

<研究ノート>

XMLに基づく教材配信システム

山島 一浩

A System for the Delivery of Teaching Materials Using XML

YAMASHIMA Kazuhiro

キーワード：XML、インターネット、データベース、学習支援

1. はじめに

近年のインターネットを取り巻く環境は、Webサーバにホームページを登録するサイト構築から、電子商取引（Electronic Commerce）への対応へと急速にシフトしている。

電子商取引において、企業間の取引を「B to B」（Business to Business）、企業対消費者間の取引を「B to C」（Business to Consumer）、消費者間取引を「C to C」（Consumer to Consumer）と分類するものがある。これらの取引で要求される情報構造はそれぞれ違うことが予想される。また、相互のシステム環境が違うこともある。

そこで、インターネット技術を利用し、データベースに蓄積された情報から、それぞれに必要な情報を取り出し、相手のシステムに円滑に情報を配信し、受信側でも効率よく処理することが可能となる仕組みが求められている。

これに伴い、これまで組織内で閉じていた情報システムは、対象となる相手に必要に

じて適切な情報を共有し提供できるシステムへと再構築する必要性が生じている。

その構築の鍵を握る技術として、インターネット上の情報構造を定義するXML（eXtensible Markup Language）と関連技術に注目が集まっている。

現在、筆者は、このXMLを研究対象とし、その事例研究として、教員対学生、教員対教員、そして学生対学生に対して、必要に応じてXML記述に加工した情報モデルを交換するシステムについて検討を行っている。

現段階では、教材提示を中心とするプロトタイプシステムを開発し、これを演習科目の講義で試行しているところである。

本文では、XMLの背景とその特徴を整理し、それに従い構築した教材配信システムについて述べる。

2. XMLについて

XMLは、マークアップ言語である。マークアップ言語は、表示する内容を<タグ名>内

容</タグ名>の形式で記述する。タグには処理を識別するために特徴を示す情報を付ける。そのタグにより挟まれた内容がどのような情報であるのかを特定する言語である。

ここでは、まずマークアップ言語について、SGMLからHTML、そしてXMLへの展開を概観し、本研究のプロトタイプシステムの構築に用いたXMLの関連仕様について述べる。

2 - 1 . SGML からの派生

現在、ホームページは、HTMLで記述されている。

元来、HTMLは、デジタル化された文書の構造を定義するSGML (Standard Generalized Markup Language) から派生したマークアップ言語である。

SGMLは、文書構造を記述するマークアップ言語のISOの規約 (ISO8879) である (1986年)、日本では、1992年にJIS規格 (JISX 4151) になっている。

SGMLの記述の特徴は、DTD (Document Type Definition) を用いて文書の構造定義を行う点である。従ってHTMLもDTDで定義されている。これにより文書構造を決定した上で、その規則に従い、内容を記述していく。

文書構造の基本は、章・節・段落といった階層構造である。SGMLでは、この他の注・参考文献・図表などの非階層の構造も表現できるようにになっている。

デジタル文書を記述する国際標準とされながらSGMLは、今日のHTMLのような一般的な普及には至っていない。その一因として、厳密な構造定義が記述の上で煩わしさとなっている点が指摘されている。

2 - 2 . HTML の発展と課題

一方HTMLは、Webブラウザの表示を決定するマークアップ言語として採用され、インターネットの普及によりWWW上で独自の進化を遂げていく。特にWebブラウザを供給し

た企業により、Webブラウザでの見せ方を重視したタグの拡張がなされていった。

ところが、急速に普及したWWWにおいて、HTMLで記述した情報資源が拡大するに伴い、HTMLに対して、次第に次のような課題が指摘されるようになった。

- 1 . HTMLは、見せ方のスタイルを示す固定化されたタグを用いるので、利用者が必要に応じてタグを拡張できない。
- 2 . 記述した内容の特徴を示す適切なタグ情報が無いために、情報検索での検索の対象の適切な絞込みや、情報共有面で送受信する側のアプリケーションの操作対象として特定しにくい。

