



XSL-FO를 이용한 XML문서의 스타일 정의

임순범*

● 목 차 ●

- 1. 서론
- 2. 웹 스타일 언어의 특징
- 3. XSL-FO 개요
- 4. XSL-FO 적용 예제
- 5. 결론

1. 서론

최근 웹 기술이 발전하고 XML(eXtensible Markup Language) 응용 분야가 확대됨에 따라 다양한 스타일의 웹 문서가 사용되고 있다. 전자 상거래를 위한 문서의 포맷 뿐 아니라 전자책 문서의 스타일 정의까지 필요로 하고 있다[1].

XML 문서를 웹 브라우저 화면에서 출력하여 보기 위해서는 스타일 파일을 적용하여 출력 포맷팅을 해야 한다. 스타일 표현 언어에 따라 기존의 웹 브라우저에서처럼 간단한 레이아웃부터 정교한 전자출판물 수준의 레이아웃까지 표현할 수 있다.

본 논문에서는 우선 웹에서 사용되는 스타일 언어들의 특징을 비교해 보고 특히 XSL-FO(eXtensible Stylesheet Language - Formatting Objects)에 대하여 집중적으로 살펴보기로 한다. 실제 출력 예를 중심으로 XSL-FO를 적용하여 XML 문서의 스타일을 정의하는 방법을 설명하고자 한다.

2. 웹 스타일 언어의 특징

기존의 HTML 문서는 자체적인 스타일 정보를 포함하고 있기 때문에 그대로 사용하거나 혹은 CSS(Cascading Style Sheets)를 적용할 수도 있다. XML 문서에서는 CSS 또는 XSL-FO를 적용하여 스타일을 지정하고 있다.. 일반적으로 웹 상에서 간단한 구조의 문서를 표현하기 위해서는 CSS를 적용하고, 구조가 복잡하거나 전문화된 문서의 포맷팅을 요구하는 환경에서는 XSL-FO를 적용하게 된다.

2.1 CSS(Cascading Style Sheets)

웹 표준화 단체인 W3C(World Wide Web Consortium)에서는 1996년 웹 문서의 출력 형태를 쉽고, 효율적으로 정의할 수 있도록 스타일 언어인 CSS1 (CSS level 1)을 제정하였다. 초기의 CSS1은 내용과 스타일이 혼재하여 복잡해지고 있는 HTML 문서에서 논리적 내용과 스타일을 분리할 수 있는 방법을 제시한 것이다[4].

처음 발표된 CSS1은 HTML문서를 웹 환경에서 출력하기 위한 스타일 정보를 정의한 것으로 글꼴, 색상, 행간 등의 정보를 표현할 수 있다. CSS1은 HTML에서 제공하는 스타일 정보와 일부 중복되지

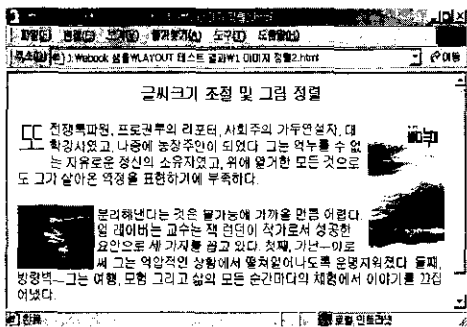
* 건국대학교 컴퓨터학과 교수

만 효과적인 스타일 표현이 가능하다.

다음으로 1998년에 제정된 CSS2(CSS level 2)는 페이지 기능, 표, 위치지정, 청각 스타일 등 CSS1에 비해 좀 더 정밀한 스타일 정보 표현이 가능하며, HTML 뿐 아니라 XML 등 다른 문서 형식에서도 사용 가능하다. 그러나 한글 환경에서 필요한 세로 쓰기와 다단 편집 등의 기능은 지원하지 않고 있다[5].

CSS3(CSS level 3)는 웹 문서의 특징인 상호작용성 및 동적인 기능 등의 향상된 사용자 인터페이스를 정의하고, 세로줄 쓰기 등을 지원하고 있으나 아직까지 표준으로 확정되지 않고 사양 초안의 제정작업이 진행 중에 있다.

