

---

# 멀티레이어 기법을 활용한 웹기반 POD 시스템

이병권\* · 김학진\* · 정회경\*\*

Web-based POD system Using Multi-Layer Technique

Byung-Kwon Lee\* · Hak-Jin Kim\* · Hoe-Kyung Jung\*\*

## 요 약

본 논문에서는 멀티레이어(Multi-layer) 기법을 이용한 A4 크기의 웹기반 POD(Print on Demand) 시스템을 설계 및 구현하였다. 이를 위해, 멀티레이어 기법을 활용하고, 접근성이 뛰어난 웹 서비스라는 환경을 이용하며, 템플릿을 생성·제공하고, 사용자가 쉽게 편집 할 수 있도록 제공하며, 편집한 파일과 동질의 인쇄물을 실시간으로 출력할 수 있는 웹 표준을 따르는 POD 시스템의 프레임워크(framework)를 제안한다. 이는 웹기반 전자출판 및 e-book 분야 등에서 널리 활용될 수 있을 것이다.

## ABSTRACT

In this paper, we design and implement A4 sized, web-based POP(Print On Demand) system using multi-layer technique. We Propose the framework of POP system, which uses multi-layer technique, uses highly accessible web-service, produces template, enables a user to easily edit, and allows the user to instantly print edited files. This system will be widely used in web-based electronic publication and e-book.

## 키워드

POD, 멀티레이어, e-book, 전자출판

## Key word

POD, Multi-layer, e-book, Electronic publication

---

\* 준회원 : 배재대학교  
\*\* 종신회원 : 배재대학교 (교신저자, hjkung@pcu.ac.kr)

접수일자 : 2012. 06. 01  
심사완료일자 : 2012. 06. 01

## I. 서 론

인터넷의 보급에 따라 인쇄 분야에도 적용되어, 최근에는 인쇄물을 사용자가 직접 제작하여 실시간으로 인쇄할 수 있는 POD(Print on Demand) 시스템을 갖춘 웹사이트들이 생겨나고 있다[1-3].

이에 인쇄 기획자들로 하여금, 포토샵이나 일러스트레이터와 비교하여 상대적으로 쉽고, 접근이 쉬운 인쇄 분야에 활용할 수 있는 명함 제작, 스티커 제작과 같은 활용 가능한 프로그램을 만들고, 접근성을 높이기 위한 프로그램의 보급이 시급하다. 또한, A4 크기의 출력물에 대해 웹상에서 출력되기를 바란다[4-7].

본 논문에서는 기존연구에서의 명함 크기 출력물에 대해 A4 크기로 확장가능 하도록 멀티레이어(Multi-layer) 기능을 이용하여 웹상에서 보여지는 모습을 고품질로 스크린샷 기술을 적용하여 이미지를 생성하고 썸네일 형태로 관리하며, 실시간으로 인쇄소에서 사용할 수 있는 형태의 POD[1] 시스템으로, 접근성과 사용의 편리성, 이미지의 고품질을 목표로 이들 기능을 수행할 수 있는 프레임워크를 제안한다.

사용자가 원하는 이미지를 위치시키기 위해서는 콘텐츠의 유동성이 필요하다. 그러나 정적인 웹 브라우저의 특성상 웹에 표현되는 모든 콘텐츠는 위치기반에 있어 정적인 구조를 갖는다. 이를 원하는 대로 표현하기 위해서는 레이어의 개념이 필요하며 여러 개의 콘텐츠를 활용하기 위해 구성 요소를 모두 레이어로 만들 필요성이 대두된다. 따라서, 멀티레이어를 이용함으로써 POD 시스템 기반에서 사용자가 원하는 위치에 이미지를 이동시킬 수 있으며 원하는 디자인을 갖출 수 있다. 이를 위해 다중의 파일 업로드 폼을 구현하였으며, 이를 서버에 저장하고, 다시 호출하여 화면에 배치시켰다. 이들의 모든 상황은 위지윅(WYSIWYG : What You See Is What You Get) 모듈에 의해 실시간 반영되며, 이를 위해 PHP의 업로드 컴포넌트와 DOM 객체가 필요하고, 레이어마다 위치 정보를 체크하기 위해 자바스크립트의 마우스 이벤트가 필요하여 이에 대한 연구를 수행하였다.