このようなHTMLの限界を補完するために、新たなマークアップ言語仕様として、XMLが登場する。

2 - 3 . XML の誕生

XML誕生までの経緯は、まず1996年のSGML/96において、Tim Brayが、XMLについての構想を発表する。Tim Brayは、Jon Bosakと共にその後のXML仕様策定に大きく貢献した人物である。この年の11月、W3CにおいてXML規格 (第1版) のドラフトが出される。特にXMLの関心を加速させたのは、1997年、当時すでにパーソナルコンピュータのOS (Operating System) で大きなシェアを占めていたMicrosoftのBill Gatesが行った、XMLの可能性を示す講演であるとされる。

その翌年の1998年に、W3C (World Wide Web Consortium) においてXMLの1.0版が勧告される。その後、XMLの拡張が検討され、2000年10月にXML1.0 Second Editionが勧告されている。

2 - 4 . W3C について

ここで、1994年に発足したW3Cについて触れておく。W3Cは、WWWで利用される技術の標準化をすすめる団体であり、ここで勧告

を受けた仕様は、WWW上の標準とされる。企業には、ここで勧告されたものをWebアプリケーションに反映させることを課している。当然ながら先に述べたHTMLも、W3Cにおいて標準化が行われている仕様である。

従ってXMLがW3Cで勧告されたことは、WWW上での標準を示すという意義を持つのである。

2 - 5 . XML の進展

XMLの目標は、先のHTMLの課題を解決するためにSGMLの特徴を生かし、HTML以上に、文書構造を正確に伝達する。その一方で、SGMLの機能を大幅に切り捨て、インターネットに対応できるよう、ハイパーリンク等の機能を拡張し追加する。その拡張機能がXlink、Xpath、Xpointerである。

またXMLの登場を受けて、それを採用した関連規格も出ている。例えば、W3Cで勧告されたマルチメディアプレゼンテーションを記述するSMIL (Synchronized Multimedia Integration Language)、プッシュ技術を利用したMicrosoftのCDF (Channel Definition Format)、Web上でメタデータを記述するRDF (Resource Description Framework) などである。

さらに、XMLのプロセッサとして、W3Cで策定したDOM (Document Object Model) とDavid Megginsらが策定しているSAX (Simple API for XML) がある。これらを利用することで、XML文書は処理される。

2 - 6 . XML の特徴

次にXMLの特徴について、HTMLで述べた課題に対する解決策として順に述べる。

まず、XMLは、タグを自分で自由に定義することができる。これにより、<テーマ>太平記について</テーマ>などのように定義した場合、「太平記について」がテーマであることがタグにより特定される。

次に、そのタグは、インターネット上で情報交換する際に相互で定義したタグ情報をもとに内容を遣り取りし、相互のシステムで処理する際に、そのタグ名を処理対象の識別として利用できる。先のタグ文を使うと、複数のホームページからテーマを一覧にする場合には、テーマタグがその処理対象であると指定することで、処理が行えるのである。

当然のことながら、それらを相互に運用する場合には、事前にその定義内容がお互いに理解できるよう、そのXMLタグに関する規約が必要となる。

2 - 7 . XSLT による XML の構造変換

同じ構造を持つXML文書同士を処理することは容易であるが、構造が違う場合には、対応づけして、相互のシステムに変換規則を適用する必要がある。そこで、XSLT (eXtensible Stylesheet Language Transformations) を用いる。

XSLTは、1999年にW3Cで勧告されたXMLの変換をおこなうための仕様である。

XSLTを利用し、XML情報モデルを別構造のXML情報モデルへと変換したい場合、その規則が明示することができれば、XSLTの記述に従って変換規則を書き、XSLTの処理系を用いることで目的の変換を行う。MicrosoftのInternet Explorer5.0には、XSLTの機能の一部が実装されている。

