Journal of KIIT. Vol. 17, No. 12, pp. 155-163, Dec. 31, 2019. pISSN 1598-8619, eISSN 2093-7571 155
http://dx.doi.org/10.14801/jkiit.2019.17.12.155

표와 텍스트기반 흔글 파일(.hwp)의 HTML로의 변환 소프트웨어 구현

오은경*, 허성우**, 정현우***[1], 전영수***[2]

Software Implementation to Covert Table and Text-Based Hangul Files(.hwp) to HTML

Eun-Kyung Oh*, Sung-Woo Hur**, Hyeon-Woo Jung***[1], and Youngsu Chon***[2]

요 약

본 연구에서는 흔글파일의 보다 효율적인 활용성을 위해 E-mail 에 첨부된 표와 텍스트 기반의 흔글 파일을 웹브라우저에서 바로 확인할 수 있도록 흔글 파일을 HTML파일로 변환하는 소프트웨어를 구현하였다. 관공서나 공공기관에서는 내부 결재 및 공문을 흔글로 작성하고, 작성된 문서를 게시판이나 E-mail을 통해 공유하고 있다. 이런 문서는 대부분 표와 텍스트로 구성되어 있고 대부분 편집이 아닌 열람을 목적으로 웹에서 공유되고 있다. 이러한 사용목적에 효율적인 변환 소프트웨어를 구현하기 위해 기존에 존재하는 변환 소프트웨어의 실행적 번거로움과 변환 오버헤드를 줄이는 방법을 제시한다. 표와 텍스트의 특성을 이용하여 서식파일의 패턴을 사용하고, 변환 후 구조를 단순화 하도록 구현하였다. 본 논문에서 제시한 방법으로 변환된 문서와 기존의 변환기에서 생성된 HTML 문서와 비교했을 때, 태그 속성의 사용이 대폭 감소되었고, 변화 후 2개의 파일만 생성되어 변환 결과가 단순화되었음을 보여주었다.

Abstract

In this study, we implemented a software that converts the table and text based Hangul files into HTML files so that the Hangul files attached to E-mail can be checked directly in a web browser. Public offices and public institutions write most of the official documents using Hangul, and share them through bulletin boards or e-mails. Most of these documents consist of tables and text, and most of them are shared on the web for viewing, not for editing. In order to implement effective conversion software suitable for this purpose, we present a method to reduce the execution overhead and conversion overhead of existing conversion software. By using the characteristics of table and text, the pattern of the template is used, and the structure after the conversion is simplified. When comparing the document converted by the method presented in this paper with the HTML document generated by the existing converter, the use of the tag attribute is greatly reduced, and the conversion result is simplified because only two files are generated after the conversion.

Keywords
HTML, conversion software, Hangul software, web browser, e-mail

* 동아대학교 컴퓨터공학과 강사
- ORCID: http://orcid.org/0000-0002-8539-3514
** 동아대학교 컴퓨터공학과 교수(교신저자)
- ORCID: http://orcid.org/0000-0002-9701-4008
*** 동아대학교 컴퓨터공학과 학부생
- ORCID[1]: http://orcid.org/0000-0002-8365-0505
- ORCID[2]: http://orcid.org/0000-0001-7513-3291

· Received: Nov. 21, 2019, Revised: Dec. 05, 2019, Accepted: Dec. 08, 2019
· Corresponding Author: Sung-Woo Hur
Dept. of Computer Engineering, Donga University
University, Donga-Univ, Hadan 2-doong, Saha-gu, Busan, Republic of Korea
Tel.: +82-51-200-7776, Email: swhur@dau.ac.kr

Ⅰ. 서 론

워드프로세서 흔글은 (주)한글과 컴퓨터에 의해 1989년 처음 개발된 이래로, 현재 한컴 오피스 2018 버전이 출시되고 사용되고 있다. 2019년 '2019 대한민국 브랜드스타'에서 소프트웨어 브랜드 가치 1위에 선정되었고, 2019년 현재 국내 시장 점유율 30%에 달하고 있다[1]. 대부분의 공공기관의 공문 및 결재 파일이 한컴 오피스의 한글 프로그램으로 제작되어지고 있다. 업무의 형태를 보면 결재 파일을 관련부서의 관계자들에게 E-mail의 첨부파일 형태로 전송하고 있는 추세다. 결재 파일이 첨부된 E-mail 수신자들은 대부분의 파일을 편집하는 용도로 파일을 사용하지 않고 열람하는 용도로 사용한다. E-mail에 MS-OFFICE 워드 프로세서인 MS-Word에서 작성한 문서가 첨부되어 있을 때의 열람은 웹 브라우저에서 바로 가능하다. 그러나 첨부된 흔글파일을 보기위해서는 한컴 오피스의 한글 프로그램을 실행 한 후 해당 문서를 확인할 수 있다. 이러한 번거로움을 벗어나기 위해 본 연구에서는 한글 파일을 HTML 파일로 변환하는 소프트웨어를 개발하여 웹상에서 흔글파일을 열람하려고 할 때 한컴 오피스의 프로그램이 실행되는 시간적, 시스템적 낭비를 막고 웹브라우저에서 바로 볼 수 있도록 하였다. 또한 한컴 오피스가 설치되어 있지 않은 컴퓨터에서 한글 뷰어를 설치하는 수고로움을 덜어준다.

