

웹을 기반으로 한 PDF 출판 솔루션에 관한 연구

이 재 득*

한성대학교 산업시스템공학과

PDF Publication Solution based on Web

Jae-Deuk Lee

Department of Industrial Systems Engineering, Hansung University

In the previous C/S publishing system, the editor or contributor can arbitrarily modify the document created by the author, in which case it is difficult to identify the changes made in the document. Another shortcoming is in that when the document is in need of tracking or editing, the client must have the respective editing system. To solve this problem, the gist of the document must be preserved along with the document itself, and the process of handling the document must be standardized.

Publishing on the web ensures a more stable and accurate result in processing documents. The significance of web publishing is made clear, when we consider the importance of information per se and the growing demand for immediate publication in the present day.

The need for a simple and straightforward apache-based PDF publishing system, in which HTML and CSS are supported, and a converting engine provides PDF standard security application support, is prominent.

This provides a library in which one can directly create a PDF via Windows, Linux, or Unix without having to rely on a client, allowing high-speed PDF creation. The development of a web-accessed PDF converting engine forms the basis for e-transactions, online brochures, electronic B/L, and many other industrial sectors.

Keywords : Web, PDF, Publication Solution,

1. 서론

PDF(Portable Document Format)의 형식은 1991년 San Jose에서 개최된 Seybold Conference에서 Adobe사에 의해서 처음 시작이 되었다.(8) 그 당시에는 현재 사용되는 PDF가 아니라 'Interchange PostScript'의 약자인 IPS 라고 제안이 되었다. 그리고 1992년 가을에 Comdex(에서 Adobe사의 Acrobat Version 1.0이 처음으로 출시되어 현재 Version 6.0까지 발전되었다.(9) 현재 대부분의 PDF는 Adobe사의 Acrobat에 의하여 이루어지고 있다. 기존의 Client/Server 출판환경에서는 저자(Author)가 생성

한 문서가 유통하는 과정에서 편집자(Editor) 또는 기여자(Contributor)에 의해 수정, 변형등이 일어날 경우 그 과정을 알 수 없어 문서의 책임이 불명확해지고 문서를 조회하거나 편집할 경우 클라이언트에 반드시 해당 편집기가 있어야 하는 제한사항이 있다. 이러한 문제를 해결하려면 문서를 저장할 때 문서의 맥락정보와 문서의 내용 자체가 함께 저장되어야 하고 그러기 위해서 문서의 생산, 유통 과정이 표준화 되어야 한다.(3)(4)

웹 환경의 출판은 문서를 보다 견고하고 프로세스 상 정확한 결과를 갖도록 보증할 수 있다. 정보자체의 중요성이 증대하고 즉시 출판의 필요성이 늘어나는 오늘날

* 본 연구는 2005년도 한성대학교 교내연구비 지원 과제임.

의 정보환경을 고려한다면 웹 환경 출판의 필요성은 분명해진다.(5)(6)

이에 HTML과 CSS를 지원 하고 PDF 표준 보안 적용 지원 하는 변환 엔진을 개발하고 웹 출판 공정을 단순화하여 Apache 기반의 PDF Publishing 시스템을 구현하는 것이 필요해진다.(7)

윈도우, 리눅스, 유닉스 서버상에서 클라이언트의 의존 없이 직접 PDF를 작성하는 라이브러리를 제공하여 고속으로 PDF 저작 시스템을 제공할 수 있게 되었고 웹 환경에서 쉽게 사용할 수 있는 PDF 변환 엔진이 개발됨으로서 전자결재 시스템, 온라인 브로슈어등 산업 여러 분야에 응용할 수 있다.

2. C/S 환경 출판 및 웹 환경 출판

2.1 문서의 생산과 유통에서 출판의 역할

컴퓨팅 환경이 사무실 환경을 점차 벗어나 “언제 어디서나” 활용할 수 있는 편의성이 강조됨에 따라 전산 자원을 구현하고 활용할 때 웹 서버 또는 어플리케이션 서버를 기반으로 구축하는 것이 점차 대세를 이루고 있다. 다소 좁게 말한다면 “웹”이니, “자바”니, “엔터프라이즈 자바 빈”이니 하는 다소 오래된 이슈에서부터 지난 1988년 제록스사의 팰러앨토 연구소(PARC)의 마크 와이저(Mark Weiser)가 처음 제시한 “유비쿼터스(Ubiquitous 컴퓨팅)”에서도 이것이 강조되고 있는 것이다.