本システムの特徴は、このXMLでの教材提示の構造定義とXSLTを用いてHTMLへ変換を行う点である。

3 . プロトタイプシステムの概要

3 - 1 . 支援目標について

冒頭で述べた通り、本システムでは、学習時に発生するコミュニケーション情報について、教員対学生、教員対教員、そして学生対学生という3つの観点から情報モデルを分類

し、データベースに登録した情報から、それぞれに必要なXMLの情報モデルを作り出し提供することを目標にしている。

教員对学生間では、教員側で作成した教材を画面上に提示し、教員側から学生への説明に用いる。学生側がこれを受けて、生じた質問をホームページから投稿できる等のコミュニケーション支援である。この場合、大教室で教員が多数の学生を相手に講義を行う場合や、遠隔地で教材を閲覧するような環境を想定している。

2番目は、課題提出や出席状況などを元に、学生の指導を行う場合、教員間の情報を共有し連携を支援するのが目的である。学生が登録した出欠情報や、提出物の一覧形式で確認でき、そこから分析が行えるような情報を提供する。

最後に、学生对学生であるが、これは、教えあい励ましあうというような学生間の情報交換により期待される学習効果を狙った支援である。学生の質問から同じものを抽出し、それを掲示板のタイトルにし、そこに書き込みができるようにして、学生間で解決策を議論してもらおうとするものである。

これらの支援目標に対して、本システムでは、先の2点について、教材提示と課題や出欠などの閲覧が行えるような機能を用意した。最後の支援機能については、表題作成の自動化における抽出方法について課題が残っている。そこで、先の機能内で学生が行う質問形式について分析を行い、それにより具体的な支援機能を付け加えることを検討している。

3 - 2 . プロトタイプシステムの概要

ここでは、実装した機能について述べる。本システムは、Webブラウザ上で全ての操作を行う。

サーバOSは、WindowsNT4.0を使用し、Webサーバには、IISを使用している。データ

を蓄積するデータベースは、MS Access2000で、ODBCを経由して、ASP (Active Server Page) のスクリプトファイルに記述したSQL操作命令でデータを操作する。

Fig. 1 は、このシステムの初期画面である。左隅に選択メニューを配し、右側に教材内容を提示する。選択メニューをクリックすると、該当するページに移れるようになっている。



Fig. 1 初期画面

3 - 2 - 1 . 教材管理

教材管理では、XMLで記述した教材を配信する。まず設計した教材提示のDTDをList 1に示す。

```
<!ELEMENT COBOL (T,T2,PG,EXP,SUB,SPG,PX)>
<!ELEMENT T (#PCDATA)>
<!ELEMENT T2 (#PCDATA)>
<!ELEMENT PG (#PCDATA)>
<!ELEMENT EXP (#PCDATA)>
<!ELEMENT SUB (#PCDATA)>
<!ELEMENT SUT (#PCDATA)>
<!ELEMENT SUBM (#PCDATA)>
<!ELEMENT SPG (#PCDATA)>
<!ELEMENT PX (PM,PIMG)>
<!ELEMENT PM (#PCDATA)>
<!ELEMENT PIMG (#PCDATA)>
```

List 1. 教材提示のためのDTD

このDTDでは、教材提示を構成する要素に、例題、解説、補足説明、入出力イメージの画像名とそのタイトルを一つの情報モデルとして定義している。

そこで教材提示を行うには、List1で定義したDTDに従い、XML文書を記述する。List2は、その一例である。XML文書をFig.3のようにブラウザで表示さ

せるには、HTML形式に変換する必要がある。

XMLのタグをどのHTMLタグと対応させるのかを List3のようなXSLTで記述する。定型なHTMLページを作成するのであれば、同じ構造を持つXML文書に対して、同じXSLTファイルを使用することでHTMLタグに変換できる。

```
<?xml version="1.0" encoding="shift_JIS"?>
<!DOCTYPE COBOL SYSTEM "mytext-01.dtd">
  <?xml:stylesheet type="text/xsl" href="cb101.xsl"?>
<COBOL>
<T>並べ替える (ソート) </T>
<T2>SORT 命令 (入力ファイルを加工してから、ソートする) </T2>
<PG>
IDENTIFICATION DIVISION.
PROGRAM-ID, ST-DATA2.
  - 省略 -
CLOSE MEIBO-F
</PG>
<EXP>
見出し部
プログラム I D
  - 省略 -
  入力ファイルを閉じる
</EXP>
<SUB><SUT>補足説明</SUT>
<SUBM>
入力ファイルを加工してからソートするには、INPUT PROCEDURE を使用します。
</SUBM></SUB>
<PX><PM>データ</PM><IMG>402.gif</IMG></PX>
</COBOL>
```

