

이동단말을 위한 적응적 웹 문서 변환

(Adaptive Conversion of Web Content for Mobile Terminals)

강 성 천 [†] 정 광 수 ^{**}

(Sueng-chun Kang)(Kwangsue Chung)

요약 본 논문에서는 이동 단말에 적응적으로 웹 문서를 제공할 수 있는 효율적인 문서 변환 방법과 적응적 태그 단순화(tag reduction)를 수행하기 위한 RHTML(Reduced HTML)을 제안하였다. 제안한 적응적 문서 변환 방법은 마크업(markup) 오류 제거 과정을 통해 HTML(HyperText Markup Language) 문서를 XML(Extensible Markup Language)의 응용 문서로 변환한다. 이를 통해 웹 문서는 DOM(Document Object Model)을 사용하여 트리 구조로 손쉽게 다루어질 수 있게 되며, 따라서 이동 단말에서 마크업 오류를 수정하는 하드웨어의 부하를 제거할 수 있다. 또한 제안한 문서를 단순화하는 과정에서는 RHTML에 정의된 3가지의 DTD(Document Type Definition)중 하나를 참고하여 이동 단말에 적응적으로 웹 문서를 제공한다.

Abstract In this paper, we proposed an efficient document conversion mechanism to provide an adaptive web document to mobile terminals. We also proposed a RHTML(Reduced HTML) to archive the adaptive tag reduction. Markup error correction process in the proposed adaptive document conversion mechanism converts a HTML(HyperText Markup Language) document into a XML(Extensible Markup Language) application document. This process makes web document easy to handle with a DOM(Document Object Model) as the tree model and removes the hardware overhead in mobile terminals. Also, tag reduction process provides the adaptive web document with three DTD(Document Type Definition)s in the RHTML.

1. 서론

최근 WWW과 이동 통신의 발전으로 이동 단말을 통해 웹에 접근하는 것은 자연스러운 것이 되었다. 그러나 일반 데스크탑 컴퓨터나 워크스테이션을 기준으로 제작된 기존의 웹 문서를 이동 단말에 그대로 제공하는 것은 처리능력이 제한된 이동 단말에서 다양한 문제를 수반한다. 따라서 기존의 웹 문서를 이동 단말 및 무선 망에 적합하도록 하는 방법이 필요하다.

이와 같은 문제점을 해결하기 위해 최근에는 이동 단말에서의 효과적인 웹 접근을 위한 HDML(Handheld Device Markup Language)이나 WML(Wireless Markup Language)같은 전용 마크업(markup) 언어가 개발되

었다. 그러나 기존의 HTML로 제작된 웹 문서들과 호환성이 없으며, 응용들의 수가 매우 부족하다. 따라서 전용의 마크업을 사용하여 문서를 새롭게 작성하는 것보다 기존의 웹 문서들을 그대로 활용하는 방법이 보다 효과적이다. 따라서 이를 위해 기존의 웹 문서를 이동 단말에 적합하게 처리 해줘야할 필요가 있다[1][2].

이를 위한 문서의 처리 방법은 문서의 부호화를 통해 전송할 데이터의 양을 감소시키는 방법, 문서의 태그나 내용의 제거 및 변경을 통해 이동 단말에 적합한 문서를 제공하는 방법이 있다. 문서를 부호화하는 방법은 문서의 내용을 그대로 유지하면서 전송할 데이터의 양을 감소시킬 수 있다는 장점을 가지나 문서를 복호화 할 수 있는 전용 복호화기가 필요하다는 단점이 있다. 반면 문서의 태그나 내용을 제거 및 변경시키는 방법은 기존에 이동 단말에 부가적인 프로그램을 추가하지 않아도 되는 장점을 가지지만 문서가 원본과 달라지게 되는 문제점이 있다. 그러나 이와 같은 문제는 처리할 문서에서 제거 및 변경시킬 내용에 대해 적절한 고려를 통해 원

[†] 학생회원 : 광운대학교 전자공학부

sckang@adams.kwangwoon.ac.kr

^{**} 종신회원 : 광운대학교 전자공학부 교수

kchung@daisy.kwangwoon.ac.kr

논문접수 : 1999년 12월 17일

실사완료 : 2000년 10월 26일

본 문서의 문맥이 파괴되는 것을 최소화시킬 수 있다.