문서를 생산하고 유통하는 면에서도 문서를 조회하거나 메시지를 교환하는 수준에서는 상당한 진전이 있어 온 것이 사실이며 문서가 모든 전산 분야 중에서 가장 먼저 전산화되었고, 사용자에게 1차적 환경이 되는 만큼 당연하다고 할 수 있다. 그러나 자세히 살펴보면 문서 생산 및 유통만큼 클라이언트-서버(C/S) 환경이 분명히 구분되고 구식의 방법이 많이 적용되는 분야도 별로 없는 것이 사실이다. 이는 문서의 양식과 포맷을 중요시하는 동북아 3국(한,중,일)에서는 특히 더한 것으로 생각된다. (1)(2)

이 연구에서는 다소 뭉뚱그려 표현하는 “문서환경”을 좀더 세밀히 정의하여 Life cycle of documentation [문서의 생애]로 바라보고, 문서를 웹 환경에 적용시킬 수 있는 방안을 제시하고자 한다. 특히 정형화된 표현 방법으로서 웹 환경 출판의 가능성을 제시하여 전산 환경에서 “유비쿼터스”적인 문서 활용을 이룰 수 있는 방안과 그 산업적 적용 방안을 모색하고자 하는 것이다.

전산화 이전 단계에서 문서란 생산자가 모든 것을 완성하는 것으로 생각되었지만 이런 개념에서 보면 문서

는 “문서 자체”에 지나지 않는다. 전산화가 진척된 오늘날에 있어서 개인이 완성한 문서는 선언적인 의미 외에 완성된 문서로 보기 어려우며 하다못해 인터넷에 올리기 전에는 문서라고 부를 수 없는 것이다.

문서 또는 documentation을 종이 또는 paper와 구분한다면 문서가 문서 생산자에 의해 게시되고 그것이 권위 있는 기관, 하다못해 문서 생산자 자신에 의해 인증되어야 문서가 될 수 있는 것이다. 이것은 공공기관, 학교, 기업 등에서 내부 결재를 거치고 나서야 문서가 실제 공신력을 갖게 되는 것을 연상할 수 있을 것이다. 그런 과정을 거치지 않은 문서는 사회에서는 용인하지 않고 피문서라든가, 공식입장이 아니라든가 하고 외면당하게 되는 것이다.

이런 측면에서 문서는 하나의 생산 과정을 거친다고 볼 수 있으며 문서의 생산 과정은 [문서의 집필] - [문서의 검중] - [문서의 수정] - [문서의 출판] 단계를 거치게 된다. 각각 저자, 편집자, 기여자, 출판자 / author, editor, contributor, publisher가 그 역할을 담당하게 된다. contributor는 다소 상이한 개념이나 여기서는 단순화하여 표현하고자 한다.

이렇게 만들어진 문서는 다시 유통 과정을 거치게 된다. 일단 출판된 문서는 [배포]되며, 제3자에 의해 [인용] 또는 [활용]될 수 있고, 최종적으로 [저장]되거나 [폐기]된다. (distribution, quotation, reuse, archive, deprecated) 이렇게 문서가 생산되고 최종적으로 저장 혹은 폐기되는 전 과정을 Lifecycle of Documentation 문서의 생애라고 한다.

문서의 출판은 위 과정에서 보듯 생산 과정을 종료하고 유통 과정을 시작하는 단계이며 영어권에서 출판, Publication이 공공화라는 뜻을 갖는 것도 이런 단계를 반영하는 것이다. 출판사에서 책을 내는 것과 마찬가지로 문서의 출판이란 독자에게 완성된 문서를 주는 것이다. 즉, 한편에서는 문서의 생산자로서 권위를 인증하고, 한편에서는 문서의 소비자에게 문서를 제공하는 시점이 시작된다는 것이다.

2.2 전산환경 변화와 출판 환경 변화

C/S 환경 아래서 출판은 위와 같은 출판의 개념을 정립하는 과정이었다고 볼 수 있다. 공공기관 또는 기업에서 전산환경이 충분히 마련되지 않은 단계에서는 출판과 유사한 개념이 주로 “결재”에 의존했다. 결재 방법에 따르면 결재 대상 문서의 정보를 담은 별지를 작성하고 해당 별지에 피결재자와 결재권자의 사인이나 도장을 받는 방법이 적용되었다. 이 방법은 오늘날에도 전자 결재 시스템에 남아 있다.

전통적인 결재 방법은 특정 업무 프로세스를 진행하는 기록을 남기고 명령을 집행하는 목적에 사용되었으므로 업무-과정-기록 문서에 한정되었던 것이, C/S 환경이 등장함에 따라 출판의 개념이 보다 확장될 수 있었다.

문서 생산자가 개인용 터미널에서 작성하는 문서가 기업의 문서 저장소에 기록되고 이를 문서 자체에 저장될 수 있는 메타정보를 기록하는 방법이 만들어지면서, 문서의 생산단계부터 해당 정보가 기록될 수 있게 된 것이다.

다만 이 상황은 미국, 유럽 등과 우리나라의 상황이 다소 다른 것이 사실이다. 우리나라의 경우 문서에 메타정보를 기록하는 것이 상당히 드문 일인 반면 미국의 경우 비서(secretary) 시스템이 발전하여 생산자의 문서에 수십개의 메타정보가 부여되는 것이 일반적이다. 사회적인 시스템 차이에 따라 문서의 생산에 비해 문서의 기록 방법에 차이가 나는 것이다.

