

XML を基にしたシラバス管理システムの実現

江本 守 荒木 雄太 大河内 久貴 大淵 滋樹
那須 正裕 松原 幸平 横田 一正 国島 丈生

岡山県立大学 情報工学部 情報通信工学科

{emoto, araki, ohkochi, ohbuchi, nasu, matubara, yokota, kunishi}@c.oka-pu.ac.jp

〒719-1197 岡山県総社市窪木 111

あらまし 最近の情報化の急速な変化は、大学の種々のシステムにも大きな影響を与えている。大学システムは大別すると、教育システム、研究システム、図書システム、事務システムに分けられ、それらが連動することが望まれている。すでに多くの大学で先進的なシステムが数多く開発されている。そこで、岡山県立大学でもシステムを統合するため、その第一歩として、Web を利用したシラバス管理システムを XML 文書管理の視点から構築し、データ記述/交換に XML を用いることの有用性を確認した。

キーワード: XML, シラバス管理システム, 教育システム

Implementation of an XML-Based Syllabus Management System

Mamoru EMOTO Yuta ARAKI Hisataka OHKOCHI
Shigeki OHBUCHI Masayuki NASU Kohei MATUBARA
Kazumasa YOKOTA Takeo KUNISHIMA

Faculty of Computer Science and System Engineering, Okayama Prefectural University

{emoto, araki, ohkochi, ohbuchi, nasu, matubara, yokota, kunishi}@c.oka-pu.ac.jp

111, Kuboki, Soja, Okayama, 719-1197 Japan

Abstract The rapid change of the recent development of information technologies gives various systems of universities a great influence. Systems to the public have been developed based on Web, while an internal systems, too, assume that all people can access easily various resources in universities. University systems are classified mainly into education systems, research systems, library systems, and administrative systems. And it is expected that they work together. An advanced system has already been developed abundantly in many universities. Though various systems have already been built in Okayama Prefectural University, each system is being developed independently. As the first step for integrating those system, we developed a syllabus mangement system, from a viewpoint of XML document management. In this paper, we report an overview of our system and discuss the effectiveness of XML.

Key words: XML, syllabus management system, education system

1 はじめに

最近の情報化の急速な変化は、大学の種々のシステムにも大きな影響を与えている。対外的な情報発信や情報公開は Web を中心に進展しており、内部の情報システムもすべての人が簡単にさまざまな情報にアクセスできることを前提に見直しが進みつつある。大学システムは大別すると、教育システム、研究システム、図書システム、事務システムなどに分けられ、それらが連動することが望まれている。すでに会津大学を始めとして多くの大学で先進的なシステムが数多く開発されている。

岡山県立大学でもさまざまなシステムがすでに構築されているが、大学システムとして考えれば、個々のシステムが独立に開発されていて、それらが連動しているとはいえない状況である。これらを現在見直しつつあり、その一環として教育システムのひとつであるシラバス管理システムを開発した。このシステムは、Web 環境で、XML とデータベースを使って実現したもので、XML の有用性を確認したので、本稿ではその概要を報告する。

2 節では、今回開発したシステムの概要と実現方式について説明し、3 節で今後の課題と拡張を述べる。

2 シラバス管理システム

本節では、シラバス管理システムについて、要件、実現するためのアプローチ、システムの概要とその評価について述べる。

2.1 教育システムとシラバス管理システム

まず教育システムの中でのシラバス管理システムを考えたい。教育システムは、まず

- シラバス
- 履修
- 成績

の管理が必要となる。シラバスには、カリキュラムが変更になったときの新旧の管理、JABEE モデルの提供が必要である。履修は、学生情報とシラバス情報が関連したもので、講義スケジュールのコンフリクト、最大履修条件の反映、先行科目の確認などが必要となる。成績は、評価式の公開、GPA の計算が必要である。