본 논문에서는 문서 내의 태그나 내용을 제거 및 변경시키는 태그 단순화에 기반 하여 적응적으로 웹 문서를 제공하는 문서 변환 방법과 적응적으로 태그를 단순화시키기 위한 3가지의 DTD를 가지는 RHTML (Reduced HTML)을 제안하였다.

본 논문은 총 6장으로 구성되어 있다. 서론에 이어 제 2장에서 마크업 언어의 기본 개념들과 이동 단말에서 웹 접근을 위한 마크업 언어에 관하여 서술하고, 3장에서는 제안한 문서 변환 방법 및 RHTML에 대하여 기술한다. 4장에서는 제안한 문서의 변환 방법을 시험하기 위해 채택한 프록시 서버의 설계 및 구현에 관하여 상세히 기술한다. 그리고 5장에서는 프록시 서버를 통해 실제 이동 단말에 적응적으로 웹 문서가 제공되는지를 시험 및 고찰하고, 마지막으로 6장에서는 결론을 맺는다.

2. 마크업 언어

본 장에서는 마크업의 기본개념을 알아보고 이것을 바탕으로 마크업 문서 표준인 SGML(Standard Generalized Markup Language) 및 XML에 대해 살펴본다. 그리고 이를 바탕으로 SGML의 응용인 HTML과 XML의 응용인 XHTML(Extensible HTML)에 대해서 알아본다. 그리고 최근에 이동 단말에 적합한 문서를 제공해주기 위한 여러 가지 연구에 대해 살펴본다.

2.1 마크업 언어의 기본 개념

2.1.1 마크업의 종류

마크업은 고유 마크업(specific markup)과 범용 마크업(generalized markup)으로 구분된다. 고유 마크업은 특별한 응용이나 장치를 위한 규정된 코드를 생성하기 위해 사용되며, 범용 마크업은 문서의 구조를 기술하나 문서가 어떻게 보일 것인지는 정의하지는 않는다. 따라서 고유 마크업으로 구성된 문서는 이식성이 떨어지는 반면 범용 마크업으로 기술된 문서는 고유 마크업으로 기술된 문서보다 이식성이 있다.

2.1.2 마크업 언어

(가) SGML

SGML은 ISO(International Organization for Standardization) 8879로 플랫폼에 독립적이고 매우 유연한 마크업 스키마(scheme)를 제공하는 메타 언어(meta-language)이다. 즉, SGML은 특정 마크업 언어의 문법을 정의하여, 그 문법을 이용해 문서를 작성할 수 있는 기반을 제공한다[3].

(나) XML

SGML은 강력하며 유연성 있는 메타 언어지만, 그에

따른 복잡성은 SGML을 실제 적용시키는데 많은 어려움을 주었다. 이러한 이유로 XML이 그 해결 방안으로 제안되었다. XML은 SGML의 장점을 그대로 살리면서 거의 사용되지 않는 기능을 없애고 중요한 많은 기능을 그대로 살렸다. 따라서 XML은 SGML의 부분집합(subset) 혹은 프로파일(profile)이라고 말할 수 있다[4].

또한 XML에 대해 이해하려면 잘 구성된(well-formed) XML 문서와 유효한(valid) 문서에 대한 개념을 알아야한다. 잘 구성된 XML 문서는 XML의 규약에 맞으며 정확한 구조를 가지고 있음을 의미하며, 유효한 문서는 XML 문서가 XML 규약을 준수하며 DTD에 순응하는 경우를 의미한다[4].