흔글 파일을 TEX 파일로 변환하는 소프트웨어로 변환하는 연구가 2010년 있었다[2]. 이는 흔글 형식으로 작성된 논문파일을 TEX 형식의 파일로 변환하는 소프트웨어로서 흔글을 전문 조판 프로그래밍 언어로 변환하여 사용하는 것을 그 목적으로 하고 있다. 이 소프트웨어를 연구한 이유는 흔글 파일이 사용량이 많아지고 논문 투고 및 공공기관의 흔글 점유율이 높기에 흔글파일을 활용할 목적으로 개발한 것이다.

본 연구는 전문 언어로의 변환과 달리 웹 표준 언어인 HTML로 변환하여, 안드로이드 기반의 스마트폰 또는 일반 PC에서 웹브라우저에서 흔글 파일을 손쉽게 열어 볼 수 있도록 변환하여 흔글 파일의 대중적 사용성의 효율을 높이려고 한다.

본 연구에서는 공공기관에서 결재 및 공고문과 같이 형식 있는 문서에 초점을 맞추어 그림이나 기타 메모장과 같은 형식을 처리하지 않고 문자와 서식, 표 등으로 처리 형식을 국한하였다. 그 이유는 앞서 조사한 바로 흔글 파일을 웹상에서 보는 문서들의 종류는 결재 문서, 공고 문서들로 주로 문자와 표로 구성되어 있기 때문이다. 따라서 열람하고자 하는 흔글 파일에 효율적인 즉 공문이나 결재문에 적합한 HTML 파일 변환기를 구현하기 위해 그 범위를 제한하였다.

개발 언어는 JAVA를 이용하였고, 개발 환경은 Windows 10이다.

본 논문의 구성은 다음과 같다. 2장은 관련 연구로 흔글 문서의 형식 및 흔글 파일을 TEX로 변경한 내용에 대해 기술하고, HTML 변환기술 및 상업용 프로그램에 대해 간략히 설명한다. 3장에서는 본 연구에서 개발한 변환기의 설계 및 구현에 대해 자세히 설명한다. 4장에서는 본 연구에서 개발한 변환기를 이용하여 변환한 변환기의 처리 속도와 변환 결과 생성되는 문서의 태그를 분석한다. 5장에서는 본 연구의 결론 및 향후 과제에 대해 기술한다.

Ⅱ. 관련 연구

2.1 흔글 파일의 포맷 형식

흔글 파일의 파일 포맷은 한컴오피스 홈페이지[3]의 흔글 문서 파일 구조 5.0[4]을 참고하여 분석하였다. 해당 문서는 흔글 문서 파일 형식 5.0의 주요한 자료 형식 및 파일 구조, 레코드 구조에 대해에 대해 기술되어 있다. (주)한글과 컴퓨터에서는 문서 형식의 개방성과 표준화라는 가치 설정을 기반으로 흔글 97의 문서형식을 무상으로 지원하고, 흔글 2002~2010 문서의 XML 형식은 HWPML(Hangul Word Processor Markup Language)에 대해서도 문서를 공개하고 있다. HWPML 형식파일은 태그 포맷으로 되어 있는 마크업 언어로 작성되며, 텍스트 파일로 저장되어 있다. HWPML 스펙은 OWP-ML이란 이름으로 한국산업표준(KS X 6101:2 011)으로 제정되었다. 또한 한컴오피스에서 기록물 장기 보존 표

준 포맷인 PDF/A-1의 지원과 ISO 국제 문서 형식인 ODF와 OOXML 파일 형식의 불러오기와 저장하기를 적극 지원하고 있다.

한글 파일 문서 파일에 저장되는 내용은, 실제 사용자가 입력한 문서의 내용과 문자 장식 정보 뿐 아니라 문서를 편집할 당시 글꼴에 대한 정보, 조판에 영향을 주는 설정 사항(용지 종류, 여백정보)등도 포함한다.

한글 파일 문서는 파일의 크기의 최소화를 위해 zlib.org의 zlib을 사용한다. zlib은 웹상에 소스가 공개되어 있는 소프트웨어이다. 한글문서 파일 5.0의 구조는 윈도우즈의 복합 파일에 기초를 두며, 문자코드 ISO-10646표준을 기반으로 한다. 대분분의 문자정보는 유니코드(UTF-16LE)형식으로 전달되고 저장된다[4].