## II. 관련 연구

본 장에서는 A4 크기 출력물을 처리하기 위해 요구되는 리사이징(resizing) 및 멀티레이어를 이동하기 위한 알고리즘에 대해 설명한다.

### 2.1. 이미지 크기 리사이징

인쇄용 출력의 해상도는 일반적으로 300 DPI(Dot Per Inch)의 해상도를 가진다. A4 규격의 인쇄용 출력을 위해서는 300 DPI로 설정된 이미지의 화면 해상도는 2,520 픽셀 x 3480 픽셀의 규격을 가지게 되는데, 이 규격은 모니터에서 한 화면에 표현 할 수 없는 크기이다 [8-9]. 또한, 이 크기는 기존의 스크린샷 기능으로 캡처가 불가능해 POD 시스템을 이용하지 못한다. 따라서, 기본이 될 배경 이미지와 업로드된 이미지를 해상도에 맞게 리사이징하는 연산이 필요하다. 원본 배경 이미지를 화면에 맞게 일정 비율로 축소하고, 업로드된 이미지 역시 이 비율로 축소하여 보여줌으로써 이용자는 실제 출력물에 가까운 디자인을 볼 수 있다. A4 크기의 편집 시 리사이징을 거치지 않을 경우 다음의 2가지 문제가 발생한다. 첫째는 이미지 깨짐이다. 최종 결과물은 PDF로 출력하지만 PDF 출력전이나 편집모드에서 화면상에서는 비율 축소로 인한 리사이징으로 실제로 잘 보였던 부분이 실제 출력에서는 심각한 픽셀화로 인해 실제 효용가치를 잃어버리는 상황을 개발 시에 실험에 의해 검증되었다. 따라서, 실제로 출력하게 될 상황에 맞게 업로드된 이미지의 크기를 원본 축소 비율에 맞게 리사이징 하여 보여주는 것이다. 둘째로 정확한 위치 포인트이다. 원본 배경의 실제 크기가 축소 비율로 보이는 상태에서 이미지의 깨짐 현상을 무시하고라도 정확한 위치를 측정하여 실행한 합성 실험 결과 매우 부정확한 결과를 보인다. 따라서, 본 논문에서는 이미지 크기에 대한 이해를 바탕으로 다중 이미지를 업로드하고 리사이징하여 멀티레이어상에서 A4 크기의 편집이 가능하도록 하고 있다.

### 2.2. 자바스크립트를 이용한 레이어 이동

자바스크립트가 가지고 있는 마우스 이벤트를 활용하여 마우스 버튼 상태에 따라 각각 함수를 호출한다. 각 레이어를 이동 시키는 알고리즘은 그림 1과 같다. 이를 이용하여 멀티레이어들을 해당 위치로 이동하게 된다.

```

<script>
<!--
var bdown = false;
var x, y;
var sElem;
function mdown(){
    if(event.srcElement.className == "drag") {
        bdown = true;
        sElem = event.srcElement;
        x = event.clientX;
        y = event.clientY;
    }
}
function mup(){
    bdown = false;
}
function moveimg(){
    if(bdown) {
        var distX = event.clientX - x;
        var distY = event.clientY - y;
        sElem.style.pixelLeft += distX;
        sElem.style.pixelTop += distY;
        x = event.clientX;
        y = event.clientY;

        var fm = document.IsLandBody;

var newY = sElem.offsetTop +
parseInt(sElem.parentNode.style.top.replace('px',''));
var newX = sElem.offsetLeft +
parseInt(sElem.parentNode.style.left.replace('px',''));
        fm.Y.value = newY;
        fm.X.value = newX;
        return false;
    }
}
document.onmousedown = mdown;
document.onmouseup = mup;
document.onmousemove = moveimg;
//-->
</script>
</head>

```

그림 1. 위치 이동을 위한 자바스크립트  
Fig. 1 Javascript for Translocation

### III. A4 편집 모듈 처리 설계

본 장에서는 A4 용지 편집 처리를 위한 A4 편집 모듈 구조와 POD 시스템의 구조, 인쇄용 데이터 처리를 위한 RGB2CMYK 변환 모듈에 대해 설명한다.