최근까지 우리나라 문서 관리 체계에서 가장 많이 선호되는 방법은 기업의 문서 저장소가 메타정보 역할을 할 수 있도록 여러 가지 폴더를 만들고 이 폴더에 문서를 저장하는 방법이다. 이 폴더에 저장되는 문서가 출판 가능한 문서인지 여부는 개인의 판단이나 메일 등을 통해 문서를 공유하여 검증하는 다소 수작업에 가까운 방법이 적용되었다고 할 수 있다. 사실 이런 “전통” 덕분에 우리나라에서 문서의 출판이란 것이 다소 낯설다고 할 수 있다.

C/S 환경에서 웹 환경으로 전산 환경이 변화한 최근 수년간, 문서 환경에서도 변화가 찾아오고 있다. 특히 문서 유통에서 변화가 시작되고 있다. 지식관리시스템(KMS, Knowledge Management System)이 비교적 적은 비용으로 높은 가시적 효과를 얻을 수 있다는 것이 알려지면서 공공기관부터 일반기업에 이르기까지 많은 기관이 이를 도입하고 있다.

KMS에서 핵심적인 기능은 물론 문서의 유통이다. 여전히 문서의 생산은 개인의 PC에서 이루어지고 있지만 많은 문서가 “등록자”의 명의를 달고 유통되기 시작한 것이다. 일단 웹에서 유통되기 시작한 문서는 다시 오프라인을 기준으로 유통될 수 없을만큼 편리한 기능을 제공한다. 최소한 내 개인PC를 벗어나 업무 가능 영역 어디서나 손쉽게 문서를 읽을 수 있고 업무 파트너도 이를 읽을 수 있다는 점만으로도 상당한 혁신을 가져온 것이다.

출판 환경의 변화도 이와 함께 시작되고 있다. 콘텐츠 관리 시스템(CMS, Content Management System)은 KMS만큼 주목받고 있지는 못하지만 대학, 기업을 중심으로 도입되고 점차 공공기관으로 확산되는 추세이다. KMS가 유통 측면에서 웹 환경 도입이라면 CMS는 생산 측면에

서 웹 환경 도입이라고 할 수 있다. 여기에는 각종 인터넷 포털과 블로그를 통해 웹을 통한 문서 생산에 많은 사람이 익숙해지고 있는 것도 한 몫을 하고 있는 듯하다.

<표 1> 출판 환경의 변화

비교 항목	전통적 결재	C/S 환경	웹 환경
대상 문서	업무프로세스 문서	공식 문서	모든 문서
출판 방법	결재권자의 결재	결재 정보의 추가	출판의 인증
유통 방법	원본 전달, 사본 작성	Email, 폴더 공유	검색 및 조회
주요 시스템	문서 수발 체계	문서관리 시스템	CMS, KMS

2.3 웹 환경 출판의 역할 및 중요성

웹 환경 출판의 중요성은 특히 프로세스 표준화 측면, 유틸리티 측면, 아카이브 측면에서 확인할 수 있다.

프로세스 표준화 측면에서 검토해 보면 현재의 전산 환경에서 문서의 1차 작성자, 저자[Author]가 제시한 문서의 내용은 편집자[Editor] 또는 기여자[Contributor]에 의해 손쉽게 수정될 수 있다. 즉, 잘못된 주장이나 부정확한 사실을 전문가가 발견하고 수정하는 것이 가능하며 이를 저자가 반영하는 것도 순쉬운 공정으로 이루어질 수 있는 것이다.

C/S 출판 환경에서도 가능한 일이지만 현실적으로 C/S 출판에서 저자는 개인이 선호하는 워드프로세서를 사용하는 경향이 있다. 예컨대 MSWord나 한글 워드프로세서를 사용하기 쉽다는 것이다. 이런 C/S 출판용 프로그램을 사용하는 경우 편집자나 기여자가 해당 응용 프로그램을 갖고 있어야할 뿐만 아니라 수정 작업을 끝낸 파일을 다시 등록해야 하는 번거로움이 생기게 된다. 번거로운 것은 사실 사소한 문제이다. 이렇게 수정한 사실을 정보로서 등록하는 것이 불가능한 것이 문제이다. 수정한 파일이 정확한 출판 공정상의 위치에 - 예컨대 수정되었다든가, 특정 부분의 사실 보충을 요청했다든가 하는 프로세스 - 있는지 여부를 판별하려면 해당 출판물에 대한 작업이 서버 상에서 이루어져야 하는데, 기존 C/S 출판 환경에서는 이것이 클라이언트에서 이루어진다는 데 문제가 있다.