(다) HTML과 XHTML

HTML은 가장 대중화된 SGML의 응용으로 DTD를 정의하고, 그 정의를 따르는 전용의 웹 브라우저를 사용하여 HTML 파일을 보여준다. 즉, 사용할 수 있는 태그들은 모두 HTML DTD에 정의되어 있어 사용자는 제한적으로 고정된 태그만을 사용해야 했다. 그러나 웹이 폭발 적으로 증가하면서 더 많은 기능과 사용 분야에 대한 요구 사항 및 문제점을 발생시켰다. 이에 대한 대안으로 XHTML이 제안되었다. XHTML은 HTML 4.0의 3가지 DTD를 XML의 응용으로써 재작성 한 것이다. 또한 XHTML은 다른 태그 집합을 조합하고 부분 집합으로 만들기 위한 모듈화를 하였다[5] [9-11].

2.2 이동 단말을 위한 마크업 언어

2.2.1 HDML

Unwired Planet(UP)에 의해 독자적으로 개발되었다. HDML은 기존의 HTML과 달리 제한된 제한을 가진 기기 상에서 데이터 표현 및 사용자 상호작용을 가능하게 한다. HDML 문서는 Deck과 Card라는 작동 개념을 사용하고 있다. 이것은 응용 프로그램이 다중의 Card로 구성된 문서를 나타내도록 한다. 논리적인 하나 혹은 다중의 Card의 집합은 하나의 Deck을 이루며 이 Card의 집합을 통해 웹을 탐색한다[2].

2.2.2 WML

WAP(Wireless Application Protocol) Forum에 의해 정의되었다. WML은 HDML 2.0을 기반으로 만든 XML의 응용으로 소형의 제한적인 처리능력을 가지는 이동 단말에 최적화되어 있고 HDML의 Deck과 Card라는 개념을 그대로 사용한다[1] [7].

3. 제안한 적응적 문서 변환 방법

본 장에서는 2장에서 다룬 마크업의 기본 개념과 이

동 단말기와 무선망의 특징 대한 내용을 바탕으로 이동 단말을 통한 웹 접근의 문제점을 지적하고 문제점을 해결하기 위한 적응적 문서 변환 방법을 제안한다.

기존의 문서 변환 방법은 일반적으로 HTML 문서를 파싱하고 태그 대 태그에 대한 단순한 변환만을 행하거나 문서를 자체적인 데이터 구조로 저장하고 그것을 처리하는 비 표준적이며 복잡한 방법을 사용하게 된다. 그러나 본 논문에서 제안하는 방법은 HTML 문서를 XML 문서로 처리하도록 함으로써 변환 과정을 단순화시킬 수 있으며 HTML 문서에 대해 각 노드에 대해 세밀한 처리를 가능하게 해준다.

본 논문에서 제안한 문서 변환 과정은 마크업 오류 제거 과정, 문서 단순화 과정, 표준 문서 변환 과정의 3가지 과정으로 구성된다. 그림 1은 제안한 문서 변환의 단계를 도시하고 있다.

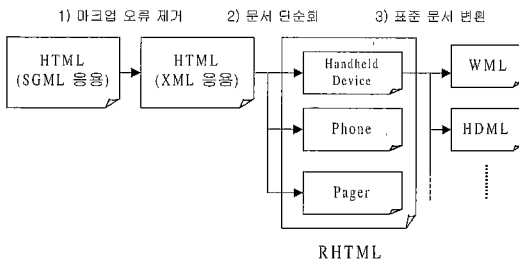


그림 1 제안한 문서 변환 과정

그림 1과 같이 본 논문에서 제안한 문서 변환 과정은 세 가지 과정을 거치게 된다. 첫 번째로 마크업 오류를 포함한 HTML 문서를 잘 구성된 HTML로 수정하는 마크업 오류 제거 과정을 거친다. 두 번째로는 이동 단말에서 지원되지 않거나 제거해도 문서상에 크게 영향을 주지 않는 태그를 삭제하는 문서 단순화 과정을 거친다. 마지막으로 이 문서를 변환하고자 하는 문서로 변환하는 표준 문서 변환 과정을 통해 최종 표준 문서로 변환하게 된다.