#### 3.1. A4 용지 편집 모듈 구조

A4 용지 편집 처리를 위한 A4 편집 모듈 구조도는 그림 2와 같다. 그림에서 보는 바와 같이 편집 모듈에 의해

업로드된 이미지는 iframe으로 submit되어 서버에 업로드된다. 업로드 과정을 거치면서 1차적으로 업로드된 이미지의 정상 여부를 판단한다. 여기서 정상 여부란 웹에서 표현될 수 있는 이미지를 판단하는 부분이다. 실제로 웹에서는 가독성과 네트워크의 안전성을 위해 적은 용량의 이미지를 사용한다. 그러나, POD 시스템에서는 출력용 이미지이므로 용량보다는 이미지의 질에 집중하여야 하지만 웹 브라우저에서 보여줄 수 있는 이미지 포맷인지 여부는 체크하여 적용 여부를 판단하여야 한다.

일반적으로 웹에서 사용이 가능한 이미지 포맷은 JPG, GIF, PNG, BMP 정도이다. 이 이외의 포맷은 코드 베이스가 필요하거나 아예 지원이 되지 않으므로 체크하여야 한다. 이렇게 업로드시 이미지 가용여부를 점검한 후, 선택된 백그라운드 이미지의 축소 비율에 따라 업로드된 이미지의 리사이징 과정을 거쳐 다시 웹에 뿌려지게 된다. 시스템에서 사용자는 위지웍 형태의 편집과정을 거치게 되며, 업로드된 이미지의 크기 등을 확인할 수 있게 된다.

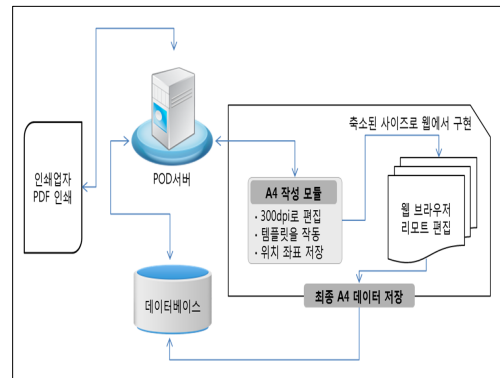


그림 2. A4 편집모듈 구조도  
Fig. 2 Block Diagram of A4 Editing Module

A4 편집을 위한 테이블 구조는 그림 3과 같다. sessnum은 수시로 저장되어 파일의 유일성을 위해 웹 브라우저의 세션을 저장하고 work\_code는 작업물의 고유 코드이다. id는 작업자의 아이디이며, 작업자의 히스토리를 위해 기록한다. files 필드에 업로드된 파일들을 기록하고, texts 필드에 기술된 텍스트를 html 형식으로 보관하며, lastfile 필드에 최종 완성된 파일명을 기록한다. memo를 통하여 별도의 기록사항 등을 메모한다.

```
CREATE TABLE IF NOT EXISTS 'pod_a4_table' (
'no' int(10) unsigned NOT NULL AUTO_INCREMENT,
'sessnum' carchar(50) unsigned NOT NULL,
'workcode' varchar(20) unsigned NOT NULL,
'files' text unsigned NOT NULL,
'texts' text unsigned NOT NULL,
'id' varchar(30) unsigned NOT NULL,
'lastfile' varchar(255) unsigned NOT NULL,
'wdate' varchar(10) unsigned NOT NULL
'ip' varchar(25) unsigned NOT NULL
'memo' varchar(255) unsigned NOT NULL
PRIMARY KEY ('no')
) ENGINE=MyISAM DEFAULT CHARSET=euckr
AUTO_INCREMENT=1 ;
```