2.2 파일변환 관련 연구

한글을 다른 파일로 변환하는 연구로는 2009년 TEX로 변환하는 연구가 있었다[2]. 변환 목적은 한글 파일이 국내의 학술지 논문 발행 기관인 정보처리학회, 정보과학회, 정보기술과학회 등지의 출판용으로 사용되고 있고 논문 등에 사용되는 한글의 수식과 표 입력 등을 포함한 조판의 편의성과 TEX의 고품질성을 편리하게 사용하기 위함이었다. 해당 변환 소프트웨어는 활용분야가 일반적이 아니라 특정 분야에 제한되어 있다는 한계점이 있다. 해당 변환 소프트웨어는 수식, 표의 변환에 중점을 두었다. 해당 구현에 사용된 언어는 마이크로 소프트 C++이며, 한글 버전 2007, TEX는 KC2008이다.

두 번째 관련연구는 한글을 HTML로 변환하는 상용 SW이다. 한글 파일을 HTML로 변환하여 보여주는 상업용 변환 사**소프트사의 프로그램이 있다[5]. 해당 업체에서는 한글 파일 내에서의 여러 가지 개체(그림, 메모장), 페이지 구별까지 처리한다. 해당 상업용 SW를 통계청에 홈페이지에 적용하여 통계청 사이트에서는 한글 파일을 한컴 오피스를 실행 시키지 않아도 웹브라우저에서 바로 볼 수 있도록 제공하고 있다. 해당 프로그램은 상용 프로그램으로 변환 완성도나 변환범위(페이지 정보, 그림 정보, 여러 가지 개체 정보)가 넓다.

세 번째 관련 연구는 HTML 변환 연구로 스냅샷 기반 연산 오프로딩이다[6]. 스냅샷 기반 연산 오프로딩은 웹 어플리케이션의 상태를 HTML, CSS, JavaScript로 이루어진 또 다른 웹 어플리케이션의 형태로 저장하는 스냅샷 기술을 이용한다. 이 때 웹 어플리케이션은 HTML 문서를 파싱해서 DOM tree를 만들고 이를 이용하여 화면에 표시하며 현재 화면의 상태 정보를 가지게 하는 변환기술이다[7].

HTML문서 내에서도 정확한 본문 영역만을 추출하여 문서를 체계적으로 관리하기 위한 본문 추출에 대한 알고리즘에 대한 연구도 있었다[8]-[10]. 이처럼 필요한 부분만을 변환하는 연구는 문서의 활용에 필요하기 때문에 꾸준히 연구가 진행되고 있다.

본 연구에서는 공문서의 90% 이상을 차지하는 문자와 표만 변환하도록 그 범위를 제한하여 변환 속도의 증가 및 변환구조의 간략화를 그 목적으로 한다. 그리고 통계청과 같은 특정 사이트가 아닌 개인 PC에서 E-mail을 통해 첨부 받은 파일을 바로 웹브라우저를 통해 확인할 수 있도록 변환하는데 중점을 두어 사용 프로그램보다 일반화된 환경을 지원하는데 목적을 둔다.

III. 한글 변환 프로그램의 구현

3.1 한글 변환 프로그램 구현

JAVA를 이용하여 한글 파일을 html로 변환하기 위해서는 한글 hwplib 라이브러리 API를 사용하였다. 이는 zlib.org로 압축된 한글 파일의 압축을 해제하고 한글 파일을 JAVA를 이용하여 읽고 HTML로 변환하기 위해서 제공된 라이브러리를 사용하였다[11].

HTML의 여러 버전 중 HTML4 버전의 특징은 요소만을 사용하고 class, id 속성을 적극적으로 활용하여 구조 설계를 맥락에 맞게 직관적으로 단순화시킨 것이다. 웹의 구조와 스타일링을 HTML과 CSS 형태로 이원화하였다[12]. 본 연구는 이러한 이원화 형태를 이용하여 구현하였다.

한글 변환 프로그램구현을 위한 개략적인 흐름은 그림 1과 같다.

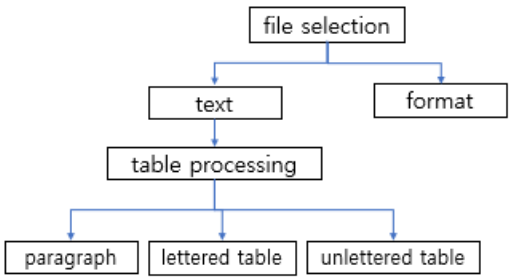
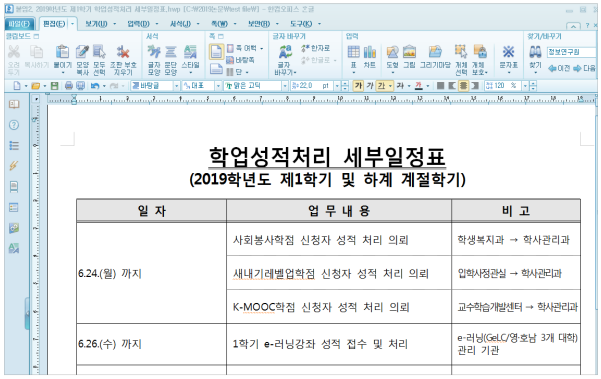


그림 1. 변환 처리 방향
Fig. 1. Conversion processing direction

본 연구에서 구현한 변환기는 2개의 HTML 파일을 생성한다. 하나는 서식관리를 위한 CSS 파일이고 나머지 하나는 문서 내용 표현을 위한 text HTML파일이다.