この他に、電子的な形態を考えれば

- 学生のレポート
- 講義内容と質疑

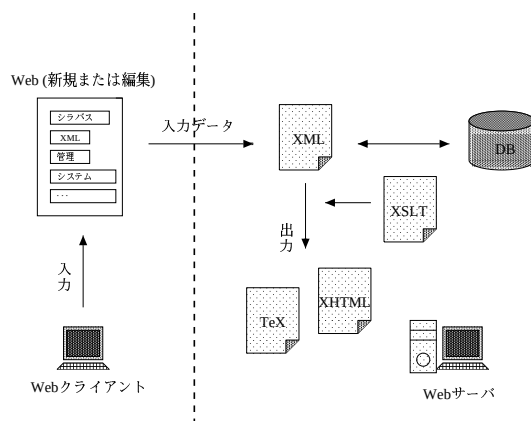


図 1: XML を基にしたシラバス管理システム

のアーカイブが必要となる。学生から電子的に提出されるレポートには、コピーをチェックするためにレポート間の類似性のデータマイニングと、評価項目の自動チェックが望まれる。講義内容は、講義の進行にともなってその内容と質疑を変更させたい。JABEE への対応を考えれば、講義内容だけでなく、種々の成績も、物理的な保管と電子的なアーカイブを対応づけたい。

また講義スケジュールの変更などの連絡を行なう掲示板に機能も持たせたい。

このような観点からシラバス管理システムを考えると、個々のシラバスの管理の他に、履修モデルへの関連づけや、他とのデータのやりとりを十分に考慮する必要がある。成績システム等の他のシステムは既存のシステムがあるため、共通モデルとして XML を基にしたシステムとした。

2.2 要件

シラバス管理システムに求められる要件は、

- 既存の他のシステムとの連携
- Web 上でシラバス登録・修正作業が行なえる
- シラバスをデータベースで管理
- シラバスの提示

である。シラバス及びその管理に必要なデータを収容するデータベースを中心とし、Web と連携することによってその管理と運用を行なう。

2.3 アプローチ

先にも述べたように、成績システム等の他のシステムとの連携を考え、XML を基にしてシラバス管理システムを実現する (図 1)。

		科目コード		21C126	
授業科目名 (英大英記)		データ工学 Data Engineering		担当教員 横田 一正 岡島 文生	
対象学生	3 年次生	授業の形態	講義	週回・必修の科目	選択
単位数	2				
履 修 条 件	3 年次生がデータベースの知識を前提として履修すること。その前提、基礎、初級を工学的に考えることを目指したデータベースについて、基礎から応用までを学ぶ。その中でよく理解した要素を基に関係データベースシステムを中心に、データベースの基本的概念と応用を理解する。さらに、複雑なデータベースや知識を扱うために、関係データベースの拡張などについても学ぶ。				
授業科目の 目 的	1. データベースの基本概念と機能 2. 関係データベースの概念、構造、設計 3. 関係データベースの処理の基礎 4. オブジェクト指向と分散データベース 5. 分散データベース、マルチデータベース、メディアエータ				
履修上の注意	履修の条件：人工知能を履修していることが望ましい。 その他：授業計画に示した項目に従って履修を行う予定であるが、番号は必ずしも一部分の講義に該当していない。				
授業 目 的	1. データ工学の概要				
	2. データベースの基礎概念：データベース、三層スキーマ				
	3. 関係モデルの基礎				
	4. 関係の操作：関係代数、SQL				
	5. 関係の設計：関係の正規化				
	6. トランザクションと障害回復				
	7. 索引構造と索引方式：B+ 木、ハッシュ関数、線形方式				
	8. 関係データベースの管理：管理による関係データベースの内部構造				
	9. 分散データベース：Distributed				
	10. 分散データベースの最適化：上層層面と下層層面				
要 点	11. オブジェクト指向データベース				
	12. 項と述語の拡張：表現のための代数的基礎				
	13. 分散オブジェクト指向：4-部からオブジェクト指向へ				
	14. 分散システムと情報科学：分散データベース、マルチデータベース、メディアエータ				
	15. 総括および演習				
成績 評 定	試験成績表およびレポートにより総合的に評価する。				
関連授業科目	計算機言語 II、情報ネットワーク				
教 材	教科書：関係データベース Web で公開、関連資料は適宜配布。 参考書：横田一正、岡島文生「データベース設計」、共立出版、北川博之「データベースシステム4」、組立書。				
担当教員への 連絡先	自宅電話番号：2699 電子メール：yokota@cs.cmu.ac.jp その他：				
備 考	特になし				