CSS의 가장 큰 특징은 HTML의 스타일 정보를 위해 제정이 되었으므로 적용이 간단하며 문법 자체도 어렵지 않다는 점이다. 또한 스타일만을 따로 관리함으로써 일관성을 대단히 높일 수 있다. CSS는 HTML뿐 아니라 XML에서도 사용 가능하며 대부분의 브라우저에서는 CSS1 또는 CSS2의 일부를 지원하고 있으며 적용 예는 다음 (그림 1)과 같다. 그러나 그 자체로써는 큰 단점이 아니지만 문서의 구조 자체를 변경하거나 확장할 수 없다는 점은 XSL과 비교해 볼 때 단점으로 부각되기도 한다.



(그림 1) CSS 적용 예

2.2 XSL(eXtensible Stylesheet Language)

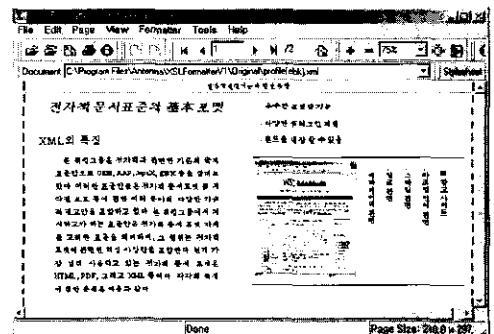
CSS가 HTML 환경에서 손쉽게 스타일을 작성할

수 있도록 한 것으로 자체적인 문법 형식을 갖고 있다. 이에 반해 XSL은 XML의 발전과 함께 XML 문서의 출력의 방법을 정의하기 위해 제정된 언어로 XML 형식으로 표현된다[6].

W3C에서 정의한 XSL 사양은 두 가지로 구성되어 있다. 하나는 XSLT(XSL Transformations)로 XML 문서를 다른 형식의 문서로 변환해주는 웹 언어이며[7], 다음은 XSL-FO(XSL Formatting Objects)로서 문서의 출력 포매팅 내용을 지정해주는 XML 태그의 집합이다[6]. XSLT는 W3C에서 1999년 1.0버전이 발표되었으며, XSL-FO는 2001년 상반기 현재 표준으로 확정되기 바로 전 단계에 있으며 조만간 웹 표준으로 발표될 것으로 예상된다.

XSL은 SGML에서 스타일 정보를 표현하는 언어인 DSSSL(Document Style Semantics and Specification Language)과 CSS를 기반으로 만들어졌다. DSSSL은 오프라인 출력을 주된 목적으로 만들어진 언어로, 출력 스타일에 대한 정밀한 제어가 가능하지만 기능이 복잡하여 구현하기가 어렵다 [3]. 반면에 CSS는 웹 상의 화면 출력을 목적으로 만들어진 언어로 출력 스타일에 대한 제어가 DSSSL만큼 정밀하지 못하다.

이에 비해 XSL-FO은 DSSSL과 CSS의 중간 수준으로 CSS 보다 더 세련되고 복잡한 편집 디자인이 가능하여 쪽모양, 문단모양, 글자모양, 정교한 스타일 지정 등 전자책에서 필요한 기본적인 스타일 기



(그림 2) XSL 적용 예

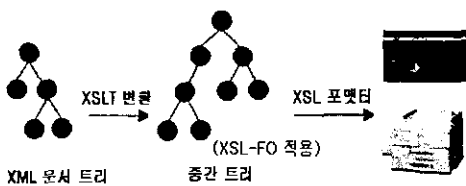
능을 제공한다. 또한 CSS에서는 지원되지 않는 세로쓰기, 다단편집, 한영 문자 줄맞춤, 각주(footnote) 등의 한글 문서환경도 제대로 처리할 수 있다.

CSS는 단순 웹 화면에 주로 사용되며 XSL-FO는 보다 넓은 분야에 활용되는데 전자출판 수준의 화면 출력이 가능하여 인쇄된 책 한 권의 편집 디자인도 가능하다. (그림 2)는 XSL을 적용한 예이다.