서식 정보를 CSS에 따로 보관하는 이유는 변환 과정에서 반복되는 서식이 HTML코드에서 중복되어 나타내어지는 부분을 간략화 시키기 위함이다. 변환 범위를 문자와 표로 제한 시켰기 때문에 이로 인해 사용되는 서식도 일정한 패턴의 반복이 주를 이룬다. 따라서 이를 효율적으로 처리하기 위해 반복되는 서식을 HTML 변환파일에 반복적으로 삽입하는 방법 대신 서식의 패턴을 CSS 파일에 일정한 형식으로 지정하여 저장하고 패턴을 삽입하는 방법을 사용한다. 이 방법을 사용하는 목적은 반복되는 속성을 줄여 파일의 크기 및 변환 속도를 향상시키기 위함이다.

문서를 나타내는 text HTML 파일은 텍스트를 크게 3가지로 분류하여 처리한다. 순수 문자들로만 이루어진 문단, 글자 처리된 표, 글자 처리되지 않은 표로 분류한다. 본 변환기는 그림 개체와 같은 개체를 처리하지 않는다. 이유는 대부분의 공문들은 텍스트와 표로만 구성되어 있기 때문에 변환 시간 및 변환 구현의 효율성을 위해 텍스트와 표의 변환으로 국한하였다. 각각의 변환 방법은 다음 절에서 차례대로 상세히 설명한다. 사용된 문서는 저자가 소속된 대학교에서 공문으로 첨부 받은 '2019학년도 제 1학기 학업성적처리 세부일정표.hwp' 문서를 변환한 파일이다. 그림 2는 한컴 오피스로 열어본 문서이고, 그림 3은 본 연구에서 구현한 변환기를 이용하여 HTML로 변환 후 웹브라우저로 열어본 문서이다.

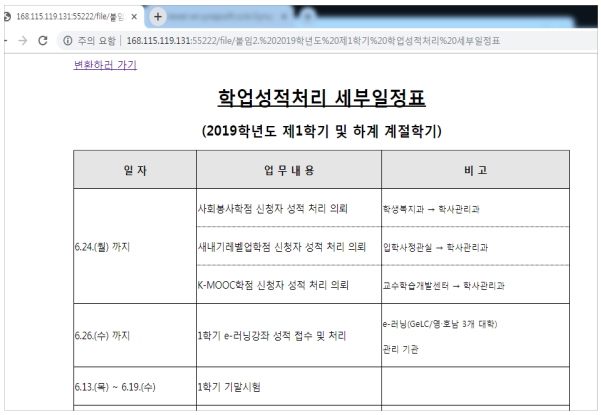


학업성적처리 세부일정표

(2019학년도 제1학기 및 하계 계절학기)

일 자	업 무 내 용	비 고
6.24.(월) 까지	사회봉사학점 신청자 성적 처리 의뢰	학생복지과 → 학사관리과
	새내기레벨업학점 신청자 성적 처리 의뢰	입학사정관실 → 학사관리과
	K-MOOC학점 신청자 성적 처리 의뢰	교수학습개발센터 → 학사관리과
6.26.(수) 까지	1학기 e-러닝강좌 성적 접수 및 처리	e-러닝(GeLC/영호남 3개 대학) 관리 기관

그림 2. 한글로 열어본 문서
Fig. 2. Document opened using Hangul



변환하러 가기

학업성적처리 세부일정표

(2019학년도 제1학기 및 하계 계절학기)

일 자	업 무 내 용	비 고
6.24.(월) 까지	사회봉사학점 신청자 성적 처리 의뢰	학생복지과 → 학사관리과
	새내기레벨업학점 신청자 성적 처리 의뢰	입학사정관실 → 학사관리과
	K-MOOC학점 신청자 성적 처리 의뢰	교수학습개발센터 → 학사관리과
6.26.(수) 까지	1학기 e-러닝강좌 성적 접수 및 처리	e-러닝(GeLC/영호남 3개 대학) 관리 기관
6.13.(목) ~ 6.19.(수)	1학기 기말시험	

그림 3. 변환 후 크롬으로 열어본 문서
Fig. 3. Document opened Chrome after conversion

그림 2와 그림 3을 비교해 보면 텍스트의 구조가 일치되고 굵기, 맞춤 등의 서식의 일치됨을 확인할 수 있다. 다음 절에서는 서식 파일의 처리에 대해서 설명한다.

