

---

# XSL-FO를 이용한 PostScript Converter에 관한 연구

유동석 · 최호찬 · 이진영 · 김차중  
한밭대학교 정보통신 전문대학원 컴퓨터공학과

## A study of PostScript Converter using XSL-FO

Dong-Suk Luy · Ho-Chan Choi · Jeon-Young Lee · Cha-Jong Kim  
Graduate School of Information & Communication, Hanbat National University  
E-mail : gidon301@hanmail.net

### 요 약

인터넷 사용중 인쇄를 하였을 때 만족스러운 결과를 가지지 못하는 것은 스크린 상의 page와 인쇄물의 page들이 다르게 나타나기 때문이다. HTML과 CSS는 브라우저 기반의 정보를 표현하는데 있어 적합하고 충분하지만, 인쇄기 반의 정보를 표현하는데는 부족하다. 그래서 사용되는 것이 XSL-FO(XSL Formatting Object) 이다. 고품질의 출력과 출력장치에 무관한 인쇄에 있어서 PostScript가 사용되고 있다. 본 논문에서는 XML-FA 와 Postscript의 장점을 이용해 사용자가 원하는 결과물을 갖는 것과, XML문서의 XML-FA 와 Postscript의 Segment간의 대응 관계를 분석하고 매 평시킴으로써 Postscript로 변환의 기술을 연구하였다. 변환 결과로 기존 인쇄물 보다 더 나은 결과물을 얻을 수 있었다.

### ABSTRACT

Web documents specified with HTML and CSS is displayed with high quality on the web browser. However, basically, any printed pages don't have the same quality of softcopy. The reason is HTML and CSS is not suit for printing. The XSL-FO(XSL-Formatting Object) is Formatting Language for imaging of web document and The PostScript is one of the most famous PDL(Page Description Language). To get high-quality pages, we propose the design of converter which translate XML-FO into PostScript format. Using the designed converter, we can get hardcopies with high quality.

### 키워드

XSL-FO, PostScript, page layout

## 1. 서 론

최근 웹 기술이 발전하고 XML(eXtensible Markup Language) 응용 분야가 확대됨에 따라 다양한 스타일의 웹 문서가 사용되고 있다. 전자 상거래를 위한 문서의 포맷에서부터 전자책 문서의 스타일 정의까지 필요로 하고 있다. 이에 따라 인쇄 방법이나 장치에 따라 달라지기도 하고, 원하는 모양의 인쇄물을 얻지 못하는 경우도 발생한다.

1985년에 발표된 PostScript기술은 프린트 및 이미지 처리의 표준 기술로서 많은 기업이나 출판사, 정부 기관에 의해 채용되었다. 현재의 Postscript기술은 복잡하고 세부화된 오늘의 프린팅 환경에 있어서 폭넓은 요구에 응할 수 있도록 최적화된 프린

팅 시스템으로 진화되었다.

Postscript언어로 구성된 파일은 프린터 장치(output device)에 상관없이, 컴퓨터의 플랫폼이나 OS(Operating System)에 관계없이 문서를 프린트할 수가 있다. 또한 이 파일은 Vector 그래픽 형식으로 확대나 축소 시에도 그림의 질이 일정하다는 장점을 가지고 있어서 Postscript는 문서가 정확하게 프린팅 되게 하고, 보다 완성도 높은 성능을 가질 수 있게 한다.

## II. 본 론

### 2. 기반 기술

## 2.1 XSL

XSL(eXtensible Stylesheet Language)은 XML에 내용과 표현의 분리라는 중요한 의미를 부여해주는 역할을 담당하고 있는 XML 기술 중의 하나이다. XSL은 1997년 개발되기 시작했으며 SGML의 DSSSL과 HTML의 CSS에 기반하고 있다. 이 기술은 XML스펙을 따르고 있으므로 XML과 같은 장점을 가진다. XML문서의 구조를 재구성하거나, 데이터를 추가하고, 프레젠테이션용으로 변환시키기 위해 사용한다. XSL은 크게 두 가지 기술로 나누어지는데 하나는 XSL-FO이며, 나머지는 XSLT이다. XSL-FO는 XML문서의 시각적 표현을 위해 사용되는 포매팅 언어이고, XSLT는 XML문서를 다른 형태의 문서로 변환하기 위해 사용되는 변환 언어이다.