図 2: シラバスの印刷例

システム内部のデータ交換を XML で行なうのが特徴である。また、シラバスのデータを XML で表現するので、XSLT などを用いて様々な形式に変換できる。具体的に、シラバスを

- 印刷物にするため、TeX ファイルに変換
- Web で公開するため、XHTML ファイルに変換を行う。

2.4 システムの概要

シラバス管理システムはシラバスの登録・管理・提供を行なう。シラバスをデータベースとして管理し、登録・修正作業は Web 上で行なう。

2.4.1 XML のスキーマ

システムの開発において、はじめに行なわなければならないのは XML のスキーマの設計である。我々は、次の手順で DTD の作成を行なった。

1. XML を処理した結果としての文書（印刷物）を分析し、タグとして表現する要素を洗い出す
2. 要素の階層関係、要素の出現順序を決める
3. 要素に付ける情報を属性として決める

まず、シラバスとはどのようなものかを調べると、シラバスとは、大学などの教育機関における授業の目標と内容、教科書・参考書、成績評価の方法など履修にあたっての準備に必要な情報を提供するものである。シラバスの印刷例は図 2 となる。こ

```
<ELEMENT syllabus
  (subject_code,
   subject_name,
   teachers,
   student,
   subject_form,
   choice,
   credit,
   outline,
   object,
   attention,
   subect_plan,
   granding,
   relation,
   text,
   contact,
   remark
)>
```

```
<ELEMENT subject_name (name+)>
<!ATTLIST name xml:lang (ja|en) #REQUIRED>
<ELEMENT teachers      (teacher+)>
<ELEMENT outline       (p+)>
<ELEMENT object        (p+)>
<ELEMENT attention      (necessary,
                           others)>
<ELEMENT necessary      (p+)>
<ELEMENT others         (p+)>
<ELEMENT subjecet_plan (li+|p+)>
<!ATTLIST subject_plan form
  (list|free) "list">
<ELEMENT li             (p+)>
<ELEMENT granding       (p+)>
<ELEMENT relation       (p+)>
<ELEMENT text           (text_books,
                           ref_books)>
<ELEMENT text_books     (book*)>
<ELEMENT ref_books      (book*)>
<ELEMENT contact        (room_num,
                           e-mail,
                           others)>
<ELEMENT others         (p+)>
```

XML のスキーマ

れを分析し、タグとして表現する要素を洗い出し、設計したスキーマは上の DTD である。ただし、この DTD は完全ではない。

このスキーマはそれほど深く掘り下げずに設計を行なった。例えば、教科書、参考書において、それぞれ 1 冊を<book>で囲むだけである。著者、作品名、出版社など細かく構造化できるが、教員が簡単に登録・編集（レイアウト）を行なえることを考慮に入れ、設計した。

また、「授業計画」においては教員からの要望で、「リスト形式」と「自由形式」の書式を用意した。「リスト形式」とは、授業のテーマを毎回書く基本的な形式で、教員は箇条書に書けば、順に番号が割

り当てられる。「自由形式」とは、研究のようにテーマが決められない、または大まかにテーマを決めたい方の記述形式で、番号は割り当てられない。「リスト形式」と「自由形式」を属性を用いて次のように表現した。

```

----- 「リスト形式」 -----
<subject_plan form="list">
  <li><p>…</p><p>…</p></li>
  <li><p>…</p></li>
  :
</subject_plan>

----- 「自由形式」 -----
<subject_plan form="free">
  <p>…</p>
  <p>…</p>
  :
</subject_plan>

```

このようにスキーマが決まれば、Web 入力部はこのスキーマに従って XML 文書を入力すればよい。そして、提示部ではこの XML 文書を入力とし目的に合った XSLT スタイルシートを作成すればよい。また、データベースでは XML 文書の格納と DB から XML 文書を生成できればよい。次に、これら 3 つのパートについて説明していく。