List. 2 DTD に従って記述したXML文書

```
<?xml version="1.0" encoding="Shift_JIS"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"
  xmlns:html="www.w3.org/TR/REC-html40" result-ns="html">
<xsl:template match="/">
<html>
<body bgcolor="#ffffff">
<div id="Layer7" style="position:absolute; left:9px; top:14px; width:542px; height:94px; z-index:13">
<font size="3" face="MS ゴシック, Osaka-等幅">【COBOL教材】 テーマ: <B>
<xsl:value-of select="表題/題"/></B></font>
<table width="100%" border="1" height="60">
  <tr> <td width="41%" bgcolor="#CCCCFF">
    <div align="center"><font color="#000000" face="MS Pゴシック, Osaka" size="3">メニュー</font></div>
  </td></tr>
  <xsl:for-each select="表題/課題">
<tr><td width="41%" bgcolor="#cccccc"> <font color="#FFFFFF" size="3" face="MS ゴシック, Osaka-等幅">
<A><xsl:attribute name="HREF"><xsl:value-of select="問題"/></xsl:attribute>
<xsl:value-of select="題"/></A></font>
  </td></tr>
</xsl:for-each>
</table>
</div>
</body>
</html>
```

List. 3 XSLT文書

具体的には、指定したXML文書とXSLTをサーバ側で処理するASP（これはVBScriptで記述する）に指定することにより、HTML文書が配信される仕組みである。

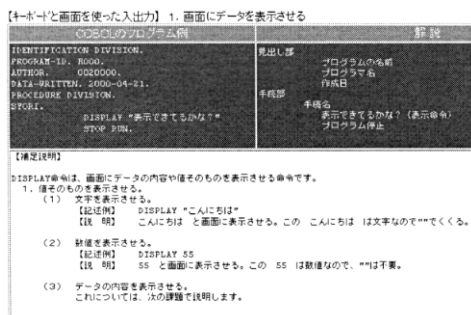


Fig. 2 表示された教材画面

3 - 2 - 2 . 課題管理

課題提出は、Fig.3のレポート送信ウィンドウの課題記入欄に入力し、学籍番号と問題番号を記入した上で送信ボタンを押すとデータベースに登録される。

```
<?xml version="1.0" encoding="Shift_JIS"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"
  xmlns:html="www.w3.org/TR/REC-html40" result-ns="html">
<xsl:template match="/">
<html>
<body bgcolor="#ffffff">
<div id="Layer7" style="position:absolute; left:9px; top:14px;
<font size="3" face="MS ゴシック, Osaka-等幅">【COBOL教材】
<xsl:value-of select="表題/題"/></B></font>
<table width="100%" border="1" height="60">
<tr> <td width="41%" bgcolor="#CCCCFF">
<div align="center"><font color="#000000" face="MS
</td></tr>
<xsl:for-each select="表題/課題">
<tr><td width="41%" bgcolor="#cccccc"> <font color="#FFFFFF"
<A><xsl:attribute name="HREF"><xsl:value-of select="問題"/><
<xsl:value-of select="題"/></A></font>
</td></tr>
</xsl:for-each>
</table>
</div>
</body>
```

Fig. 3 レポート送信ウィンドウ

ここで、模範解答がある場合には、送信後に模範解答が表示され、学生が自己採点を行

うようになっている。合っていたかどうかをYESかNOで選択させ、質問や感想を記入し再送信させている。自動的に判定させず、自己採点させるようにしたのは、例えばプログラミングの課題で、アルゴリズムの間違いであるにも関わらず、正答内容と区別できずに正解とした場合が問題であるとし、その傾向を分析するための情報を収集することを意図した。