3. XSL-FO 개요[6]

3.1 XSL 처리 과정

XSL 프로세서가 XML 문서와 XSL 스타일시트를 받아들여 출력결과를 얻는 과정은 일반적으로 (그림 3)과 같이 변환과 포매팅의 두 가지 단계를 거친다. 첫 번째 단계는 XSLT 트리 변환으로서 XML 문서에 대한 트리를 출력을 위한 트리, 즉 포매팅 객체 트리로 변환한다. 두 번째 단계로 XSL-FO 포맷터가 중간 객체 트리를 이용하여 XSL-FO 요소로 지정된 스타일 결과대로 화면, 종이, 음성, 또는 기타 매체에 해당하는 출력을 한다.



(그림 3) XSL 처리 과정 : 변환과 포매팅

XSL-FO의 포매팅 모델은 포매팅 객체 트리의 각 노드에 해당하는 스타일의 의미를 부여하여 작성하게 되는데, 스타일 적용 대상인 각 노드를 포매팅 객체(formatting object)라고 한다. 포매팅 객체를 표현하는 데에는 쪽, 문단, 표 등의 56가지 XSL-FO 요소(element)가 있고 이들에 대해 스타일 값을 지정하는 다양한 속성들(properties)이 있다.

3.2 XSL-FO 포매팅 객체 요소

XSL-FO 요소의 name-space는 fo: 로서 XSL-FO 태그는 <fo:객체요소 속성=값 ... > 와 같은 형태를 가지며, 다음과 같이 관련되는 몇 개의 그룹으로 나누어 볼 수 있다.

3.2.1 선언 및 쪽 모양 포매팅 객체 (19개)

- fo:root, fo:page-sequence, fo:page-sequence-master, fo:single-page-master-reference, fo:repeatable-page-master-reference, fo:repeatable-page-master-alternatives, fo:conditional-page-master-reference, fo:layout-master-set, fo:simple-page-master, fo:region-body, fo:region-before, fo:region-after, fo:region-start, fo:region-end, fo:declarations, fo:color-profile, fo:flow, fo:static-content, fo:title

3.2.2 문단 포매팅 객체 (2개)

- fo:block, fo:block-container

3.2.3 글줄 포매팅 객체 (10개)

- fo:bidirectional-override, fo:character, fo:initial-property-set, fo:external-graphic, fo:instream-foreign-object, fo:inline, fo:inline-container, fo:leader, fo:page-number, fo:page-number-citation

3.2.4 표 포매팅 객체 (9개)

- fo:table-and-caption, fo:table, fo:table-column, fo:table-caption, fo:table-header, fo:table-footer, fo:table-body, fo:table-row, fo:table-cell

3.2.5 리스트 포매팅 객체 (4개)

- fo:list-block, fo:list-item, fo:list-item-body, fo:list-item-label

3.2.6 링크 및 다중 포매팅 객체 (6개)

fo:basic-link, fo:multi-switch, fo:multi-case,
fo:multi-toggle, fo:multi-properties,
fo:multi-property-set

3.2.7 기타 포매팅 객체 (6개)

fo:float, fo:footnote, fo:footnote-body, fo:wrapper,
fo:marker, fo:retrieve-marker

3.3 XSL-FO 스타일 속성(properties)

XSL-FO 스타일 속성은 매우 다양하며 여러 가지 포매팅 객체에 공통으로 사용되는 것과 특정 부류의 포매팅 객체에만 적용되는 것이 있다.

3.3.1 공통으로 사용되는 속성의 그룹

- Accessibility 속성 : source-document, role속성
- Aural 속성 : 음성으로 들려줄 때의 속성
- Border, Padding, and Background 속성 : block 및 inline의 영역 둘레, 테두리, 배경을 지정
- Font 속성 : 폰트 선택 시 사용되는 속성
- Hyphenation 속성 : language, script, country에 따라 hyphenation을 제어
- Margin-Block 속성 : block-level의 포매팅 객체의 spacing 과 indent를 지정
- Margin-Inline 속성 : inline-level의 포매팅 객체의 spacing을 지정
- Position 속성 : 포맷된 영역의 절대위치 및 상대위치를 지정

3.3.2 Page Layout에 사용되는 속성의 그룹

- Pagination & Layout 속성 : 쪽의 크기, 방향, 순서, 쪽 마스터 영역, 다단 등을 지정
- Layout-related 속성 : clip, overflow, span, reference-orientation 등의 값을 설정
- Area Dimension 속성 : block 및 inline 영역의 치수를 지정