3.1 마크업 오류의 제거

HTML로 제작 된 웹 문서는 시작 태그와 해당되는 종료 태그가 항상 존재하며, 올바른 중첩 구조를 가지고 정의된 곳에 올바르게 태그가 나타나야 한다. 그러나 실제 인터넷상의 웹 문서들 다양한 마크업 오류들을 포함하고 있다. 이것은 문서의 명확한 처리를 어렵게 만들며, 문서를 변환하였을 경우에도 계속 마크업 오류들을 포함하게 만든다[12].

또한 대부분의 상용 웹 브라우저들의 마크업 오류 수

정 기능으로 마크업 오류를 포함한 웹 문서도 정상적으로 보여진다. 하지만 브라우저 측의 오류 수정 기능은 처리 능력이 제한적인 이동 단말에게 더 큰 처리 능력을 요구하므로 적합하지 않다. 따라서 이동 단말에 문서를 제공해주기 전에 마크업 오류 제거를 통해 이동 단말 측에 부하를 감소시킬 필요가 있다.

본 논문에서는 이를 해결하기 위해 오류를 포함한 기존의 HTML 문서를 잘 구성된 HTML 문서로 변환 해준다. 이를 통해 문서의 마크업 오류가 수정되며 마크업 오류 수정은 다음의 오류 수정 기준을 따른다.

- 1) 중첩 구조가 올바르게 없을 경우 올바른 중첩구조를 가지도록 한다.
- 2) 요소와 속성 명을 모두 소문자로 변경한다.
- 3) 종료 태그가 존재하지 않는 경우 종료 태그를 삽입한다.
- 4) 속성 값이 인용부호로 묶여 있지 않을 경우 인용부호를 추가한다.
- 5) 속성과 속성 값이 쌍으로 존재 하지 않을 경우 속성과 그 값이 쌍으로 존재하도록 수정한다.
- 6) 빈 요소에는 />를 추가한다.

이 외에도 HTML 문서의 기본 구조를 구성하는 html, head, title, body 요소가 존재하지 않을 경우 문서에 부족한 요소를 삽입시키는 과정을 통해 올바른 HTML 문서가 되도록 해야한다. 표 1은 이 과정을 통해 문서의 마크업 오류가 수정된 예이다[8].

표 1 마크업 오류 수정의 예

수정 전	수정 후
<pre> <TITLE> This is the title </TITLE> <P> Here is a paragraph Bold <I> Bold Italic Bold? </I> </pre>	<pre> <html> <head> <title> This is the title </title> </head> <body> <p> Here is a paragraph Bold <i> Bold Italic </i> Bold? </p> </body> </html> </pre>

마크업 오류 제거 과정으로부터 SGML의 응용인 HTML문서는 XML의 응용 문서가 되며 웹 문서를 DOM(Document Object Model)을 사용하여 트리 구조로 다룰 수 있게 된다. DOM 구조는 HTML 문서를 손쉽게 파싱하고 각 트리 노드에 대해 세밀한 변환을 가능하게 해준다.

3.2 문서의 단순화

문서 단순화 과정에서는 이동 단말에서 지원되지 않는 불필요한 내용을 제거하고 태그를 단순화한다. 스타일 정보는 데스크탑과 같은 환경을 고려하여 제작된 것이며, 스크립트 코드는 제한적인 처리 능력을 가지는 이동 단말 측의 브라우저에서 지원되기 어렵다. 또한 주석은 실제 문서에서 어떤 실제적 기능을 하지 않기 때문에 스타일, 스크립트, 주석 관련 요소 및 정보들은 우선적으로 제거된다[6][12].

태그 단순화는 본 논문에서 제안한 RHTML의 3가지의 DTD를 참조하여 이루어지며, 본 논문에서는 세 가지 DTD의 집합으로 정의된 마크업 언어를 RHTML이라고 정의하였다. RHTML은 HTML 2.0, 3.2, 4.0의 부분집합으로 XHTML과 호환성이 있다. 표 2는 RHTML의 3가지 DTD를 나열하고 각각을 적용적으로 적용할 대상을 기술한 것이다.