웹 출판 환경에서는 사용자의 작업 환경에 서버 프로세스와 사용자 정보가 그대로 남아있으므로 편집자나 기여자가 편집이나 수정을 가한 후 등록하기만 하면 특

정 프로세스 상에서 진행된 작업이라는 점이 복잡한 공정 없이 보증되는 것이다. 즉, 웹 출판 환경은 서버의 프로그램을 응용하여 하나의 작업 기관에서 문서의 생산부터 유통까지의 단계를 표준화하는 것이 가능한 반면, C/S 출판 환경은 표준화 작업을 사용자 교육을 통해 수행해야 하며 검증 수단이 적합하지 않다는 것이다.

유비쿼터스 측면에서 살펴 볼 때 그 장단점은 보다 분명해 진다. C/S 출판 환경에서는 사용자가 원본 파일을 작업하기 위해 자기 터미널로 돌아와야 하거나, 작업 터미널을 원래의 자기 터미널과 마찬가지로 구성해야 한다. 작업자가 특정 터미널에 의존하게 되므로 언제, 어디서든 작업할 수 없다는 것이다.

반면 웹 출판 환경은 웹 브라우저가 있는 환경이라면 어디서든 작업할 수 있으며, 작업 환경을 구성하는 수준과 범위에 따라 모바일 장비까지 포함하여 어떤 환경에서든 용이하게 문서를 작성 및 수정할 수 있게 된다는 점에서 큰 이점이 있다. 문서를 읽는 쪽의 입장에서도 별도의 응용프로그램 없이 이런 작업을 수행할 수 있으므로 훨씬 편리하다고 할 수 있다.

아카이브 측면은 위의 두 측면과 모두 연관되어 있다. 문서는 일정한 기간 활용한 후 폐기 문서와 저장 문서로 분류된다. 이때 문서는 단지 문서의 내용으로만 판단되는 것이 아니고 문서를 생산하고 유통할 때의 맥락에서 판단될 수 있다. 예컨대 1월1일자로 만들어낸 성명서가 3월1일 발표될 수 있다. 이때 문서의 의의는 3월1일 발표된 점이다. 또한 문서의 책임소재를 밝힐 때 문서의 최초 생산자보다는 이를 검증한 편집자나 출판자가 더 큰 책임을 져야 할 수도 있다.

문서를 저장할 때 문서의 맥락정보와 문서의 내용 자체가 함께 저장될 수 있으려면 물론 문서의 생산, 유통 과정이 표준화되어야 한다. C/S 환경에서 이런 생산, 유통 과정에 대한 정보가 기록된다는 보증이 없고 - 신뢰할 수 없는 정보라는 것이다. - 그렇게 기록할 정보의 수준도 낮다는 점을 볼 때 문서의 아카이브[자료관 이관] 측면에서 웹 환경의 필요성이 매우 높다고 할 수 있다.

또한 저장된 문서는 일정 기간이 지난 후에 다시 재활용될 가능성이 높다. 정확히 말하면 일정 기간이 지난 후에 재활용할 문서를 아카이브한다. 일정 기간이 지난 후에 해당 문서를 읽거나 활용할 수 있으려면 당연히 해당 문서의 포맷이 저장하기에 적합해야 한다. 웹 환경에서 HTML 혹은 XML 기반의 데이터는 견고하므로 (견고하다는 것은 일부 데이터가 유실되더라도 원본을 유추하기에 쉽다는 것을 의미한다.) 상당기간이 지난 후에도 활용할 수 있는 반면, C/S 환경의 문서들은 응용프로그램 제작사의 상황이나 지원 포맷의 변화 등으로 인해 더 이상 활용되지 않을 가능성이 있다.

즉, 사용자가 문서를 읽을 때 제한이 덜한 웹 환경의 문서가 아카이브한 경우에도 장기적으로, 견고성 측면에서 유리하다는 것이다.

결국 웹 환경 출판은 문서를 보다 견고하고 프로세스 상 정확한 결과를 갖도록 보증해 주게 된다. C/S 환경 출판이 완전히 사라지거나 그 역할이 없어지지는 않겠지만, 정보 자체의 중요성이 증대하고 즉시 출판의 필요성이 늘어나는 오늘날의 정보환경을 고려한다면 웹 환경 출판의 비중이 더욱 커지게 될 것임은 분명한 사실이라고 할 수 있다.

3. 웹 기반 PDF Publishing의 적용 및 전망

3.1 적용 방안 및 전망

웹 환경 기반 출판의 장점을 위에서 지적했으며, 이와 관련하여 KMS, CMS의 도입이 이루어지고 있다. 여기서 한가지 지적해야 할 것은 HTML, XML 기반의 문서 유통이 갖는 문제점이다. 문서의 저작 방법으로 HTML, XML이 이미 지적한 장점을 갖고 있기는 하나 단점 역시 갖고 있다.