提出したレポートは、IDとパスワードを入力することにより、閲覧できる。

教員側からは、課題別にレポートの内容を確認できるように、課題番号別の提出物一覧として閲覧できる。学生側からは、どの課題が未提出なのか、提出した内容は正しい答えになっているのかが確認できるよう、提出した課題番号別の一覧で見ることができる。

Fig.4は、学生の提出物一覧の例である。提出した課題レポートと、模範解答の一覧を見ることができる。更に提出の際に質問事項があれば、それを送信できるので、送信した場合には、そのコメントが一覧についてくる。学生は何度でも解答できるが、見かけ上一覧出力には、最後に登録したものが見える。一方、教員側では、学生が提出した内容全てが見えるので、誤答から正答を導き出すに至る傾向を把握することも可能となろう。

3 - 2 - 3 . 出欠管理

講義の出欠を省力化して把握するために、出欠管理機能を用意した。授業開始時に、各自が学籍番号とパスワードを入力し送信することで仮登録が完了する。授業完了後に教員側が登録時刻を確認し、認証を与えることで出席が確定する。未登録の場合には、欠席となる。

() さんのレポートは以下の通りです。

課題番号	課題解答	提出物	提出時間	本人のコメント	アドバイザー等
2	IDENTIFICATION DIVISION. ENVIRONMENT DIVISION. DATA DIVISION. PROCEDURE DIVISION.	IDENTIFICATION DIVISION. ENVIRONMENT DIVISION. DATA DIVISION. PROCEDURE DIVISION.	00/04/21 10:22:30		
F0401	WORKING-STORAGE SECTION. 01 A PIC 9(5). 01 B PIC 9(5). 01 C PIC 9(5).	WORKING-STORAGE SECTION. 01 A PIC 9(5). 01 B PIC 9(5). 01 C PIC 9(5).	00/04/28 14:32:52		
F0402	WORKING-STORAGE SECTION. 01 F. 02 G PIC X(10). 02 H PIC X(5). 02 I PIC X(4). 01 J PIC X(3). 01 K PIC 9(6). 01 L. 02 M PIC X(7). 02 N PIC 9(6).	WORKING-STORAGE SECTION. 01 A. 02 B PIC X(10). 02 C PIC X(5). 02 D PIC X(4). 01 E PIC X(3). 01 F PIC 9(6). 01 G. 02 H PIC X(7). 02 I PIC 9(6).	00/04/28 14:44:58		

Fig. 4 学生が確認する個人レポート一覧

また、全員に毎回声をかけることが不可能な状況で、コミュニケーションに要する時間を拡大する手段として、コメントを書き込めるようにした。試行している講義で、現在までに11件の書き込みがあった。その内容は、近況やその日の気分、操作環境についてであり、いずれも意図した内容となっており、同様の書き込み内容が、レポート提出内容に混入されることなく区分できている。

クラス出席簿		00/04/14	00/04/21	00/04/28	00/05/12	00/05/19
学籍番号	氏名					
	■■■■	○	○	○	○	○
	■■■■	○	○	○	○	○
	■■■■	○	○	○	○	○
	■■■■	○	○	○	●	●
	■■■■	○	○	○	○	○
	■■■■	○	○	○	○	○
	■■■■	○	○	○	○	○
	■■■■	○	○	○	○	○
	■■■■	○	○	○	○	○
	■■■■	○	○	○	○	○
	■■■■	○	○	○	○	○
	■■■■	○	○	○	○	○
	■■■■	○	○	○	○	○
	■■■■	○	○	○	○	○

Fig. 5 出欠一覧

さんの出欠状況は、以下の通りです。
当日の授業分については、授業終了後30分ほどして更新されます。
その間も、遅刻状態にありますが問題はありません。

日付	テーマ	認証
00/04/14	ガイダンス	出席
00/04/21	簡単な入出力	出席
00/04/28	データ部(作業節)	出席
00/05/12	データ部(ファイル節)	出席
00/05/19	基本的なファイル入出力・環境部とデータ部の記述について・テキストエディタによるファイルの作成とプログラムでの複製	欠席
00/05/26	ファイル入出力 一覧表の作成	出席
00/06/02	一覧表の出力(ファイルと印刷設計)	出席
00/06/09	一覧表出力(編入処理)	出席