## 2.2 DOM & SAX

DOM은 HTML이나 XML과 같은 문서를 관리하기 위한 프로그램 API로서, 파서에 의해 파싱된 문서를 메모리 내에서 트리 형태의 계층 구조로 저장하여 문서 내의 어떤 부분도 프로그래머가 자유자재로 검색할 수 있게 해주고, 특정 요소를 문서의 어느 곳에도 삽입, 수정, 삭제 할 수 있게 해준다. XML문서를 처리하는 데에는 DOM과 SAX 두 가지 방법이 있는데, 그중 DOM은 W3C가 공식적으로 인정하는 표준으로 XML문서 처리를 위해 다양하게 사용된다.

## 2.3 PostScript

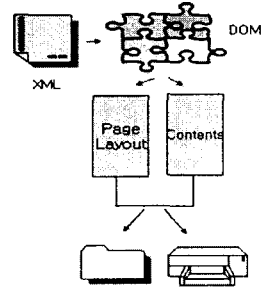
PostScript란 Adobe Systems Ltd.에서 개발한 인쇄용 언어로, PostScript 파일은 Vector 그래픽 형식으로 되어 있어 확대나 축소할 때 그림의 질이 일정하다는 장점이 있다. PostScript 파일은 PostScript 폰트가 내장된 프린터에서는 파일은 lpr 이나 copy 명령어를 이용하여(예: copy test.ps lpt1) 출력이 가능하나, PostScript 폰트가 내장되지 않은 대부분의 프린터에서는 GhostScript 등 인터 프린터를 이용하여 Image로 변환 과정을 거친 뒤, 인쇄가 가능하다. 현재로서는 GMT에서의 한글 구현은 지원되지 않기 때문에 생성된 PostScript 파일을 Photoshop, Goret Draw 및 Illustrator 등의 Graphic Software를 이용하여 편집하여야 한다.

### III. 시스템 설계

문서를 기술할 때 XML문서(XSL-FO)로 기술하게 되면 문서를 출력 디바이스에 관계없이 인쇄가 가능하다. 현재는 XSL-FO를 이용하여 XML문서를 PDF형태의 문서로 만드는 것을 가장 많이 사용하고 있고, XSL-FO에 대해 언급된 문서를 보면 앞으로는 각 문서 형식에서의 변환을 현재 우리가 사용

하고 있는 브라우저에서도 가능하게 될 것이라고 조심스레 전망하고 있다.

## 3.1 Page Layout



Postscript에서 사용되는 좌표 계는 2차원 좌표 계이다. Unit는 'pt'(point)를 사용하는데 1 point는 1/72 inch이다. 72points는 1 inch이다.

그림 1. 전체 시스템 구성

예를 들면 8.5×11 inch page는 612×792 points 넓이이다. 또한 이 page의 center는 306,396이 된다. layout될 전체 page의 size나 margin등의 정보를 받아 PostScript상에 배치 시켜 줘야 한다.

Page layout과 관련된 엘리먼트들은 layout-master-set, simple-page-master, page-sequence-master가 있고, layout-master-set는

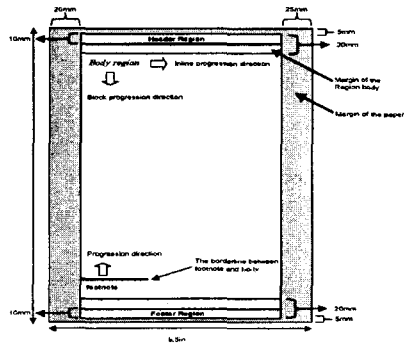


그림 2 The Page Layout

하나 이상의 simple-page-master를 가지고 있는데, 실제로 page layout을 지정하는 부분이다. 한 페이지에서 region과 margin의 관계를 정의 해야 하고, 각 region들은 서로간의 영역들을 겹치지 않게 적당한 간격을 띄워야 한다. 하나 이상의 마스터 페이지를 필요로 할 경우 page-sequence-master를 이용해서 간단하게 지정한다.

## 3.2 Content

Content를 배치하는 적당한 시작점으로 이동해 그

곳에서부터 content를 표시하는 것이다. PostScript는 page 넓이 보다 내용일 길더라도 새로운 line으로 이동하지 않는다. 현재 위치(currentpoint)가 여백을 뺀 page의 넓이 이상인지를 점검해서 새로운 line에서 시작하게 하는 프로시저를 생성해야 한다.

paragraph 사이 간격은 <fo:block>엘리먼트의 속성중 'space-after, optimum'의 값으로 계산되고, page number도 page의 하단 부에 배치시켜야 한다. 이외에도 paragraph시작 부위의 들여 쓰기를 추가할 수 있는데 이때도 PostScript의 좌표 계에 대응되는 값을 추가해야 한다.