2.4.2 Web 入力（登録・修正）

シラバスの入力・編集は授業の各担当教員が Web 上で行なう。登録画面は図 3 である。

入力が完了しデータを送信すると、サーバ側では XML 文書として出力される。これは 1 授業 1 ファイルの形である。この XML 文書を基に、TeX ファイルが作成され、DVI を経て、図 2 のような PostScript（以下、PS）ファイルが作成される。そして、この PS ファイルにリンクが張られる。この変換と同時に XML 文書をデータベースに保存する。図 4 に登録における一連の処理を示した。

編集の場合は、前の入力データをデータベースから取り出し、あらかじめ入力フォーム内に埋める。これにより、教員の入力の手間を低減できる。

また、この PS ファイルは最終的に学生や大学関係者に配布する出版物になる。出版用であるため、物理的なレイアウトは定められている。各担当教員はこの PS ファイルを見ながら、見栄えの良さを考慮に入れ、また誤字・脱字など修正・編集を行なう。よって、シラバスの内容や見栄えは各担当教員に責任が委ねられる。例えば、図 3 のような入力をする

図 3: シラバスの登録画面

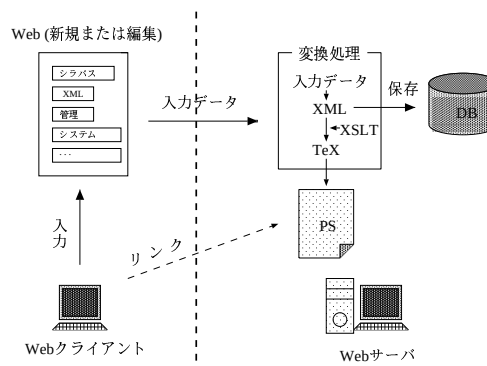


図 4: シラバスの登録

この入力システムは Perl を用いて CGI プログラムで実現した。また、この一連の変換作業（XML → PS）はシェルスクリプトで記述し、CGI プログラムからそのシェルスクリプトを呼び出している。

ここで、入力データから XML 文書を生成するときに注意しなければならないのは、入力データ中にタグ付けに使う 5 つの文字

<, >, &, ' (アポストロフィ), " (二重引用符)

が含まれていれば、エンティティに置換しなければならない。置換しなければ、XML 文書は well-formed でないので、XML プロセッサはエラーを吐き、TeX ファイルが生成されない。その結果、PS

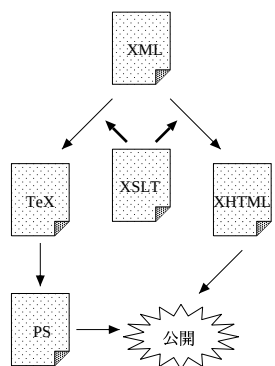


図 5: XSLT による変換

ファイルも生成されないことになる。この置換作業は、Perl の強力な正規表現によるパターンマッチと置換を用いて行なった。

2.4.3 提示に関する変換

シラバスのデータは XML で表現されているので、XSLT を用いて、入力データの XML 文書を次の 2 つに変換する (図 5)。

1. シラバスの一覧を配布するため、高品位文書作成に適した $\text{T}_{\text{E}}\text{X}$ ファイルに変換
2. シラバスを Web 上で公開するため、XHTML に変換

1 つ目は、出版用である。また教員が編集を行なうにあたって $\text{T}_{\text{E}}\text{X}$ ファイルから生成される PS ファイルが必要となる。つまり、この PS ファイルは教員が登録の送信ボタンを押す毎に自動的に生成される (XML $\rightarrow$ $\text{T}_{\text{E}}\text{X}$ $\rightarrow$ PS)。