- Area Alignment 속성 : inline영역의 줄맞추기
- Color-related 속성 : color, color-profile-name, rendering-intent를 설정
- Keeps and Breaks 속성 : 쪽, 단, 줄의 keeps 와 breaks에 대한 특성을 지정
- Writing-mode-related 속성 : block 및 inline에 대한 글의 진행 방향 속성을 지정

3.3.3 Block 및 Inline 객체에 적용되는 속성 그룹

- Block and Line-related 속성 : line-area의 구성과 block-area 및 line-area의 배치를 지정
- Character 속성 : word-spacing, letter-spacing, suppression 등 문자에 관한 속성을 지정
- Leader and Rule 속성 : leader와 rule의 구성
- Area Alignment 속성 : 서로 다른 언어의 inline 줄맞추기를 지정

3.3.4 기타 속성의 그룹

- Table 속성 : 표의 레이아웃 포매팅을 설정
- Markers 속성 : header나 footer에 사용되는 marker의 특성 및 검색 속성을 지정
- Float-related 속성 : side와 before float의 배치
- Dynamic Effects 속성 : link 및 기타 dynamic effects의 표현과 action을 설정
- 숫자-문자 변환 속성 : 쪽번호 또는 기타 포매팅 번호의 구성에 사용
- Miscellaneous 속성 : content-type, id, src, visibility 등 다른 그룹에 속하지 않는 속성
- Shorthand 속성 : 개별 속성으로서 확장하여 사용할 수 있다.

4. XSL-FO 적용 예제

한국전자책 권소시움에서 정의한 전자책 표준 DTD에 따라 잡지 1종(샘터 2001년 3월호 일부)과

학습서 1종(정훈사 공통영어)에 대하여 XML 문서로 구성하고, XSL-FO를 적용하여 스타일 파일을 작성하였다[1]. 이 전자책 예제를 중심으로 XSL-FO 적용방법을 작성 단계별로 설명하도록 한다. 예제 출력에 사용된 브라우저는 Antenna House XSL Formatter V1.0으로서 현재 약 80% 가량의 XSL-FO 기능을 지원하고 있다[2].

4.1 시작

우선 XSL 파일도 하나의 XML 문서이므로 파일의 맨 앞에 XML 문서 표시가 있어야 한다. 이어서 XSL에 대한 name-space를 명시해 준다.

포매팅 객체를 정의할 때는 최상위 객체로서 fo:root 객체를 정의하고, 그 안에는 문서 전체적으로 적용되는 스타일에 대해 fo:layout-master-set 와 같이 해당하는 객체들을 이용하여 정의해 주면 된다. XSLT 변환을 적용할 경우는 template match를 위하여 해당 객체의 경로를 적어준다. fo:block 과 같이 독립적으로 적용되는 부분들 역시 해당 객체에 template match를 시켜주면 된다.

```

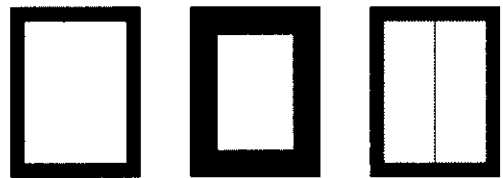
<?xml version="1.0" encoding="utf-16" ?> 시작
<xsl:stylesheet name-space
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fo="http://www.w3.org/1999/XSL/Format"
  version="1.0">

<xsl:template match="ebook">
  <fo:root ... > root 객체
    <fo:layout-master-set> ... fo: 객체
  </fo:root>
</xsl:template>

<xsl:template match="ebook/books/book/front/intro/title">
  <fo:block font-size="11pt" ...><xsl:apply-templates />
  </fo:block> ... fo: 객체
</xsl:template>

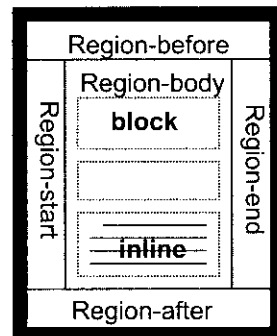
```

4.2 쪽 모양 (Page Layout) 마스터 정의
fo:layout-master-set 에서는 여러 종류의 쪽 모양들의 집합을 정의하고 fo:page-sequence 에서 쪽의 순서 및 들어갈 내용의 모양을 정의해 준다. 쪽 모양 마스터는 fo:simple-page-master 요소로 정의하는데 예제 코드에서는 아래 그림과 같이 “표지, 머리말, 2단본문” 3가지에 대하여 쪽크기, 여백, 본문 영역, 테두리, 배경색, 단의 개수 등을 정의하였다.