<표 2> HTML/XML의 단점 비교

	HTML ,XML	PDF
데이터 소스의 복사	용이	곤란
보관 데이터	이미지, 텍스트 등 여러 파일	한 개의 파일
출력 시 동일성	브라우저, 시스템별로 상이	동일
가독성	낮음	높음

웹 기반 Publishing을 적용하고자 할 때 가장 큰 단점은 해당 문서를 출력했을 때 시스템이나 브라우저 별로 출력 내용이 상이하다는 점이다. 웹 기반 Publishing이 비록 종이없는 사무실(Paperless Office)을 구현하는데 사용되기는 하나 상이한 폰트는 사용자에게 상이한 내용으로 인식될 가능성이 크기 때문에 출력 내용이 상이하다는 점은 문제점으로 볼 수 있다.

같은 문제로 웹 기반 Publishing을 도입함으로써 문서의 가독성이 떨어진다는 점에서 오히려 웹기반 Publishing에 의한 퇴보의 가능성도 있다고 할 수 있다.

실제 사무실 환경에서 웹 기반 Publishing을 구현할 때 하이퍼링크나 외부 데이터를 반영하여 기록할 수 있다는 점도 문제점으로 지적되고 있다. 즉, 웹 페이지를 잘

라붙여 문서를 작성하는 경우 해당 페이지(특히 이미지)는 웹페이지가 아니라 URL로 참고만 되어 있으므로 일정 시간이 경과하여 해당 웹페이지의 내용이 변경되는 경우 출판물의 내용이 사라지거나 변경되는 문제가 발생하게 됩니다.

PDF Publishing은 웹기반 출판의 보완 방법으로서, 출판 단계에서 HTML/XML을 PDF로 변환하여 해당 출판물을 고정시키는 역할을 합니다. PDF로 변환하는 단계에서 하이퍼링크된 데이터나 여러개로 분산되어 있는 리소스를 하나의 파일 안에 저장하게 되므로, 일정한 기간이 경과한 후에도 해당 내용이 변경되지 않는 효과를 갖게 되는 것이다.

부수적으로 PDF의 특성 상 출력물과 동일한 품질을 지니게 되므로 상이한 플랫폼에서도 동일한 출력물을 가질 수 있고, 플랫폼이나 브라우저가 바뀌는 경우에도 원본과 동일한 내용을 서비스할 수 있게 된다.

이러한 방법은 웹 환경 출판의 장점을 그대로 살리면서 웹 환경 출판의 단점을 보완하는 것으로 사용자의 편의성을 높이면서도 표준성, 아카이브성, 유비쿼터스성을 모두 보장하는 방법이다.

본 연구의 범위를 벗어나는 것이지만, C/S 환경에서 생산된 문서를 웹 환경에서 생산된 문서와 별도로 관리하는 것이 불가능하다는 측면을 고려할 때, C/S 환경에서 생산된 문서를 PDF로 변환하여 저장하여 두는 경우, 웹 환경이 C/S 환경의 문서까지 수용할 수 있다는 점도 큰 장점이 될 수 있다.

즉, C/S 환경에서 만들어진 문서를 웹 환경으로 이관하여 이용하는 경우 사실상 사용자 측면에서는 C/S 환경과 그다지 큰 차이점이 없는 것이고 시스템 유지 측면에서도 C/S 환경을 그대로 유지하는 데 그치는 반면, 웹 환경을 PDF로 보완하는 경우 C/S 환경을 완전히 웹 환경으로 이동하는 것이 가능해 진다는 것이다.

3.2 사례 - 전자 결제

웹 기반 PDF Publishing 시스템을 적용할 수 있는 가장 좋은 사례는 전자 결제이다. 결제 공정은 공정 중에 여러번 수정을 거치고 수정된 내용을 인증하여 출판되어야 하므로 가장 전형적인 출판의 형식을 지니고 있다.

반면, 전자 결제의 대상이 되는 문서는 전자결제의 공정 정보는 물론, 결제와 관련된 본문, 그리고 첨부 문서로 구성하게 된다. 이를 각각 별도의 시스템 저장소에 저장하는 경우 원적으로 해당 문서가 배포되는 경우 첨부문서의 누락이나 공정 정보의 누락과 같은 문제가 수시로 발생하게 되는 단점이 있다. 또한 웹 환경에서 결제 본문만을 대상으로 한다고 해도, 이미 앞에서 지적한

웹 문서 복사하기에서 발생하는 결제 시점 원문과 일정 시간이 경과한 결제문이 변경되는 문제가 발생하게 된다. 특히 결제문의 변경 문제는 이미 전자우편을 기반으로 하는 결제 시스템에서는 지속적으로 문제가 발생하는 이슈이기도 하다.

본 연구의 웹기반 PDF Publishing 시스템을 이용하는 경우 결제 정보 본문을 PDF로 변경하는 것은 물론 첨부문서를 PDF로 변환하여 단일한 결제 문서를 변경하는 것이 가능하다. 이 방식을 이용하면 결제 시점의 메타정보(Meta data)와 결제 본문 및 첨부 파일을 단일한 파일로 관리하게 되어 결제 문서 자체의 유통 시에도 파일의 누락 및 잘못된 정보 전달의 우려가 없을 뿐만 아니라 중요한 결제 정보를 일정 시점이 지난 후에 파악할 수 없는 단점을 해결할 수 있다.