2 つ目は、シラバスの登録がすべて完了した後、シラバスを Web 上で公開するため行なう変換処理である (XML $\rightarrow$ XHTML)。

この XSLT の実装方法は、まず基となるフォーマットを決める。例えば、 $\text{T}_{\text{E}}\text{X}$ ファイルに変換する XSLT であれば、最終的に図 2 になるような $\text{T}_{\text{E}}\text{X}$ ファイルを作成する。そして、XSLT はこの $\text{T}_{\text{E}}\text{X}$ ファイルを基に、各項目に XSLT の処理命令のひとつ `xsl:value-of` を記述することで、各授業のデータを挿入することができる。右に $\text{T}_{\text{E}}\text{X}$ ファイルに変換する XSLT の一部を示す。このように記述すれば、各授業のデータを挿入することができる。

しかし、実装は容易なものではなかった。例えば、`xsl:template` を指定していない要素の内容が出力されることがよくあった。これは、XSLT にはマッチするテンプレートがなくても処理が続行できるようにするための「ビルドイン・テンプレート規則」

```

<?xml version="1.0" encoding="euc-jp"?>
<xsl:stylesheet
  xmlns:xsl
    ="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:output method="text"
    encoding="euc-jp"/>

  <!-- シラバス XML の文書ノードに
        対して TeX の枠組みを出力する -->
  <xsl:template match="syllabus">
    \documentclass[a4j,10pt]{jarticle}
    :
    % エラーの無視
    \batchmode

    \begin{document}

    \begin{tabular}{|c|c|c|c|c|c|c|c|}

    % 科目コード
    \multicolumn{5}{c|}{ }
    <xsl:text disable-output-escaping="yes">
      &
    </xsl:text>
    <xsl:text> </xsl:text>
    \parbox[c][6mm]{20mm}
    {\centering 科目コード}
    <xsl:text disable-output-escaping="yes">
      &
    </xsl:text>
    <xsl:text> </xsl:text>
    \multicolumn{3}{c|}{
    <xsl:value-of select="subject_code"/>
    }
    \\ \hline
    :
    \end{tabular}
    \end{center}
    \end{document}
  </xsl:template>
  :
  <xsl:template match="p">
    <xsl:value-of select="."/>
  </xsl:template>

</xsl:stylesheet>

```

$\text{T}_{\text{E}}\text{X}$ ファイルに変換する XSLT の一部

が定められているからである。知識のなさということもあるが、XSLT の実装には、個々のテンプレート・ルールが実際にどのように適用されるかを確認しながらとなり、大変な作業となった。

この XSLT を用いて出力される $\text{T}_{\text{E}}\text{X}$ ファイルに、コンパイル ($\text{T}_{\text{E}}\text{X} \rightarrow \text{DVI}$) 時にエラーを無視するための命令 `\batchmode` を組み込む。エラーが起こるのは、 $\text{T}_{\text{E}}\text{X}$ における特殊文字 (%, \$ など) や機種依存文字 (ローマ数字, 半角カタカナなど) が入力データ中に存在する場合である。結果として、生

成される PS ファイルは文字化けが起こり、教員はこれを見て修正することになる。また、この命令を入れることでエラーを無視するので、サーバ側でコンパイルが止まることなく XML から PS ファイルの生成までのプロセスを終了させることができる。

また、XSLT プロセッサには Apache XML プロジェクトが提供する Java ベースの XSLT プロセッサ Xalan を使用した。Xalan を用いたのは、3 節で提案するように、JSP と Bean とサーブレットを使用する MVC モデルの適用を考え、将来的にシステム全体の開発言語を Java で統一するためである。これによりシステムの設計が容易になるとともに、開発効率、再利用性が考えられる。

2.4.4 DB と XML の変換

本システムは、データベースを中核に構成されていると述べたが、現段階では、データベースの設計を行っていない。入力データは XML 文書に変換された状態でサーバに保存されている。今後、カリキュラムが変更になったときの新旧の管理や JABEE モデルの提供、最大履修条件の反映、先行科目の確認などシステムを統合していく上で様々な要求があると考え、現在、データベースをどのように設計するか検討している。