3.2 서식 변환을 위한 처리

본 변환기에서는 텍스트와 표로 특정되어진 부분만 변환하기 때문에 서식 등이 반복되어진 형태로 처리되고 표시됨을 확인할 수 있었다. 기본적으로 서식은 표 서식, 글꼴 서식, 페이지 서식으로 처리한다. 표 서식은 위치정보를 보관하고 있는 td, 테두리 정보를 가지고 있는 borderfill, 글꼴 서식 정보를 가지고 있는 charshape, 문단 서식 정보를 가지는 parashape등의 명칭으로 구별하여 처리하였다. 테두리종류는 기본 값을 18개, 테두리 두께는 16개로 정의 해놓고, 이 서식과 다른 서식이 변환 시 발견되면 서식을 추가 등록한다.

```
.borderfill29{
border-bottom: 0.1mm solid #000000;
border-left: 0.12mm solid #000000
;border-right: 0.12mm none #000000
;border-top: 0.12mm solid #000000;
}
```

그림 4. borderfill 서식 구성 요소
Fig. 4. Borderfill formatting component

```
.charshape17{
font-size: 14.0pt;
color: #000000;
font-family: Gulim;
font-weight:bold;
}
```

그림 5. charashape 서식 구성요소
Fig. 5. Charashape formatting component

borderfill에 정의 되는 서식은 그림 4와 같이 구성된다. 테두리에 대해서는 4가지 위치에 대해 굵기, 선 종류, 테두리 색에 대한 정보를 각각 지정하여 패턴을 저장한다. 그림 4는 테두리에 대한 서식으로 패턴이름은 borderfill29이다. 테두리 서식정보로 셀에서 아래쪽 테두리 서식 정보 식별자인 border-bottom, 두께 0.1mm, 테두리 모양 solid, 테두리 색 검정색을 16진수로 저장하고 있다.

글꼴 서식은 다양한 글자체로 인해 기본 서식 6개를 지정하고 그 외는 변환되어질 때 조금이라도 앞에 등록된 서식과 다르면 변환 시 서식 스타일을 추가 등록하도록 처리하였다. 서식 스타일의 구성요소는 그림 5와 같다. 글꼴 서식은 글꼴 크기, 글꼴색, 글꼴 스타일, 글꼴 효과(굵게, 이탤릭체, 밑줄 등)에 대한 정보를 하나의 패턴으로 저장한다. 그림 5의 서식의 패턴 이름은 charashape17이고, 글자 크기 14.0pt, 글자색 검정색, 글꼴은 굴림체, 굵게 처리된 서식을 나타내고 있다.

문단 서식은 표 또는 단순 텍스트로 구성된 텍스트 내에서 단락 서식과 관련된 부분을 처리한다. 문단 서식은 기본 값 없이 변환 시 발견되는 서식을 등록하여 처리한다. 그림 6은 parashape의 구성요소를 나타낸다. 문단 서식은 텍스트 맞춤 등의 정보로 패턴을 저장한다. 그림 6의 parashape12는 문단의 맞춤이 가운데 맞춤을 나타낸다. parashape13은 문단의 양쪽 맞춤의 서식을 저장하고 있다.

그림 7은 상용프로그램에서 변환한 문서의 구조를 크롬의 개발자 모드를 통해 확인한 저장 구조이다.

```
.parashape12{
text-align:Center;
}
.parashape13{
text-align:Distribute;
}
.parashape14{
text-align:Justify;
}
```

그림 6. parashape 서식 구성요소
Fig. 6. Parashape formatting component

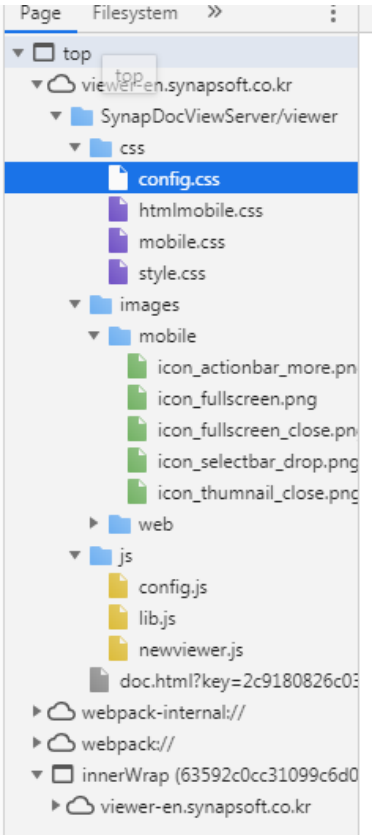


그림 7. 상용 변환기로 변환한 문서구조
Fig. 7. Document structure converted by commercial converter

그림 7에서 보면 상용 변환기는 모든 개체, 페이지 테두리, 페이지 경계선을 위한 여러 가지 GUI 그리고 화면을 보기위한 페이지 내용까지 처리하기 때문에 변환 후 문서의 구조가 복잡하다. 이에 비해 본 연구에서 변환기를 이용하여 변환한 그림 8에서 나타난 문서의 구조는 2개의 파일 CSS파일과 HTML파일 2가지로 간단함을 확인할 수 있다. 단순하지만 비교한 상품과 비교했을 때 웹브라우저로 보여지는 문서는 동일한 결과를 보인다.