또한 content의 내용이 가변적이기 때문에 다음 글의 위치를 동적으로 지정해 주어야 한다는 것이다. 이는 fo가 block단위로contents를 지정해 주기 때문이다. PostScript는 2차원 좌표 계를 사용한다는 점을 미리 언급한바 있다. PostScript에서 가지고 있는 현재 위치 바로 이점을 이용해서 이를 해결할 수 있다.

Text를 중앙으로 배열을 하기 위해서 먼저 해야 할 것은 PostScript의 X coordination 이 display될 시작점을 찾아야 한다. PostScript는 일반적으로 BoundingBox(default image area)로 612x792를 사용하고 있다. page의 중앙을 찾은 후, text의 길이를 얻어 오는데 이것은 text가 중앙을 기점으로 해서 좌우로 같은 길이만큼 배치된다는 점을 이용한 것이다.

### 3.3 인쇄

javax.print패키지의 클래스를 사용하여 인쇄 포맷 종류를 선택하고, 인쇄에 요청할 속성들을 선택한다. 인쇄 작업 이름, 인쇄용지 크기, 매수 등을 지정할 수 있고, 요청을 처리할 수 있는 사용 가능한 인쇄 서비스를 검색한다.

## IV. 시스템 구현

본 시스템은 IBM-PC호환 컴퓨터 (Pentium4, 800M)에서 개발하였으며, 운영체제는 Windows 2000 Professional을 사용하였고, 개발 언어는 Java SDK1.4.1을 사용하여 개발하였다. 파서는 apache사의 Xerces-1\_4\_4를 사용하였고, FO파일을 얻기 위해 XT XSLT Processor(엔진)를 사용하였다.

Converter에 대한 전체적인 구성은 변환하고자 하는 문서를 텍스트 창에서 확인하고, 문서의 구조를 쉽게 파악하기 위해 Object Viewer로 엘리먼트, 속성,내용들을 트리 형태로 확인한다. 문서의 수정이 필요한 경우 Editor를 이용해 엘리먼트를 하위 삭제, 추가할 수 있고, 내용 수정도 할 수 있도록 하였다. 수정을 마친 파일은 Conversion module를 이용해 PostScript 파일로 변환할 수 있도록 구성하였다. 변환된 파일은 Printing module에서 인터페이스를 이용해 원하는 모양, 인쇄 서비스 이름등

일반적인 속성들을 설정한 후 인쇄하도록 하였다.

### 4.1 GUI

PostScript Converter는 XSL-FO 문서를 well-formed xml문서의 한 형태임을 생각하고 설계하였고, PostScript Converter는 문서의 구조 및 문법을 잘 모르는 사용자라고 하여도 쉽게 사용할 수 있도록 하였다. 변환 하고자 하는 파일을 선택해서 왼쪽의 텍스트 창에서 원문 내용을 확인하면서 수정 도할 수 있다. 메뉴 항목으로 파일, 편집, 인쇄, 보기, 도움말을 사용하였고, 각각의 하위 메뉴를 추가하였다. Conversion module을 이용해 PostScript형태로 쉽게 변환해서 오른쪽 텍스트 창에 나타나도록 하였고, 변환된 파일은 인쇄 메뉴를 이용해 프린트 할 수 있다.

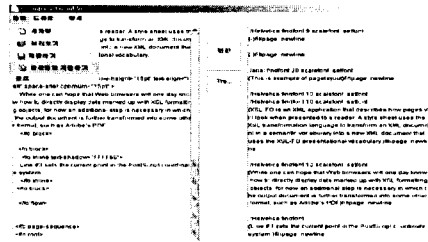


그림 3 사용자 인터페이스 화면

### 4.2 Conversion module

파싱된 문서는 각 type별로 해당 변환을 실행한다. 원문은 block 엘리먼트를 중심으로 내용이 기술되기 때문에 block 엘리먼트의 속성을 중심으로 변환을 하였고, 두 문서가 페이지 기술 언어이기는 하나 상이한 점이 많은 관계로 필요한 속성이나 segment가 없는 경우 일반적으로 사용되는 값을 첨부하였다. [2]

### 4.3 Object Viewer

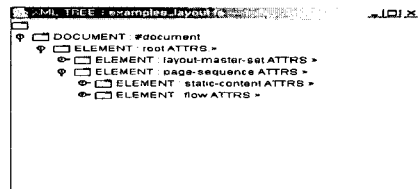


그림 4 Object Viewer 화면

XSL-FO는 XML형식을 사용하고 있고, 문서의 내용(엘리먼트, 속성, 속성값, txt)들을 텍스트 창에서 보는 것보다 tree형태로 보게 되면 문서의 구조를 쉽게 파악할 수 있기 때문에 Viewer를 제공해서 문서의 형식을 잘 알지 못하는 사용자도 쉽게 이해할 수 있도록 인터페이스를 제공하였다. 그림

은 Object Viewer화면이다.

