definition）と呼ばれる構造定義体によって定義する。また，XML 文書に与える「体裁」はスタイルシートにより記述する。DTD の概念は SGML から引き継がれたものであり，スタイルシートの概念は HTML から引き継がれた。

－ XML 宣言－

```
<?xml version = "1.0" encoding = "iso2022-jp"?>
```

－ XML インスタンス－

```
<研究者 登録番号 = "3661">
    <名前>浅川伸一</名前>
    <連絡先>
<郵便番号>167-8585</郵便番号>
<住所>東京都杉並区善福寺 2-6-1 東京女子大学現代教養学部</住所>
        <電話番号>03-5382-6746</電話番号>
         <FAX>03-5382-6709</FAX>
    </連絡先>
    <URL>http://www.cis.twcu.ac.jp/~asakawa/</URL>
</研究者>
```

XML インスタンスとは実際の内容にタグが付けられている部分，つまり XML 文書の本体のことである。この部分は「要素（ element）」と「属性（ attribute ）」からなっている。すなわち

開始タグ　内容　　終了タグ
<郵便番号>167-8585</郵便番号>

となる。このあたりは HTML とよく似ている。

上の例では「連絡先」という要素の中に「郵便番号」「住所」「電話番号」「FAX」という要素が含められている。このとき、「連絡先」を「親要素」、「郵

便番号」「住所」「電話番号」「FAX」を「子要素」と呼ぶ。ちょうど HTML が<body>要素の中に，<p>要素や<ul>要素を子どもの要素として含んでいたことと同様である。XML 文書は（HTML 文書と同じように），親要素から子要素、そのまた子要素、というように要素を階層的に作り上げることで成り立っている。ある要素の開始タグと終了タグの対の中に、子要素として別の要素の開始タグと終了タグを「入れ子」にして書くことによって階層関係を作る。

タグの対応が取れている場合でも、親要素と子要素が入れ子構造を形成していないような次のようなタグ付けは許されないのも HTML と同じである。

要素には、その内容に子要素も文字列ももたない「空要素」と呼ばれる要素がある。その場合には，<要素名 />としてタグの終わりにスラッシュ(/) が必要である。これは HTML の img 要素や br 要素と同じである。

2.4 演習

UNIX には find というコマンドラインユーティリティがある。使い方は，

```
find 検索するパス名 フォーマット
```

となる。例えば，自分のホームディレクトリ直下にある Library というディレクトリから xml という拡張子のついたファイルを探したければ

```
find ~/Library -name '*xml'
```

とする。'*xml' とシングルクォートで囲むのはシェルがこの名前を展開してしまうのを避けるためである。おそらくたくさんのファイルがみつかるであろう。それぞれを順に眺めたければ，less コマンドにパイプでつなげればよい。

```
find ~/Library -name '*xml' | less
```

いくつ xml ファイルがあるかを調べるには wc コマンドをつかう。

```
find ~/Library -name '*xml' | wc -l
```

find コマンドでみつかった .xml ファイルのいくつかを実際に cat, less または lv してみよ。どのような情報が書かれているか。

2.5 演習

自分のホームディレクトリ直下に comp2d というディレクトリを作れ。ワードを起動し “Hello, world” とだけ書かれたファイル Hello.docx を作り，

~/comp2d に保存せよ。

ターミナルエミュレータを起動して comp2d-2009 に移動し，Hello.docx というファイルができていることを確認せよ。

次のコマンドを実行せよ

```
unzip Hello.docx -d Hello
```

このコマンドにより Hello というディレクトリにファイルが展開される。Hello というディレクトリにどんな xml ファイルがあるか確認せよ。

./Hello/word/document.xml という xml ファイルに文書の内容が保存されている。例えば以下のようにである。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
... 中略...
w:rsidRDefault="00943420"><w:r><w:t>Hello,
world.</w:t></w:r></w:p><w:sectPr w:rsidR="009D3640"
... 後略...
```

このファイルの内容のうち，<w:t>Hello, world.</w:t>という部分がタイプした平文であったわけである。例えばこの部分を Hi, all. It is rain, today. のように書き換えて，ターミナルエミュレータを~/comp2d-2009/Hello に移動し，以下のコマンドをタイプすると hello2.doc というファイルができあがる。