3.3 사례 - 온라인 브로슈어

CMS와 같은 적극적인 활용 외에도 온라인 브로슈어와 같은 ASP 모델을 활용할 수 있다. 이러한 적용 방법은 종이 문서를 웹에서 생산하는 방법으로 적극 검토할 수 있다.

일반적으로 중소기업의 기업은 제품의 카달로그나 회사 소개서를 작성하기 위하여 많은 비용을 사용하지 못한다. 전문업체에 해당 내용을 위탁하는 비용이나 이를 검증하는 데 필요한 시간을 충분히 갖지 못한다는 것이다.

온라인 브로슈어는 웹 형태로 사전에 고안한 템플리트를 제공하여 서비스하는 것을 의미하며 이러한 전략은 이미 포털사이트에서 홈페이지 템플리트를 제공하는 형태로 이루어지고 있다. 이러한 템플리트를 브로슈어나 카달로그로 사용하지 못하는 이유는 이것의 출력형태가 상이하다는데 있다.

이러한 온라인 브로슈어를 PDF로 변환하는 방법으로 출력형태가 상이한 문제점, 폰트가 없는데 따른 미려하지 않은 출력이 나타나는 문제점 등을 해결할 수 있으며, 이를 이용한 ASP 서비스가 가능하다는 것이다.

이를 이용하여 단체나 협회 등에서 회원사에게 적합한 브로슈어 템플리트를 제공함으로써 상품 유통을 활성화시키고 고품질의 제품 및 회사소개서를 수시로 변경하여 제공할 수 있는 기반을 제공할 수 있게 된다.

본 연구를 이용하는데 따른 장점은 손쉬운 HTML 템플릿 작성의 이점과 미려한 PDF 출력의 장점을 결합한다는데 있으며, 동일한 아이디어에 따라 광범위한 활용이 가능하다.

4. 웹 기반 PDF Publishing 시스템의 구현

4.1. 개요

본 연구에서 핵심적인 부분은 HTML, XML을 PDF로 변환하는 변환엔진이다. HTML 및 XML은 근본적으로 페이지 개념이 없이 위에서 아래로 흐르는 포맷 형식을 갖고 있으며, 페이지 크기의 개념도 화면 지향적인 성격을 갖는다. 반면 PDF는 종이 문서의 개념을 기반으로 하여 특정한 크기의 페이지에 문서의 내용을 기록하기 위한 포맷이다. 따라서 HTML 및 XML을 PDF로 변환하기 위해서는 HTML 및 XML의 정보를 제한하여 PDF로 변경하는 개념이 필요하다.

HTML은 CSS를 스타일 언어로 사용하고 XML은 XSL을 스타일 언어로 사용한다는 점도 PDF 변환에 있어서 일관성을 갖는데 문제점이 된다. 본 연구에서는 XML을 XSLT를 이용하여 HTML로 변환한 후 HTML-CSS 기반의 변환으로 적용하는 방법을 채택했다.

PDF 문서는 포맷의 내용과 사용이 일반에게 허락되어 있는 Public Domain 정책을 취하고 있다. PDF 문서의 생성을 위해서 한국비즈넷이 개발한 PDF 생성 라이브러리를 그대로 사용하기로 하였다. 웹 기반 PDF Publishing 시스템의 구현을 위해서 웹 출판 공정을 단순화하고 이를 Apache 서버를 기반으로 구현하였다.

4.2. HTML 문서의 FO 정의 및 구현

HTML의 각 태그에 대한 스타일을 PDF 변환 시 적합한 스타일로 변환하기 위하여 CSS 태그를 응용하였다.

다음 HTML 태그는 비스타일 태그로서 변환시 무시하도록 설계하였다.

```
<html> <header> <meta> <body> <tbody>
```

이외 HTML 태그는 스타일시트 파일 또는 스타일 선언에 따라 CSS 블록정보와 연계하여 스타일세트로 번역하였다. 스타일세트는 다음과 같이 정의하였다.

<표 3> HTML 스타일 세트

	설 명	하위 정보
Character 세트	문자 모양 정보	폰트, 크기, 색상 등
Border 세트	외곽선 모양 정보	외곽선 스타일, 굵기 등
Floating 세트	위치 정보	상대 위치, 고정 위치 등
Paragraph 세트	문단 분기 정보	앞뒤 문단 위치 등

HTML은 Cascade 모델 [겹겹이 둘러싼 상자 모양]로 스타일을 정의하므로 HTML 태그 트리를 기준으로 본 스타일 정보를 해석한 FO를 생성하도록 하는 방법으로 구현하였다.

<table>의 정의와 구현은 예외적인 구현으로서 <table>은 그 특성상 별도의 페이지 수준의 파싱을 요구하게 되므로 파서가 table을 처리할 때마다 그 결과에 따라 HTML 문서 전체를 재처리해야 하므로 재귀적으로 파서를 구현하였다.