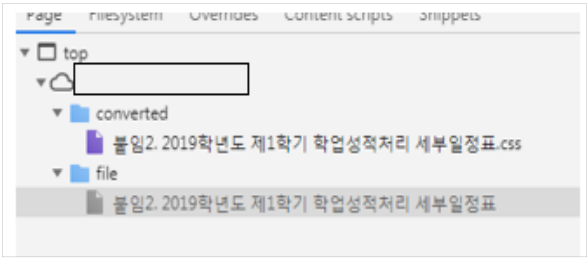


그림 8. 본 변환기로 변환한 문서 구조
Fig. 8. Document structure converted by commercial converter

3.3 텍스트 변환을 위한 처리

본 변환기에서의 텍스트는 크게 3가지로 나눈다. 문자로만 구성되어 있는 문단, 글자 처리된 표, 글자 처리되지 않은 표로 나눈다. 본 변환기는 단순성을 목적으로 텍스트와 표에 대한 변환기를 구현하였기 때문에 그림이나 주석 등과 같은 개체는 처리 범위에 포함시키지 않았다. 글자 처리되지 않은 표는 개체이기 때문에 이러한 표에 대해 따로 처리해야 할 필요성이 있다. 특히 글자 처리되지 않은 표는 정확한 위치 좌표가 지정되지 않은 문제점이 있다. 따라서 저장했을 때의 위치가 글자처럼 특정위치에 지정되지 않기 때문에 작성한 컴퓨터와 다른 컴퓨터, 환경이 다른 모니터에서 열람되었을 때 작성자가 의도한 위치에 보존되지 않고 다른 위치에 배치되는 문제점이 발생한다. 이러한 문제점을 해결하기 위해 본 연구에서는 페이지의 크기와 문단, 글자 처리된 표의 위치와 글자 처리되지 않은 표의 위치의 충돌 여부를 체크한다. 글자 처리되지 않은 표의 위치와 크기를 이용하여 다른 요소들과 충돌이 생기지 않는 위치로 글자 처리되지 않은 표를 배치시키고 위치를 지정하여 HTML로 변환할 때 지정한 위치를 사용하도록 설정하였다. 전체적인 텍스트 변환을 위한 처리 설계는 그림 9로 나타내었다.

그림 9의 설계도는 한글의 실제 문자 값에 대한 정보를 저장하고 처리하는 PageRecordChar와 문단, 표(글자 처리된 표), 사각형(글자처리 안된 표)를 구별하는 열거형 클래스인 PageRecordType로 PageRecord를 구성한다.

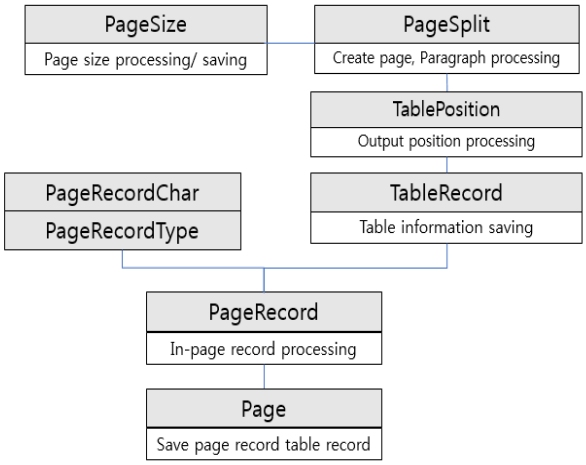


그림 9. 텍스트 처리 설계도
Fig. 9. Text processing architecture

글자 처리된 표는 문단과 동일하게 처리하기 때문에 이 루틴을 통해 처리되고, 글자 처리되지 않은 표는 페이지 크기를 처리하는 PageSize의 페이지를 나누는 기준에 대한 정보를 이용하여 PageSplit에서 페이지 경계 및 다른 문자들 사이의 충돌 여부를 확인 한 후 글자 처리되지 않은 표를 찾아낸다. 이 정보로 구성된 표를 매개 변수로 전달받은 TablePosition에서 글자처리 되지 않은 표에 대한 부분을 처리한다. 이 때 글자 처리 되지 않았기 때문에 글자들과 표가 충돌 가능성이 있어 문단들과 충돌이 발생하는지의 여부를 이용하여 충돌이 발생하지 않는 위치로 표의 위치를 설정한다.

처리된 정보들은 종류에 따라 문단과 글자 처리된 표의 정보를 PageRecord에 바로 저장하고, 글자 처리되지 않은 표는 TableRecord에서 정보를 받아 PageRecord에 저장한다. PageRecord는 Page를 구성하는 기본 단위가 된다. 한글에서는 페이지 단위로 페이지에 저장된 모든 글자, 문단, 표, 서식들에 대한 정보를 저장한다. 본 변환기의 텍스트는 Page에 저장된 PageRecord를 이용하여 HTML 문서를 생성하여 text HTML 문서를 완성한다.

Ⅳ. 구현 결과 분석

4.1 실험 파일 및 변환 분석

실험에 사용한 파일은 저자가 소속된 학교에서 공문으로 첨부 받은 한글 중의 하나를 사용하였다.