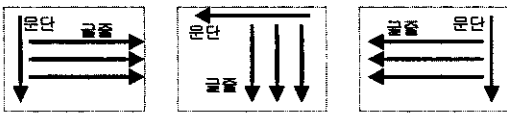


(그림 4) 쪽 모양의 예 : 표지, 머리말, 2단본문

본문 영역(region)을 정의하는데는 (그림 5)에서와 같이 5가지 구분의 영역(region)이 있다. 상하좌우 가운데로 각각 fo:region-before, fo:region-after, fo:region-start, fo:region-end, fo:region-body 요소에서 원하는 크기와 여백, 배경색, 테두리 등을 정해 준다. 그리고 글쓰기 방향(direction)도 설정하는데 문단(block) 및 글줄(inline) 진행 방향을 별도로 설정할 수 있다. 이에 따라 세로쓰기, 위에서 좌로 쓰기는 등 모든 경우의 조합으로 글쓰기 방향 설정이 가능하다.



(그림 5) 본문 영역



(그림 6) 글쓰기 방향(direction)의 예

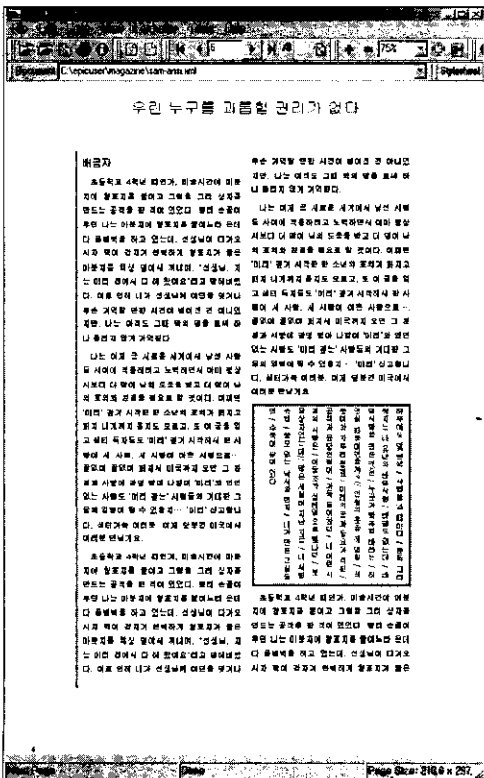
본문 영역 내에는 문단 또는 개별 객체 단위로 내용을 포함하도록 되어 있고, 이들 개별 내용에 대한 스타일 속성은 별도의 template으로 지정해 주면 된다.

```

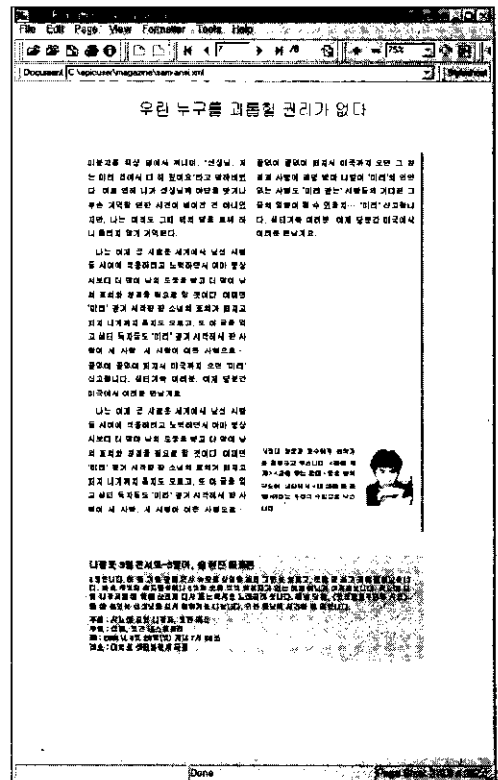
<fo:layout-master-set>
  <fo:simple-page-master master-name="표지"
  <속성 page-height="240mm" page-width="160mm"
  margin-top="20mm" margin-right="20mm"
  margin-bottom="20mm" margin-left="20mm" ... >
  