이에 따라 <table>이 많이 포함된 문서의 경우 FO 작성 시간이 기하 급수적으로 늘어나는 문제가 발생하였다. 이를 웹 브라우저와 비교한 결과 동일한 문제가 발생하는 것으로 HTML 파싱의 특징으로 보았다. 다만, PDF 구현 시 좌우 페이지가 넘치는 문제가 발생하는 바, 페이지 크기의 한계를 주어 일정 크기가 넘어가는 HTML 부분을 PDF 파일에서 화면 밖으로 처리하게 되었다.

4.3. XHTML 및 XML의 FO 정의 및 구현

XHTML은 확장된 HTML로서 HTML 구현과 동일하다. XML은 XSL 스타일시트를 이용하여 스타일을 구현한다. 다만 XSL 구현에 있어서 XPath와 같은 확장된 기능은 사용자와 대화형으로 구현되므로 PDF와 같은 정적인 페이지에 적합하지 않은 관계로 구현하지 않기로 하였다.

XML-XSL을 직접 구현하는 공개 파서를 이용하는 방법을 검토하였으나, FO 정의 및 구현에 있어서 오히려 비용이 많이 드는 단점을 고려하여 XSLT를 이용하여 XML을 HTML로 변환하는 방법을 채택하였다.

XSLT의 이용은 CSS의 이용에 비해서는 다소 어렵거나 XML의 다른 스타일 정의 방법에 비해서 가장 쉽고, 또한 XML 표준에 의거할 수 있다는 점 때문에 채택하였다.

XSLT는 MS사의 공개 파서인 MSXML 파서와 IBM사의 공개 파서인 Sax 파서를 적절한 상황에서 사용할 것을 제안하였다. MSXML 파서는 윈도우 시스템에서만 구동되는 단점이 있으나 속도 및 신뢰성 측면에서 효과적이다. IBM의 Sax 파서는 여러 시스템에서 구동되기는 하나 시스템이 비교적 무겁고 사용법이 복잡한 단점이 있다.

5. PDF 변환과 웹 기반 출판 시스템의 구현

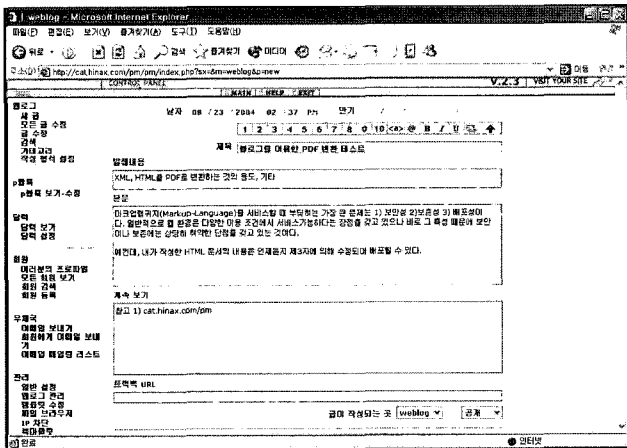
HTML/XML에서 생성된 FO를 PDF로 변환하기 위해

서 PDF 버전 1.5를 응용하였다. PDF의 모든 스펙을 구현하기 보다는 PDF의 스펙 중 Formatting 정보를 처리하는 부분을 중심으로 구현하였으며, 이를 통해 개발 기간을 최소화하였다.

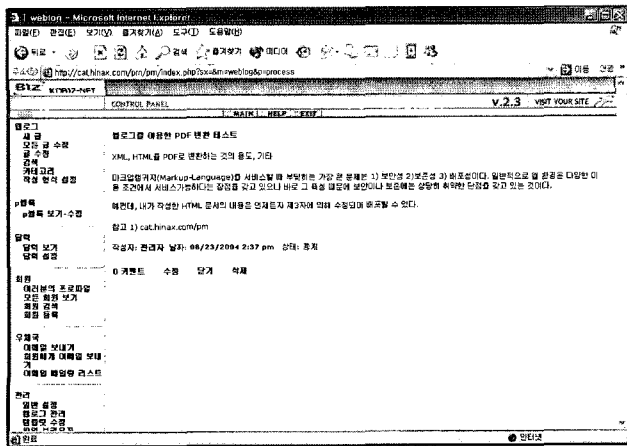
웹 기반 출판 시스템의 핵심엔진인 HTML/XML의 PDF 변환을 주 대상으로 하고, 이를 구현하는 웹 기반 출판 시스템을 예시하는 측면에서 다음의 페이지를 구현하였다.