2.5 評価

ここでは、XML を基にしたことによる設計・開発について評価する。

まず設計に関しては、XML のスキーマを定めることにより、Web 入力部は XML 文書を出力すればよく、提示部では XML 文書を入力とし目的に合った XSLT スタイルシートを作成すればよい。また、DB では XML 文書の格納と DB から XML 文書を生成できればよい。このように 3 つのパートに分けられ開発を分散化ができる。

以上のように、XML を基にしたことで、設計・開発作業が容易なものとなった。また、他のシステムとの連携を行なう準備が整った。

しかし、本システムは 1 ヶ月ほどで設計・開発・運用したこともあり、システムとしてはあまり評価できない。というのも、実際には、問題点を修正しながらの運用となってしまった。大きなトラブルはなかったものの、シラバスの登録・編集作業において指摘される問題が多かった。その問題点を以下にまとめた。

1. 思い通りのレイアウトができない
2. PS ファイル中の一部に文字化けが発生

これら 2 つの問題について解決方法を述べる。

2.5.1 問題点 1. 思い通りのレイアウトができない

システムを開発する上で、どんなレイアウト機能を持たせるかが最も重要なポイントとなった。なぜなら、レイアウトの意味も XML で表現するため、機能をまとめないと XML のスキーマが決まらない。ここでのレイアウト機能とは、難しいものではなく、改行の扱いやリスト表示などである。我々が特に力を入れたのが、シラバスにとって最も重要となる授業計画の欄である。この欄では基本的に一回毎の授業のテーマを書くのであるが、授業によっては毎回のテーマを書けないものもある。そこで、「リスト形式」と、「自由形式」の 2 種類の入力フォームを用意することによって、この要望に答えた。

しかし、この入力システムは $\text{T}_{\text{E}}\text{X}$ を使ったことがない方にとって、非常に使いにくいものとなった。例えば、スペースである。 $\text{T}_{\text{E}}\text{X}$ では連続したスペースは 1 つのスペースと見なされるので、思い通りに編集できない。また $\text{T}_{\text{E}}\text{X}$ の特殊文字を誤って入力し、生成される PS ファイルは文字化けしてしまう。逆に、 $\text{T}_{\text{E}}\text{X}$ を使ったことがある方の中には、命令（例えば、 $\backslash\text{TeX}$ ）などを使って見栄えよくする方もいた。

以上から、より簡単な入力とレイアウトが求められ、また、 $\text{T}_{\text{E}}\text{X}$ は様々な命令や環境を用いることによってはじめて美しい文書を作成できるので、もう一度必要な機能を洗い出し XML のスキーマを再検討する必要がある。

2.5.2 問題点 2. PS ファイル中の一部に文字化けが発生

これは入力の中に、 $\text{T}_{\text{E}}\text{X}$ の特殊文字、ローマ数字や丸囲み数字などのいわゆる「機種依存文字」や半角カタカナが含まれることによる。今回サーバの OS は Linux であり、サーバで変換処理を行なうので、特に Windows の機種依存文字の文字化けが目立った。

この問題を解決するには、半角カタカナに関しては、全角カタカナでは文字化けは起こらないので、全角カタカナに置換する。これは日本語文字コード変換ライブラリ「jcode.pl」を用いて置換で

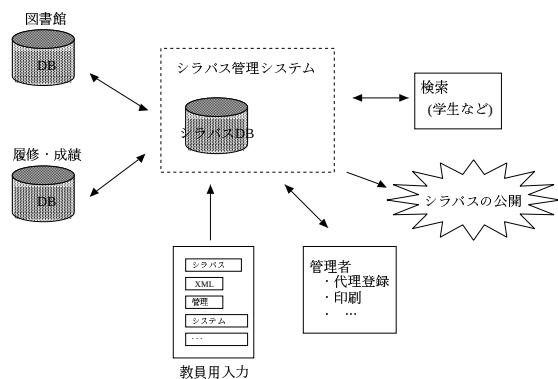


図 6: 今後のシラバス管理システムの構成

きる。また $\text{\TeX}$ の特殊文字と機種依存文字に関しては、現段階では正規表現を利用することを考えている。置換の対象となる文字をあらかじめ洗い出し、置換規則を用意する。入力データに対してパターンにマッチする文字を探し出し、見つければ置き換えるのである。具体的に、ローマ数字の2であれば、“II”（Iを二つ並べる）に置換する。