그림 3. pod\_a4\_table 구조  
Fig. 3 Structure of pod\_a4\_table

3.2. POD 시스템의 구조

본 논문에서 제안한 웹기반 POD 시스템은 요청이 있을 때 출력할 수 있는 실시간 서비스를 제공하기 위한 목적을 가지므로 PHP기반의 웹 서비스를 통해 이루어졌다. 이를 위해 기본적인 웹의 설계가 필수적으로 이를 위한 기본 골격은 다음과 같다.

모든 웹 페이지는 팝업창을 제외하고, main화면(main.php)을 통해 이루어진다. 화면의 전체적인 구성은 그림 4와 같다.

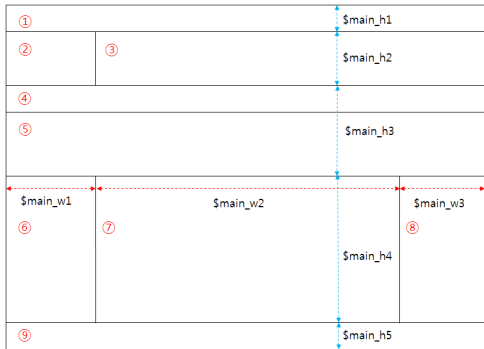


그림 4. 웹 페이지의 구조  
Fig. 4 Structure of Web Page

이 페이지를 통해 팝업이나, 블랙리스트 검사, IP검사 등을 효율적으로 수행할 수 있다. 변수 \$main\_h1은 셀의 픽셀(pixel)을 의미하며 해당하는 값이 "0"인 경우에는 해당하는 곳을 표시하지 않는다. 많은 페이지가

웹페이지에 존재하는데, 각각의 구분은 \$request값을 통해 구분할 수 있고, \$request가 없는 경우에는 "홈"을 눌렀을 때를 의미한다. 이때 \$main\_total\_w = \$main\_w1 + \$main\_w2 + \$main\_w3로 이 값은 페이지가 오른쪽으로 늘어나는 것을 방지하는데 사용하여야 한다. 내용이 포함될 ⑦에 긴 텍스트가 존재하는 경우 전체적으로 화면 늘어나는 상황이 발생할 수 있다.

3.3. RGB2CMYK 변환 모듈

인쇄용 데이터 처리를 위해서는 RGV(Red Green Blue) 색상을 CMYK(Cyan Magenta Yellow Black) 색상 변환이 필요하다[10]. 다음에 RGB 색상을 CMYK 색상으로 변환하는 알고리즘을 보인다.

```
imgname = "imagecolor.jpg";
$image = @imagecreatefromjpeg($imgname);

$imagehw = GetImageSize($imgname);
$imagewidth = $imagehw[0];
$imageheight = $imagehw[1];

print("Width : $imagewidth : Height : $imageheight ");

for ($row = 0 ; $row < $imageheight ; $row++)
{
    for ($col = 0 ; $col < $imagewidth ; $col++)
    {
        $rgb = ImageColorAt($image, $col, $row);
        $r = ($rgb >> 16) & 0xFF;
        $g = ($rgb >> 8) & 0xFF;
        $b = $rgb & 0xFF;

        /* RGB -> CMY */
        $C = 1 - ( $r / 255 );
        $M = 1 - ( $g / 255 );
        $Y = 1 - ( $b / 255 );

        /* CMY -> CMYK */
        $var_K = 1;

        if ( $C < $var_K ) $var_K = $C;
        if ( $M < $var_K ) $var_K = $M;
        if ( $Y < $var_K ) $var_K = $Y;

        $C = ( $C - $var_K ) / ( 1 - $var_K );
        $M = ( $M - $var_K ) / ( 1 - $var_K );
        $Y = ( $Y - $var_K ) / ( 1 - $var_K );
        $K = $var_K;

        print("$row : $col -- $C : $M : $K :$K");
    }
}
```

#### IV. POD 시스템 구현

본 장에서는 앞에서 설계한 모듈들을 사용하여 POD 시뮬레이션 시스템 구현 내용을 보인다.