하나의 웹 문서는 사용자의 요구 사항이나 무선망의 상황 등의 다양한 조건에 따라 RHTML의 3가지 DTD 집합 중 한 가지를 참고하여 태그가 단순화되고, 이로부터 단일한 웹 문서에 대해 다른 수준의 서비스를 제공할 수 있는 3가지의 문서가 생성된다. 이와 같은 RHTML의 DTD 집합을 정의할 때 레이아웃에 거의 영향을 주지 않으며, 단말기 측에 큰 처리능력을 요구하는 요소 및 속성 제거하였고 문서의 구조나 정보에 관한 요소나 속성은 가능하면 그대로 유지하도록 했다.

표 2 RHTML의 3가지 DTD

DTD	적용할 대상
Handheld Device	흑백 혹은 그 이상의 화면을 가지는 소형의 이동 단말
Phone	매우 작은 화면을 가지며 제한적인 조작 방법을 가지는 이동 전화기
Pager	문자 메시지 서비스를 받을 수 있는 소형의 호출기

레이아웃 구조에 있어서 일반적으로 이동 단말은 제한적인 처리 능력 때문에 흑백의 매우 작은 화면을 가지며 단일한 폰트를 사용하는 경향이 있다. 따라서 컬러 정보나 배경 색 및 이미지 표시 및 폰트에 관련된 요소

및 속성은 제거되거나 단순화되어 질 수 있다. 또한 이동 단말은 처리 능력이 낮기 때문에 브라우저 상에서 부가적인 소프트웨어 구성 요소를 실행하는 것이 어렵다. 따라서 자바 애플릿(java applet)같은 외부 객체를 웹 페이지에 끼워 넣기 위한 요소, 프레임 요소는 제거하였다. 그리고 문서의 구조와 스타일 정보를 분리하여 특정 이동 단말의 레이아웃에 종속되지 않도록 하고, 문서가 단순화 된 후에도 다른 표준 문서로 변환했을 때 변환 된 문서가 원본 문서의 문맥을 그대로 유지하도록 문서의 구조나 내용을 기술하는 요소나 속성은 유지하였다[6].

3.3 표준 문서로의 변환

RHTML은 HTML과 호환성을 가지기 때문에 변환 과정이 없이 그대로 사용 될 수 있으나 WML/HDML 등과 같은 다양한 표준 문서를 지원하는 이동 단말에 웹 문서를 제공하기 위해 문서의 단순화 과정을 거친 웹 문서는 또 다른 표준 문서로 변환되며 RHTML로부터 변환되기 때문에 단일한 웹 문서에 대해 적용적인 3가지의 표준 문서를 생성할 수 있다[12].

4. 프록시의 설계 및 구현

본 장에서는 3장에서는 제안한 문서 변환 방법을 시험하기 위해 프록시(proxy) 구조를 채택하고 프록시 서버의 설계 및 구현에 관하여 상세히 기술한다.

문서의 변환을 행하는 게이트웨이(gateway)는 인터넷(Internet)/인트라넷(Intranet)망과 무선망 사이에 존재할 수 있으며 일반적인 구조는 그림 2와 같다. 본 논문에서는 문서 변환을 위해 게이트웨이로써 프록시 구조를 채택하였으며, 본 논문에서 설계 및 구현한 프록시 서버는 단일한 서버가 다수의 이동 단말 측의 브라우저에게 서비스를 해주는 HTTP(HyperText Transfer Protocol) 프록시와 매우 유사한 구조를 가진다.