3 今後の課題と拡張

今後の課題としては、来年度のシラバスの登録（1月中旬）までに、2.5 で取り上げた登録・編集作業の際の問題点を解決する。特に、入力者が操作を迷うことのないよう、入力に伴う説明、エラー処理を徹底して行なう必要がある。

また、シラバスには、カリキュラムが変更になったときの新旧の管理、JABEE モデルの提供が必要であり、他のシステムとの連携も考慮に入れ、シラバスのデータベースを設計する必要がある。今後、シラバス管理システムは図 6 のような構成となると考えられ、

- 履修者に休講通知などのメール配信
- よくある質問を Web 上のシラバスに反映
- 図書館に授業の参考書があるか確認

など、様々な要望を実現していかなければならない。

そして、シラバス、レポートなど教育に関する情報を Web 上で提示する機会が増えれば、学生は携帯電話などのモバイル端末からアクセスすると考えられる。アクセスする端末を判別し、それに見合った形式で提示を行なうことが求められる。データを XML で表現することで、端末に応じた XSLT を用意すればよい（図 7）。このように幅広い拡張が求められる。

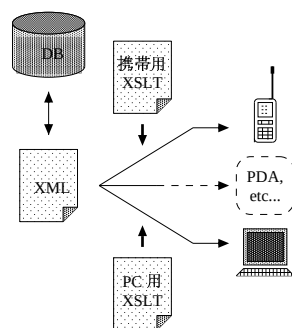


図 7: XSLT を活用した提示の例

そこで、今後のシステムの拡張のことを考え、Web システムを今まで CGI で行なっていたのを JSP とサーブレットに切り替えることと、提示に関しては、XSL を用いることも検討している。

3.1 JSP とサーブレットによる Web システム構築の提案

今後、教育システムに様々なサービスが求められると考え、Web ページ・デザイナーと Java プログラマーの分業化やシステムの拡張/再利用などをより高いレベルで実現する必要がある。JSP と Bean とサーブレットを使用する MVC モデルを適用することを考えている。

3.2 XSL

XSL とは、XSLT（XML 文書の変換を行なうための言語）と XSL-FO（印刷/表示のためのレイアウト記述言語）の 2 つの仕様により構成されている（参考文献 [2]）。Apach XML プロジェクトの XSL-FO プロセッサ FOP などを使用して印刷/表示を行なう。FOP は、現在、出力フォーマットとして PDF や PostScript, SVG などに対応している。

この XSL を用いるメリットは、 $\text{\TeX}$ と XML の文法の違いを気にしないで済む。例えば、シラバスの XML のスキーマを XSL-FO のタグセットに従って決められる。また、入力データ中の $\text{\TeX}$ の特殊文字を気にする必要がなくなる。しかし、仕様が複雑で、2001 年 10 月に W3C 勧告になったばかりということもあり、レイアウトのためのオーサリングツールがあまりないなどのデメリットがある。

4 まとめ

本稿では、教育システムのひとつであるシラバス管理システムを、既存の他のシステムの連携を考

え，共通モデルとして XML を基にして実現し，その概要を報告した．特に，シラバスデータの移植性の高さ，提示への変換など XML を基にしたことの有用性が確認された．

今後は，Web システムを，JSP と Bean とサーブレットを使用する MVC モデルを適用し，その実装検証を行なうほか，データベースの設計，XSL を用いてシラバスを印刷用に加工する予定である．

参考文献

- [1] Extensible Markup Language (XML) 1.0
(Second Edition),
“<http://www.w3.org/TR/REC-xml>”
W3C, 2000 年 10 月 6 日.
- [2] Extensible Stylesheet Language (XSL)
Version 1.0,
“<http://www.w3.org/TR/xsl>”
W3C, 2001 年月 10 日 15.
- [3] XSL Transformations (XSLT) Version 1.0,
“<http://www.w3.org/TR/xslt>”
W3C, 1999 年 11 月 16 日.