##### 4.1. 시뮬레이션 환경

시뮬레이션 환경으로 플랫폼은 Windows XP에서 개발 언어로는 PHP 4.x, 데이터베이스는 MySQL 5.0.77 community-nt, 웹서버는 APM\_APACHE2, 언어셋은 euc-kr에서 하였다.

##### 4.2. A4 편집 모듈 화면

편집 모듈에 대한 그림은 그림 5와 같다. 화면 아래 부분에 있는 이미지 파일 업로드 부분에서 파일을 업로드하여 처리한다. 이 부분은 찾아보기 버튼을 이용하여 탐색 브라우저를 통해 업로드된 파일을 선택하여 각 항목을 채우고, 우측의 “업로드” 버튼을 누름으로서 서버에 업로드를 진행한다.



그림 5. A4 편집 화면  
Fig. 5 Screen for A4 Editing

위의 순서는 업로드된 후 올려질 레이어의 순서와 같다. 레이어의 순서가 중요한 이유는 이미지가 겹쳐졌을 때 상위의 이미지가 어떤 이미지가 선택될 것인가에 영향을 미친다. 위의 순서로 인해 레이어의 깊이를 조절하고 이미 업로드된 이미지의 썸네일을 보여줌으로서 넓은 화면에서 볼 때와 각 객체구성으로 볼 때의 구성 요소를 판단하여 진행해야 한다.

또한, 업로드 옆의 삭제 버튼을 통해 임의로 업로드된 이미지를 제거할 수 있다. 별도의 삭제 버튼을 둔 이유는 레이어로 올라간 이미지에 대해 del 키를 이용해 삭제할 경우 편집모듈의 특성상 화면에서는 지워지지만, 실제 저장 공간에는 파일이 존재하는 상태로 남아있게 된다. 장기간 이런 이미지가 쌓이게 되면 후에 웹 저장공간의 낭비를 초래하여 불필요한 하드웨어를 추가로 구축하는 결과를 가져오므로, 이를 방지하기 위해 임의의 삭제 버튼으로 이미지를 삭제한다.

#### V. 결 론

최근에 인쇄물을 사용자가 직접 제작하여 실시간으로 인쇄할 수 있는 POD 시스템을 갖춘 웹사이트들이 생겨났다. 이러한 POD 시스템들은 기존의 인쇄 분야의 핵심 소프트웨어로 자리 잡고 있는 기존 오프라인 전문 편집 프로그램을 배우지 않아도 된다는 장점을 가지고 있지만 편집된 이미지의 질적인 면에서 여러 문제를 가지고 있다. 또한, 기본적으로 웹을 기반한 POD 시스템에서는 작은 크기의 인쇄물에 대해 편집이 가능한 것이 현실이다.

이에 본 논문에서는 멀티레이어 기법을 이용하여 A4 크기의 인쇄물 처리가 가능하다. 또한, 위치워 방식으로 웹상에서 작업된 인쇄 작업물에 대해 인쇄 요구가 있을 때, 작업물을 웹상에서 보여지는 모습 그대로 고품질로, COM 포트를 이용하여 스크린샷하여 이미지를 생성하고, 썸네일 형태로 관리된다. 이는, 실시간으로 인쇄소에서 사용할 수 있는 형태의 POD 시스템으로, 접근성과 사용의 편리성, 이미지의 고품질을 목표로 연구되었다.