```
zip -rp hello2.docx *
```

このファイルをワードで読み込んでみよ。

Emacs でファイルを編集すると，そのファイルのあるディレクトリに，バックアップファイルができてしまう。バックアップファイル名は 編集しているファイル名~である。このファイルを削除してから zip コマンドを実行しないとワードでは読み込んでくれないようである。

3 情報処理技術者試験の過去の問題集より

問 タグを使って文書の論理構造や属性を記述する方法を定めた国際規格であって， 電子的な文書の管理や交換を容易に行うための文書記述言語はどれか。

ア DML

イ HTML

ウ SGML

エ　UML

答　ウ

解説 **ア：**DML（Data Manipulation Language）は、SQL の言語体系の一つで、データ操作言語のことである。

イ：HTML（ HyperText Markup Language ） は、タグ形式のテキストで、HTML で記述したテキストは Web ブラウザを使用して表示できる。

ウ：SGML（Standard Generalized Markup Language ） は、代表的なマークアップ言語の１つで、 文書の中にタグを埋め込んで文書の構造を記述し、図・表などを含む文章の整形言語である。

エ：UML（ Unified Modeling Language ） とは、Java 等のオブジェクト指向の ソフトウェア開発におけるプログラム設計図の統一表記法のことである。 主なモデル図としては、クラス図、ユースケース図、シーケンス図等がある

問 XML の特徴のうち，最も適切なものはどれか。

ア　XML では，HTML に Web ページの表示性能の向上を主な目的とした機能を追加している。

イ　XML では，ネットワークを介した情報システム間のデータ交換を容易にするために，任意のタグを定義することができる。

ウ　XML で用いることができるスタイル言語は，HTML と同じものである。

エ　XML は，SGML を基に開発された HTML とは異なり，独自の仕様として開発された。

答　イ

解説　XML（ eXtensible Markup Language ）は、SGML や HTML と同様にデータの構造や意味をタグを 用いて表現する言語で、独自のタグを定義して使用することができる。

問 XML 文書を構成する最小単位である要素の定義方法に関する記述のうち、適切なものはどれか。

ア　開始タグと終了タグが対になって構成され、どちらのタグも省略できない。

イ　データを開始タグと終了タグで囲んで構成するが、データがないこともある。

ウ　１つの XML 文書には、階層構造を表すために複数のルート要素を定義できる。

エ　要素の種別を表すために注釈情報を付加して、これを要素名として識別する。

答　イ

解説 XML（ eXtensible Markup Language ）は、メタ言語（言語を作る言語ための言語）で 構成されており、 XML により作成された言語を用いて構成された文書やデータを XML 文書と呼ぶ。中には数字の羅列のようなデータ塊のものもある。記述される内容としては、 要素、属性、処理命令、 CDATA セクション、 DTD などがあり、要素の内部には開始タグ、 終了タグ、空要素タグ、文字列などがある。開始タグ、終了タグにより要素を区切り、 要素が空要素のときには空要素タグで示す様にできている。
ア：空要素の定義では、空要素タグを入れることとなっている。
イ：空要素のことを指す。よって、正解。
ウ： XML 文書は木構造となっているので、ルート要素を複数定義することはできない。
エ：注釈情報をタグに付加しても、要素名として識別することはできない。

問 XML に関する記述のうち，適切なものはどれか。

ア　HTML を基にしてその機能を拡張したものである。

イ　XML 文書を入力するためには専用のエディタが必要である。

ウ　文書の論理構造と表示スタイルを統合したものである。

エ　利用者独自のタグを使って文書の属性情報や論理構造を定義することができる。

答　エ

解説 XML（ eXtensible Markup Language ）は、SGML や HTML と同様にデータの構造や意味をタグを 用いて表現する言語で、独自のタグを定義して使用することができる。
ア：誤り。XML は、SGML を基にしてその機能を拡張したものである。
イ：誤り。XML 文書を入力するためには、一般のテキストエディタで十分である。
ウ：誤り。文書の論理構造と表示スタイルを統合したものは、SGML で

ある。
エ：正しい。

問 XML に関する記述として，適切なものはどれか。

ア　C++ を基本としたオブジェクト指向言語

イ　テキスト処理用のインタプリタ言語であり，Web サーバ上で動く CGI（Common Gateway Interface ）プログラムの標準言語

ウ　デスクトップパブリッシングの標準的なページ記述言語

エ　データの構造や意味をタグを用いて表現する言語

答　エ

解説 XML（ eXtensible Markup Language ）は、SGML や HTML と同様にデータの構造や意味を タグを用いて表現する言語である。 XML は、独自のタグを定義して使用することができる。
ア：C++ を基本としたオブジェクト指向言語は、Java である。
イ：テキスト処理用のインタプリタ言語で、Web サーバ上で動く CGI（ Common Gateway Interface ）プログラムの標準言語は、Perl である。
ウ:デスクトップパブリッシングの標準的なページ記述言語は、PostScript である。