파일의 크기는 16KB이고 1개의 표와 텍스트로 구성되어 있다.

문서를 HTML로 변환 한 후 시중에 판매중인 변환 프로그램과 HTML 소스를 비교하여 보았다.

서식 속성과 관련된 HTML 코드를 비교해보면, 글자를 표현하기 위한 <p>태그와 <p>태그에 서식을 지정하기 위해 사용되는 태그의 속성을 비교해보면 표 1과 같다.

표 1. 문자와 서식을 위한 태그 비교(단위 : 개수)
Table 1. Comparing tags for characters and formatting

	this converter	commercial SW
<p> tag	57	60
tag	66	166

표 1에서 보듯 속성의 사용이 차이가 많이 나는 것을 확인 할 수 있다. 이는 서식을 CSS 파일에 저장해놓고 CSS 파일에서 불러와서 처리함으로 인해 태그의 사용이 대폭 감소되었다. 따라서 HTML의 문서가 간단히 처리됨으로 처리속도와 변환 속도 향상의 요인이 된다.

두 번째로 실제적인 서식의 속성의 사용에 대해 비교해 보도록 한다.

본 변환기에서 처리한 방식으로 변환한 HTML 문서의 소스코드에서 해당 서식의 사용 빈도수를 표 2로 나타내었고, 흔글 xhtml 변환 상용 소프트웨어 사**소프트 문서 뷰어에서 변환한 xhtml 코드에서 나타난 서식 관련 빈도수를 표 3에 나타내었다. 표 2는 각각 테두리 서식, 글꼴 서식, 문단 서식으로 분류하여 노출 빈도수를 나타내었다.

표 3을 보면 문서가 표로 구성되어 있기 때문에 테두리 속성이 많이 사용되었고, 이 속성들은 동일한 서식이 적용되고 있으나 각 셀에 서식을 나타내야 하기 때문에 같은 테두리 서식 속성의 사용이 많이 나타남을 확인할 수 있었다. 본 연구는 앞서 설명한대로 CSS 파일에 패턴으로 만들어 놓은 서식을 class 속성으로 지정하여 HTML에 저장하기 때문에 동일한 서식의 속성을 감소할 수 있다. 본연구로 변환한 문서에서 적용된 서식의 사용빈도수는 총 176회로 표 3의 총 빈도수 858회와 비교했을 때 서식 속성의 사용이 대폭 감소되었다.

표 2. 제시한 변환기로 변환한 서식 속성 사용 빈도수
Table 2. Frequency of formatting attributes converted by the proposed converter.

borderfill (border)		charshape (font format)		parashape (paragraph format)	
No.	count	No.	count	No.	count
0~2	each 0	0~5	0	0~11	each 0
3	11	6	12	12	1
4	10	7	8	13	3
5	1	8	1	14	14
6	0	9	11	15	22
7	10	10,11	each 1	16	4
8	0	12	2	17	1
9~29	each 1	13	1	18	9
		14	2	19	1
		15	3	20	2
		16	1		
		17	2		
		18	5		
		19	2		
		20,21	each 1		
		22	5		
		23	0		
		24	3		
		25~28	each 1		
sum	53		66		57
total	176				

표 3. 상용 SW로 변환한 서식 속성 사용 빈도수
Table 3. Frequency of formatting attributes converted by commercial SW

border		font		align	
attr.	count	attr.	count	attr.	count
top	70	family	166	vertical	52
right	30	size	159	line-height	59
bottom	52	weight	59		
left	36				
solid	175				
sum	363		384		111
total	858				

V. 결론 및 향후 과제

5.1 결 론

본 변환기 구현의 목적은 E-mail에서 첨부된 공문과 같이 열람을 위해 작성된 텍스트와 표 위주의

한글 파일을 HTML로 변환하여 웹브라우저에서 바로 열어볼 수 있도록 하는 것이다. 시중에 상용SW로 한글 파일을 변환하여 웹으로 확인할 수 있으나 비용이 비싸고, 또한 한글전체를 변환하기 때문에 변환기의 처리 오버헤드가 크다. 본 연구는 이런 점을 보완하기 위하여 공무원, 학교 등의 대부분의 기관에서 열람을 위해 업무상 첨부된 텍스트와 표위주의 한글 파일의 변환에 효율적인 변환기를 제작 구현하였다. 본 변환기를 이용하여 변환한 후 한글 원본 파일과 비교해 보았을 때 전체적인 모양은 유지되면서 웹상에서 빠른 속도로 변환된 문서를 확인할 수 있었고, 상용SW보다 변환 후 생성된 파일 구조도 간단하고, HTML 문서의 크기를 비교해 보면 한글 원본 파일의 크기가 16KB의 문서를 본 변환기로 변환하여 작성한 HTML 문서의 크기는 10KB이고 상용SW로 변환된 HTML 문서의 크기는 29KB이다. 본 연구에서 제시된 방법으로 변환되어 작성된 파일의 크기가 작음을 확인할 수 있었다. 표와 텍스트로만 작성된 한글 변환기로의 본 연구에서 제시된 변환기의 효율성을 입증할 수 있었다.