#### 4.4 Editor

사용자의 효율적인 사용을 위해 원문 수정이 필요한 경우가 발생할 수 있기 때문에 에디터를 추가하여 원문을 수정할 수 있도록 인터페이스를 제공하였다. 에디터 왼쪽의 트리 구조에서 마우스의 오른쪽 버튼을 이용해 자식 노드를 추가, 삭제할 수 있게 하였고, 오른쪽 창에 텍스트 데이터의 입력을 할 수 있어 내용 편집이 가능하다. 다음은 편집기의 화면이다.

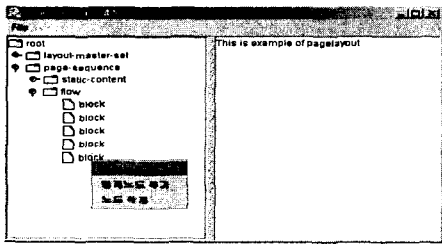


그림 5 원문 편집기

#### 4.5 printing module

다음 그림에서 볼 수 있듯이 변환된 파일을 출력할 수 있는 기능을 제공하였고, 속성을 선택할 수 있는 메뉴를 제공해서 사용자가 원하는 형태의 출력을 얻을 수 있도록 하였다. 원하는 인쇄 서비스를 선택해서 페이지를 설정하고 인쇄될 페이지의 모양도 사용자가 원하는 대로 선택할 수 있도록 구현하였다. 다음은 인쇄 메뉴 화면이다.

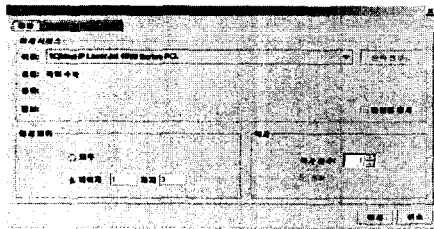


그림6 인쇄 메뉴

### V. 결론

본 연구에서는 XML-FO를 PostScript 포맷으로 변환하는 Converter를 설계하였다. 따라서 기존의 인쇄물 보다 고품질의 인쇄물을 얻을 수 있고, 또한 간단한 인터페이스를 이용해 사용자가 쉽게 문서를 편집, 확인할 수 있었다.

다음은 Converter로 변환을 실행한 파일을 PostScript Viewer인 GhostScript로 읽어 들인 화면이다.

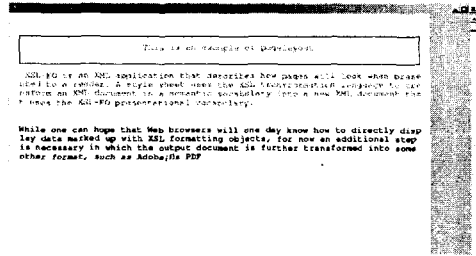


그림 7 변환된 문서

지금까지 설명한 XML 파일의 PostScript로의 변환은 최적의 인쇄물을 얻고자 하는 목적에 대한 시작점에 불과하다. 페이지를 기술하는 부분에 있어서 XML-FO는 많은 부분을 다루고 있으나 동적인 방법으로서의 접근은 이뤄지지 않고 있다.

웹상에서의 인쇄는 대부분 웹 브라우저에서 이뤄진다. 만약 브라우저에 XML파서와 DOM/SAX 파서가 지원된다면 얼마든지 PostScript로의 변환이 이뤄질 수 있고, XML로 구성된 웹 페이지에서 인쇄를 하고자 할 때 이 웹문서를 브라우저에서 변환을 시켜 PostScript로 인쇄를 하면 사용자가 원하는 최적의 인쇄물을 얻을 수 있을 것이라고 확신한다.

#### 참고 문헌

- [1] Ed Taft/ Jeff Walden by, PostScript Language Reference Manual, Addison Wesley, 1997
- [2] Dr.MichaelB.Spring &DavidDubinby, Hands-On PostScript, HAYDEN, 1992
- [3] KurtCale by, PROFESSIONAL XSL, wrox, 2001
- [4] <http://www.w3.org/TR/xsl/>
- [5] Rae A. Enranshaw by, Workstations and Publication Systems, 1987
- [6] 김차중, 동적포매팅 방식을 이용한 X윈도우/Motif상의 전자출판 시스템에 관한 연구, 1990.12
- [7] John Seybold/ Fritz Dressler by, Publishing From The DESKTOP, A Bantam Book, 1987
- [8] J.C.van by, Text Porcessing and Document Manipulation, CAMBRIDGE UNIVERSITY, 1986