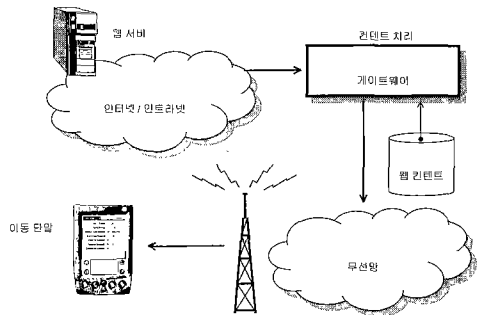


그림 2 문서 처리를 위한 게이트웨이 구조

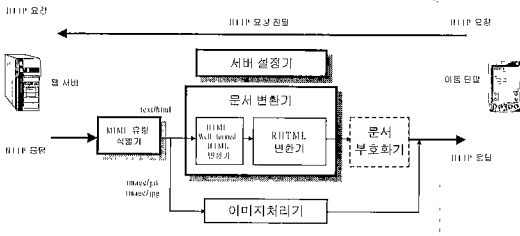


그림 3 프록시 구조

일반적으로 무선망과 인터넷/인트라넷 망은 클라이언트/서버 구조를 가지고 있다. 이와 같은 구조는 클라이언트 측의 HTTP 요청이 직접 웹 서버로 전달되고 웹 서버는 해당하는 문서를 클라이언트에 제공해준다. 예를 들어 'http://adams.kwangwoon.ac.kr/'라는 URL에 요청을 하기 위해 클라이언트는 'GET /index.htm HTTP/1.0'라는 요청과 부가적인 헤더 정보를 'adams.kwangwoon.ac.kr'에 보낸다.

이와 같은 서버/클라이언트 구조 사이에 프록시 서버를 위치 시켜 원하는 문서 처리를 가능하게 할 수 있다. 이 구조는 클라이언트가 서버에게 요청을 직접 하는 것이 아니라 프록시를 통하게 된다. 즉, 앞의 예에서 클라이언트에서 서버로의 HTTP 요청은 프록시 서버로 전달되고 다시 그 요청은 'GET http://adams.kwangwoon.ac.kr/index.htm HTTP/1.0'의 형태로 해당하는 웹 서버에 전달된다. 웹 서버는 요구에 대해 응답하여 원하는 자원을 프록시에게 전달해 주고 프록시 서버는 그 자원에 대해 해당하는 처리를 한 후 클라이언트에게 전달해 준다[13] [14].

설계한 프록시 서버 역시 앞에서 설명한 일반적인 프록시 서버와 유사하게 동작하며, 자원의 형태를 판별하는 MIME 유형 식별기(MIME type identifier)와 문서 변환기(document converter), 이미지 처리기(image processor), 서버 설정기(server configuration)로 구성되어 있다. 그림 3은 프록시 서버의 전체 구조를 나타낸 것이다.

4.1 MIME 유형 식별기

프록시를 통해 해당하는 웹 서버에 웹 콘텐츠를 요청하게 되면 웹 서버는 그에 대한 응답으로 상태 코드, 문서에 대한 헤더 정보, 요청 된 데이터를 프록시에 보내게 된다. 프록시는 그 정보로부터 MIME 유형을 판별하여 HTML 문서는 문서 변환기로 전달하고 이미지라면 이미지 처리기로 전달해 주게 된다.

4.2 문서 처리기

문서 처리기는 HTML 문서를 SGML의 응용에서 XML의 응용으로 변환 해주는 HTML-잘 구성된 HTML 변환기(HTML to Well-formed HTML converter)와 태그를 단순화를 행하는 RHTML 변환기(RHTML converter)로 구성된다. HTML-잘 구성된 HTML 변환기는 통해 문서의 마크업 오류를 제거하며 문서를 DOM을 사용하여 다룰 수 있게 한다.

RHTML 변환기는 다시 문서 처리기(document handler)와 태그 처리기(tag handler)로 구성된다. 문서 처리기는 잘 구성된 HTML 문서를 트리로 파싱하고 해당하는 각 트리 노드를 만나게 되면 그에 해당하는 태그 처리기를 활성화(instance)시킨다. 각각의 태그 처리기는 해당하는 태그에 대해 요소와 속성 정보를 얻어 RHTML의 3가지 DTD를 참조하여 RHTML 문서를 생성한다. 또한 태그 처리기는 시작 태그 외에 종료 태그에 대해서도 어떠한 작업을 해야하는 지를 지정할 수 있다. 그림 4는 이와 같은 작업을 행하는 RHTML 변환기의 구조를 도시한 것이다. 본 논문에서는 RHTML로 단순화 된 HTML을 HTML 3.2 로 다시 변환하였다. 하지만 태그 처리기를 확장하는 방법을 통해 다양한 마크업 변환도 가능하다.