5.2 향후 과제

본 연구에서 제시한 변환기는 표와 텍스트 기반의 한글 변환기로 웹브라우저에서 한글의 보다 대중적인 활용을 높이기 위해 한글 파일을 HTML 로 효율적으로 변환하는 방법을 제시하였다. 공문으로 사용되는 문서 중에는 직인등과 같이 간단한 그림 파일들이 간혹 포함이 되는 것을 볼 수 있다. 본 연구에서는 한글 파일의 열람 목적으로 봤을 때 내용 전달에 있어서 이 부분의 비중을 두지 않았으나, 간단한 그림을 처리하는 방법을 추가한다면 좀더 안정적인 변환기가 될 것이라 기대할 수 있겠다.

References

[1] Excerpt from Hankyung newspaper. 25. Mar. 2019 https://www.hankyung.com/economy/article/2019032501531 [accessed: Oct. 8,2019]

[2] Sung-Won Kim, Hanna Lee, Sanghoon Park, and Chang-hyuk Oh, "Implementation of conversion software of Hangul file into TEX", Journal of Korean Data & Information Science society, Vol. 21, No. 1, pp. 99-107, 2010.

[3] https://www.hancom.com/main/main.do [accessed: Oct. 8,2019]

[4] https://www.hancom.com/etc/hwpDownload.do?gnb0=269&gnb1=271&gnb0=101&gnb1=140 Hangul Documents file structure 5.0, revision 1.3:20181108

[5] Synapsoft Corporation,Experience Document Viewer. http://support.synapsoft.co.kr/demo/demo_viewer_layer2.syn [accessed : Nov. 19, 2019]

[6] Hyuk-Jin Jeong and Soo-Moon, "Offloading of Web Application Computations: A Snapshot-Based Approach", Proc. of the IEEE 13th International Conference on Embedded and Ubiquious Computing, pp. 90-97, 2015(in Portugal)

[7] JinSeok Oh, Jin-woo Kwon, Hyunkwoo Park, and Soo-Mook Moon, "Migration of Web Applications with Seamless Execution," Proc. of the 11th ACM AIGPLAN/SIGOPS International Conference on Virtual Execution Environments, pp. 173-185, 2015.(in Turkey)

[8] Hyun-Gee Jeon and Chan KOH, "Text Extraction Algorithm using the HTML Logical Structure Analysis", Journal of Digital Contents Society, Vol. 16, No. 3, pp. 445-455, Jun. 2015

[9] Cristian K., Peter F. Wolfgang N., "Boilerlate Detection using Shallow Text Features," The third ACM international conference on Web search and data minig, pp. 441-450, 2010.

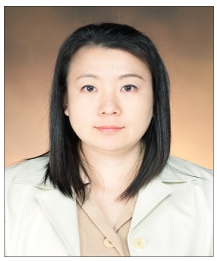
[10] Deng C., Shipeng Y., Ji-Rong W., Wei-Ying M., "VIPS: a vision-based Page Segmentation Algorithm", Microsoft Technical Report(MSR-TR-2003-79), 2003

[11] https://github.com/noelod0/hwplib [accessed: Nov. 2, 2019]

[12] http://www.w3.org/standards/webarch/ [accessed: Oct. 3, 2019]

저자소개

오 은 경 (Eun-Kyung Oh)

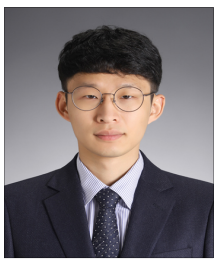


1997년 2월 : 동아대학교 컴퓨터공학과(공학사)
2003년 2월 : 동아대학교 컴퓨터공학과(공학석사)
2010년 2월 : 동아대학교 컴퓨터공학과(공학박사)
2000년 ~ 현재 : 동아대학교 컴퓨터공학과 강사
관심분야 : Cad Algorithm, 전산 수학, VLSI 알고리즘, 최적화 알고리즘, 모바일 프로그래밍

허 성 우 (Sung-Woo Hur)

1981년 2월 : 경북대학교 전자공학과(공학사)
1983년 2월 : 한국과학기술원 전산학과(공학석사)
2000년 6월 : Univ. Illinois (Chicago) EECS (Ph.D.)
2002년 2월 ~ 2006년 8월 : 미국 인텔 연구소 (산호세) 기술자문위원
1986년 3월 ~ 현재 : 동아대학교 컴퓨터공학과 교수
관심분야 : Cad Algorithm, Combinatorial Optimization

정 현 우 (Hyeon-Woo Jung)



2013년 3월 ~ 현재 : 동아대학교 컴퓨터공학과 재학
관심분야 : 알고리즘, 운영체제, 웹 프로그래밍

전 영 수 (Young-Su Chon)

2013년 3월 ~ 현재 : 동아대학교 컴퓨터공학과 재학
관심분야 : 인공지능, 알고리즘, 웹 프로그래밍