```

```

</fo:simple-page-master> <머리말 master
<fo:simple-page-master master-name="머리말"
page-height="24cm" page-width="16cm" ... >
<fo:region-body margin-top="30mm" <속성
<본문영역 margin-right="30mm" margin-bottom="30mm"
margin-left="30mm" ... padding="6pt" />
<fo:region-before extent="25mm"
border-after-style="solid" border-after-width="thin"
display-align="after" />
<fo:region-after extent="25mm"
border-before-style="solid" border-before-width="thin"
display-align="before" />
<fo:region-start extent="25mm"
border-end-style="solid"...display-align="after" />
<fo:region-end extent="25mm"
border-start-style="thin" ... writing-mode="tb-rl" />
</fo:simple-page-master> <2단본문 master
<fo:simple-page-master master-name="2단본문"
  
```



(그림 7) 출력 예 : 수필 (2단 본문)



(그림 8) 출력 예 : 수필, 저자소개 (표)

```

page-height="12cm" page-width="8cm" ... >
<fo:region-body margin-top="30mm"
  margin-right="30mm" margin-bottom="30mm"
  margin-left="30mm" ... column-count="2" />
<fo:region-before extent="30mm" 다단편집속성
  ... column-count="1" />
<fo:region-after extent="25mm" ... />
<fo:region-start extent="25mm" ... />
<fo:region-end extent="25mm" ...
  display-align="before" writing-mode="tb-rl" />
</fo:simple-page-master>
</fo:layout-master-set>

```

4.3 쪽 순서 및 흐름 설정

fo:root 객체 내에 쪽의 순서 등을 fo:page-sequence 요소에서 정의해 준다. fo:page-sequence 내에서 각 쪽에 고정적으로 들어가는 내용의 스타일은 fo:static-content 요소에서 정의해 주고, 쪽 모양의 크기에 따라 자동으로 길이를 절단해주는 내용의 스타일은 fo:flow 요소에서 정의해 주면 된다.

```

쪽순서 설정
<fo:page-sequence master-name="머리말" format="1">
  <fo:static-content flow-name="xsl-region-after">
static내용 <fo:block text-align="center" font-size="small">
    <fo:page-number /> </fo:block>
  </fo:static-content>
flow내용 <fo:flow flow-name="xsl-region-body" line-height="1.7">
    <xsl:apply-templates
      select="books/book/front/intro/p[1]" />
    <xsl:apply-templates
      select="books/book/front/intro/title" />
  </fo:flow>
</fo:page-sequence>

```

4.4 개별 스타일 속성 설정

본문 영역 내에서는 문단(block) 단위 또는 문장의 글줄(inline)에서 개별 객체 단위로 스타일을 정의한다. 우선 스타일을 적용할 내용에 대해

template match로 지정하고 font, background, border, padding, margin, alignment, Indentation 등 개별 속성의 값을 정의해 준다.

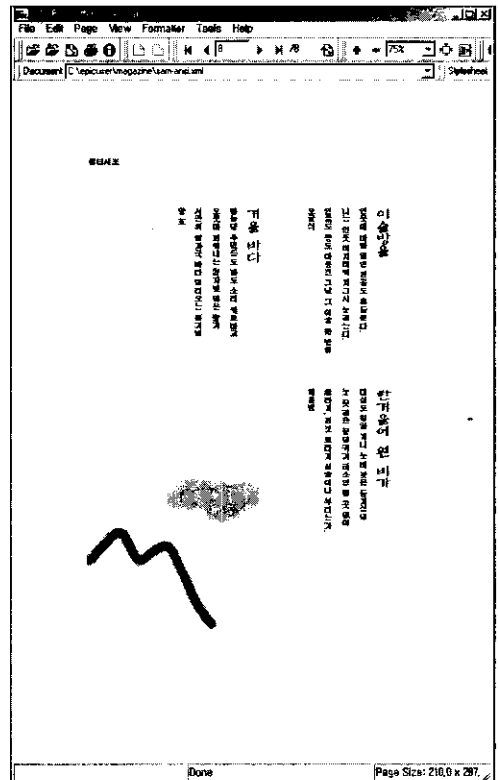
```

대응되는 문단
<xsl:template match="ebook/books/book/front/intro/title">
  <fo:block font-size="25pt" font-family="신명조"
    space-before="4em" ... text-align="left">
    <xsl:apply-templates /> </fo:block> 문단 속성 설정
</xsl:template>