본 논문에서 제안한 시스템은 A4 크기의 편집이 가능하고, 다양한 템플릿을 제공하여, 최소한의 데이터 작업만으로, 원하는 결과물을 얻는 것이 가능한 형태의 시스템에 대해 웹을 통해 구현함으로써, 타 시스템이나 프로그램에 비해 향후 확장성이나, 빠른 추가 기능의 개발이 가능한 환경을 구축할 수 있을 것으로 사료된다.

이는 웹기반 전자출판 및 e-book 분야에서 널리 활용될 것이다.

향후 연구로는 Ajax와 같은 최신 웹 기술로 전반적으로 적용하는 시스템의 개발과 HTML 5.0 기술로 확장하

여 다양한 미디어를 적용하는 방안의 연구가 계속되어  
져야 할 것이다.

### 참고문헌

- [1] 심정훈, 김정미, 새로운 도서 혁명 : Print-on-Demand, 유비쿼터스 트렌드 제3호, May 2006.
- [2] 조진환, “TEX:조판, 그 이상의 가능성,” The Asian Journal of TEX, 제1권, 제 1호, April 2007.
- [3] 김도현, “GPL 폰트를 내장한 PDF 전자문서의 저작권,” 동국대학교 비교법문화연구소, 비교법연구, 제6권 제1호, 2005.
- [4] 이지현, “Ajax 기반의 웹 2.0 서비스를 위한 인터랙션 디자인 가이드라인에 대한 연구,” 제29권, 제4호, pp. 569-574, 2010.
- [5] 정우성, 이은주, “CSS 가독성 향상을 위한 최적화 기법,” 한국컴퓨터정보학회, 제15권, 제7호, July 2010.
- [6] 김경원, 임태범, 이석필, “이미지 특성 추출 정보를 이용한 콘텐츠 검색 엔진 설계,” 대한전자공학회, 제33권, 제1호, 2010.
- [7] 홍지영, 채행성, 조운정, 정대현, 김종완, 김성은, 이해정, 한광희, “Internet Protocol TV 환경에서 효율적인 웹 탐색기법의 사용성에 대한 비교 연구,” Journal of the Ergonomics Society of Korea, 제 27권, 제4호, 2008.
- [8] Heman. A, “Image Browsing on a Large Display,” Information Technology Interfaces. 2007, ITI 2007. 29th International Conference on. pp. 245-250, 25-28, 2007.
- [9] M. Keller and M. Nussabaumer, “Cascading style sheets: a novel approach towards productive styling with today’s standards,” in Proc. of the International Conference on WWW, 2005.
- [10] 김선희, 이용환, “Adobe Photoshop에서 인쇄목적으로 한 색상모드 변환에 관한 연구- RGB에서 CMYK로 변환된 데이터를 중심으로-,” 한국사진학회지, 제18호. pp. 6-19, 2008.

### 저자소개

#### 이병권(Byung-Kwon Lee)



2002 : 한밭대 전자공학과 학사  
 2010 : 한밭대 전자공학과 석사  
 2003년~2007년 : (주)한백전자  
 2007년~2011년 : (주)엠택

2012년~ 현재 새라코아 연구원  
 ※관심분야: 모바일 시스템, 임베디드 시스템, 웹기반 시스템

#### 김학진(Hak-Jin Kim)



1978년 한남대학교 화학과 (이학사)  
 1980년 숭실대학교 화학과 (이학석사)

1988년 한남대학교 화학과(이학박사)  
 2010년~현재 배재대학교 산학협력단 교수  
 ※관심분야: 기술창업 및 사업화, U-Health, IT 융합기술

#### 정희경(Hoe-Kyung Jung)



1985년 광운대학교 컴퓨터공학과(공학사)  
 1987년 광운대학교 컴퓨터공학과(공학석사)

1993년 광운대학교 컴퓨터공학과(공학박사)  
 1994년~현재 배재대학교 컴퓨터공학과 교수  
 ※관심분야: 멀티미디어 문서정보처리, 웹기반 시스템, XML, SVG, Web Services, Semantic Web, MPEG-21, Ubiquitous Computing, USN