- * 저작 페이지 - 이름, 비밀번호, 제목, 요약, 내용 등록, 등록 페이지, 삭제 페이지, 수정 페이지, 출판 페이지, 검색 페이지, 조회 페이지
- * 관리 페이지 - 출판 관리, 목록 페이지, 삭제 페이지, 검색 페이지, 조회 페이지
- * 조회 페이지 - 조회 관리, 검색 페이지, 조회 페이지

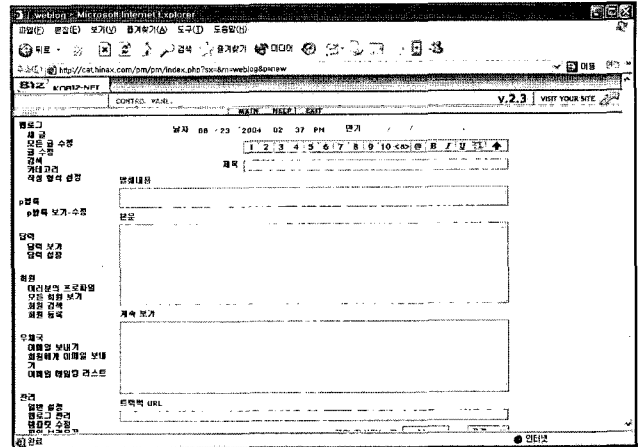
위의 각 페이지중에서 등록을 위한 입력화면 <그림 1>, 등록된 문서 <그림2>, 글쓰기 화면 <그림 3>, 그리고 댓글 화면 <그림 4>를 보면 이해가 쉽게 되리라고 생각한다.



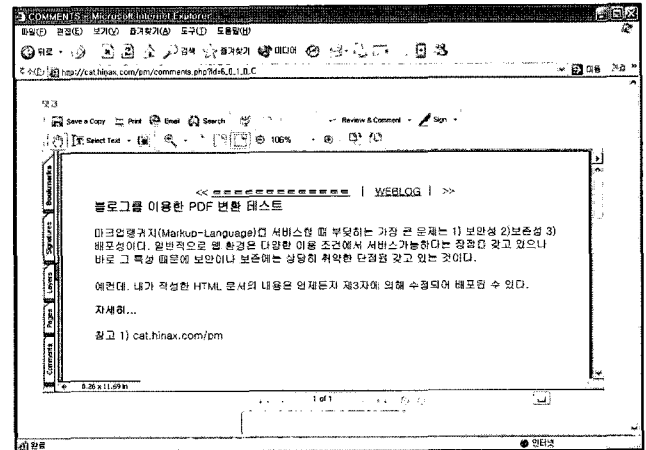
<그림 1> 등록을 위한 입력화면



<그림 2> 등록된 문서



<그림 3> 글쓰기 화면



<그림 4> 댓글 화면

6. 결 론

HTML과 CSS를 지원 하고 PDF 표준 보안 적용 지원하는 변환 엔진을 개발하고 웹 출판 공정을 단순화하여 Apache 기반의 PDF Publishing 시스템을 구현하려고 하였다. 즉 이 연구의 목표는 1) HTML, XML 문서의 PDF 변환 엔진 2) 웹 기반 출판 시스템의 구현 이었다. 이를 구체적으로 표현하면 1)CSS 태그 응용(HTML의 태그를 PDF변환시) 2)XSLT를 이용하여 XML을 HTML로 변환(MSXML 파서, Sax 파서)하는 것이라고 할 수 있겠다.

이는 윈도우, 리눅스, 유닉스 서버상에서 클라이언트의 의존 없이 직접 PDF를 작성하는 라이브러리를 제공하여 고속으로 PDF 저작 시스템을 제공할 수 있게 되며 웹 환경에서 쉽게 사용할 수 있는 PDF 변환 엔진이 개발됨으로서 전자결제 시스템, 온라인 브로슈어, 전자 B/L등 산업 여러 분야에 응용할 수 있기 때문이다. 현재

웹 기반 PDF 출판솔루션 엔진이 개발중에 있으며 물류 포워더의 보안 및 분명한 책임소재가 필요한 원본 B/L 발행 업무에 적용할 수 있는 결과가 기대된다..

참고문헌

- [1] 이명식, “한글 PDF와 인터넷서버”, 디넷소프트, 1999
- [2] 이송만, “액로벳 4.0K PDF활용”, 사이버출판사, 2000
- [3] Joseph Marin, Julie Shaffer, “The PDF Print Production Guide”, Graphic Arts Technical Foundation, 2003
- [4] Gordon Kent, “Internet Publishing With Acrobat : A Comprehensive Reference for Creating and Integrating PDF Files With HTML on the Internet or Intranets”, Adobe Press, 1996
- [5] Thomas Merz, “Web Publishing With Acrobat/ PDF”, Springer-Verlag, 1998
- [6] Frank J. Romano, “PDF Printing and Workflow”, Prentice Hall, 1998
- [7] Sid Steward, “PDF Hacks”, O'Reilly, 2004
- [8] <http://www.prepressure.com/pdf/history/history01.htm>
- [9] <http://www.adobe.com/products/acrobat/main.html>