```

4.5 표, 목록 등 정의

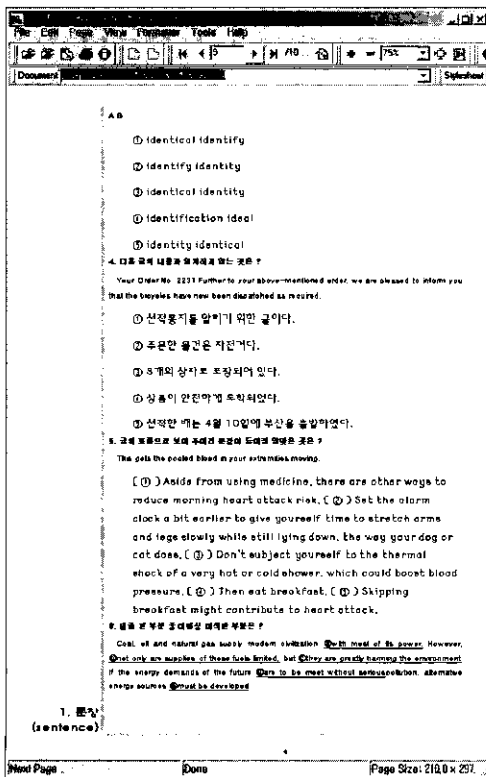
마지막으로 목록이나 표에 대해서 물리적인 스타일을 지정하였다. 목록의 스타일에 대해서는 fo:list-block, fo:list-item, fo:list-item-label, fo:list-item-body 요소를 이용한다. 표의 스타일 작성은



(그림 9) 출력 예 : 시 (다단 세로쓰기)

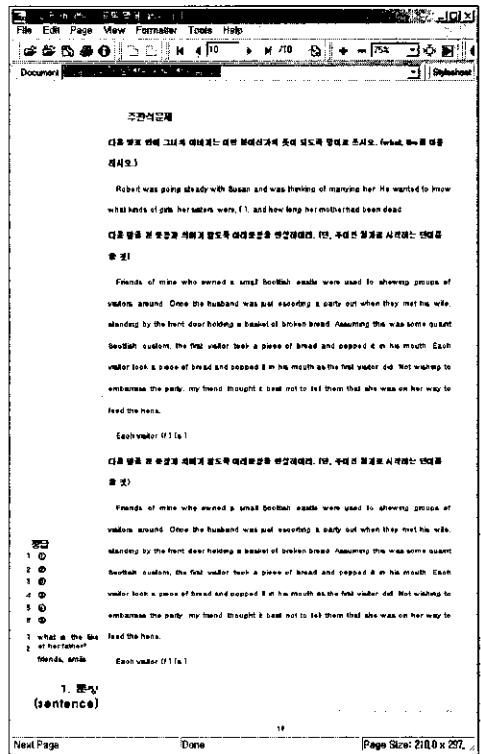
fo:table, fo:table-column, fo:table-header, fo:table-body, fo:table-row, fo:table-cell 등의 요소를 이용하여 포매팅 속성을 지정한다. 표의 각 셀에 들어가는 내용은 fo:block 요소를 이용하여 스타일 속성을 지정하면 된다. 예제에서 수필의 저자소개나 광고는 표를 이용하여 작성하였다.

```
<xsl:template match="book/body/part[1]/chapter[2]/author/p">
  <fo:table border="0pt solid black" text-align="center"
  표 border-spacing="3pt">
    <fo:table-column column-width="1.8in" /> 열
    <fo:table-column column-width="1in"
      number-columns-repeated="1" />
    <fo:table-body>
      <fo:table-row> 행
      <fo:table-cell padding="6pt" border="0pt solid blue"
  cell의 내용 number-rows-spanned="1">
```