Fig. 6 授業履歴一覧

2000年度 COBOL演習 授業履歴一覧

Aクラス		
日付	時間	課題
00/04/14	3	ガイダンス
00/04/21	3	簡単な入出力
00/04/28	3	データ部(作業節)
00/05/12	3	データ部(ファイル節)

日付	時間	課題
00/04/14	3	指定された3つのプログラムを完成させてください
00/04/21	3	課題番号: 2 4つのDIVISIONを入力してください。
00/04/28	3	課題番号:F0401 F0402(P105) を提出しなさい。
00/05/12	3	課題番号:6 を速達で提出しなさい。 また、教科書F0305~F0315 (P81-99) までの問題について

Fig. 7 授業履歴一覧

Fig.5は、教員側で閲覧する出席状況一覧である。学生側には、Fig.6のように、何日の授業の何のテーマの際に欠席したかが分かるようになっている。

3 - 2 - 4 . 授業履歴揭示板

授業履歴を日付、テーマそしてその日の課題について一覧で出力する。ブラウザで誰でもその日の授業のテーマを知ることができ、さらにそこで出された課題も確認できるようになっている。Fig.7は、その一覧の表示画面である。

4 . 講義での試行

このプロトタイプシステムを実際の講義で試行している。試行した科目は、「プログラミングⅠ演習」である。

講義では、平木茂子他著の「ファイル処理入門」 - COBOLの文法 - をテキストに使用している。このテキストは、読者が自習で学べるように考慮されている。

このテキストを用いた学習の展開は、次のようなものである。読者はまず焦点が絞られた解説を読む。その後、豊富な課題が用意されているので、これをこなす。いわゆる「覚えるより慣れる」により、学習の目標を達成する。

そこで講義では、この本の補足説明用に提示システムを利用し、課題提出機能では、テキスト内の課題を送信する他、応用課題もその送信対象として利用した。

現在のまでの5ヶ月間に、2265件のレポート提出が行われ、237の質問が送信されている。

学生からの発問をその場で収集することにより、教員側の講義内での説明とそれに対し誤った解釈を行った学生について、その関連性を分析するためのサンプルが、このシステムを利用することで収集できつつある。

今後は、これらを元に、誤った答えを提出した学生に対して、補助教材へと自動的に導く自動返信機能を拡張する予定である。

6 . まとめ

XMLについて概観し、その仕様を活用して構築した教材配信システムについて述べた。プロトタイプシステムの特徴は、一つの情報モデルを必要に応じてXML形式の情報モデルへと変換し配信する点である。

XML勧告以降、様々な関連仕様が登場している。それらをブラウザやデータベース等の関連するソフトウェアへ対応させていくスピードも早い。例えば、ベンダーやフリーソフトで提供されているデータベースにXMLを処理する機能が標準で装備されつつある。また、XML文書をそのままの構造でデータベースに登録し、操作するシステムも登場している。

W3Cにおいては、核となるXML標準化活動が一段落しつつあり、今後は一層実用化と普及へのアプローチが強化されるものと思われる。

今後は、これらの動向を含め、得られた知見を用いて、インターネットを通じた遠隔教育への対応も検討していきたい。

【参考文献】

- [1] SGMLの活用 根岸正光 石塚英弘 オーム社1994
- [2] ネットワーク社会におけるXMLの役割とその展望 大野邦夫 情報知識学会誌 Vol.10.No.3 2000
- [3] www.w3.org/TR/WD-xml961114.html
- [4] www.w3.org/TR/1998/REC-xml-19980210
- [5] www.w3.org/TR/2000/REC-xml-20001006
- [6] 標準XML完全解説 XML/SGMLサロン 技術評論社 1998
- [7] ファイル処理入門 - COBOLの文法 - 平木茂子・荒木雄豪・今井恒雄 恒星社厚生閣 1990