(그림 10) 출력 예 : 학습지 객관식 문제

```
<fo:block text-align="justify"
  padding-before="0" font-size="8pt">
<xsl:apply-templates />
</fo:block>
</fo:table-cell>
<fo:table-cell padding="6pt" border="solid black">
  cell의 내용 <fo:block> <fo:external-graphic src="auth2.jpg"
  content-height="60%" content-width="60%" />
</fo:block>
</fo:table-cell>
</fo:table-row>
</fo:table-body>
</fo:table>
</xsl:template>
```



(그림 11) 출력 예 : 단문 및 주관식 문제

5. 결론

XML문서의 포매팅 정의를 위한 스타일 언어를

살펴보았다. CSS1, CSS2와 XSL-FO의 특징을 비교하였고, XML 문서 스타일시트 예제를 통하여 XSL-FO의 적용방법을 살펴보았다. XSL-FO는 단순 웹 문서뿐만 아니라 전자책과 같은 고품위 문서에 대한 출력스타일의 정의도 가능함을 보여 주었다. 특히 한글 문서 환경을 제대로 반영한 스타일 시트 작성은 XSL-FO를 적용하면 가능해 진다.

금명간 XSL-FO가 웹 표준으로 확정될 전망이므로 이에 대하여 여러 가지 준비가 필요할 것이다. 다양한 목적으로 사용될 XSL-FO 출력 포맷터 혹은 브라우저의 개발이 가장 시급하다. 현재 XSL-FO를 완전히 지원하는 웹 브라우저가 아직 시판되지 않고 있으며 제한된 기능을 지원하는 브라우저는 일부 나와있다[2].

그리고 이에 못지 않게 XSL-FO 스타일 편집기가 필요할 것이다. 전문적인 문서의 스타일시트는 보통 편집 디자이너가 작성하게 되는데, 이들이 XSL 태그를 이용하여 직접 코딩을 하기에는 많은 무리가 따른다. 따라서 XML 문서 편집기에 WYSIWYG 방식의 스타일 편집 기능을 추가하여 자동으로 XSL-FO 태그를 생성시키도록 하거나 기존의 문서 편집기에서 해당 포맷의 스타일을 XSL-FO 파일로 변환시켜주는 다양한 소프트웨어의 개발이 필요하다.

참고문헌

[1] 한국전자책컨소시엄, 한국 전자책 문서표준에 관한 연구, 2001.5.

[2] Antenna House, Antenna House XSL Formatter, <http://www.antenna.co.jp/>, 2001.

[3] ISO/IEC 10179, Document Style Semantics and Specification Language (DSSSL), <ftp://ftp.ornl.gov/pub/sgml/WG8/DSSSL/>, 1996.

[4] W3C, Cascading Style Sheets (CSS) level 1, <http://www.w3.org/TR/REC-CSS1-961217>, 1996.

[5] W3C, Cascading Style Sheets level 2 (CSS2), <http://www.w3.org/TR/1998/REC-CSS2-19980512>, 1998.

[6] W3C, Extensible Stylesheet Language (XSL), <http://www.w3.org/TR/2000/CR-xsl-20001121/>, 2000.

[7] W3C, XSL Transformations (XSLT), <http://www.w3.org/TR/1999/REC-xslt-19991116>, 1999.

저자약력



임 순 범

1982년 서울대학교 계산통계학과 (이학사)
 1983년 한국과학기술원 전산학과 (석사수료)
 1992년 한국과학기술원 전산학과 (공학박사)
 1989년-1992년 (주)휴먼컴퓨터 이사/연구소장
 1992년-1997년 (주)삼보컴퓨터 부장
 1997년-현재 건국대학교 컴퓨터학과 조교수
 관심분야: 컴퓨터 그래픽스, 웹 멀티미디어 응용, 전자출판 (폰트, 전자책, 사이버 교재)