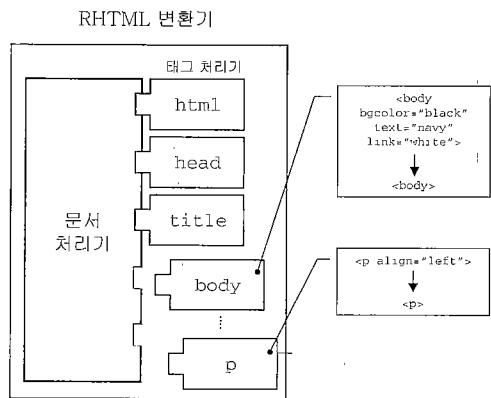


그림 4 RHTML 변환기의 구조

4.3 서버 설정기

서버 설정기는 각 표준 마크업 응용의 DTD를 정의하며, 현재 서버의 작업에 대한 상태 값을 저장하는 상태 변수를 정의한다. 프록시의 각 처리 모듈들은 서버 설정기로부터 값을 참조하여 작업을 수행한다.

4.4 이미지 처리기

문서 데이터 외에 부가적인 이미지를 압축 및 변형하

는 처리한다. 즉, 이미지의 경우도 웹 문서처럼 적응적으로 처리하여 이동 단말에 제공 할 수 있다.

5. 시험 및 결과

본 장에서는 제안한 문서 변환 방법을 사용하여 구현한 프록시의 구현 환경 및 동작을 기술하고 기존의 웹 문서를 직접 이동 단말에 제공하는 것과 프록시를 통해 이동 단말에 웹 문서를 제공하는 두 가지 경우를 다음의 3가지 관점에서 비교·분석 하여본다. 첫째로 문서가 태그 및 내용의 손실을 가지고도 최종 적으로 변환 된 HTML 문서가 원본의 레이아웃을 잘 유지하는지를 살펴본다. 둘째로는 이동 단말로 최종적으로 HTML로 변환 되어 전송된 실제 문서의 양이 감소하는지를 살펴보고 마지막으로 전송된 문서의 처리 시간이 감소하는지를 살펴본다.

5.1 시험 및 구현환경

사용된 운영체제는 Microsoft Windows NT 4.0 Server를 사용하였으며, 프록시 서버와 문서 데이터 크기와 처리시간을 측정하기 위한 클라이언트는 Sun의 JDK(Java Development Kit) 1.2.2를 사용하여 순수 자바 언어로 구현하였고, Run-time 환경으로는 Java HotSpot Performance Engine 1.0.1을 사용하였다. 이동 단말기 에뮬레이션을 위해 3Com의 Palm OS 3.0을 지원하는 Palm OS Emulator 2.1d29를 사용하였으며, 에뮬레이터 상에서 전송 된 웹 문서를 보기 위한 웹 브라우저로는 PalmScape PR4를 사용하였다. 그리고 100Mbps Ethernet LAN으로 연결된 64MB의 메모리를 가지는 Pentium II 300MHz PC 2대를 이용하였다.

5.2 시험 결과

프록시 서버를 시험하기 위해 검색엔진으로 가장 많이 이용되고 있는 Yahoo와 복잡한 레이아웃 구조를 가지는 잡지 사이트로서 Byte 웹사이트를 시험 웹사이트로 선정하였다. 이 외에 Lycos(<http://www.lycos.com>), Wall Street Journal(<http://interactive.wsj.com>), WAP forum(<http://www.wapforum.org>)의 사이트들에 대해서 데이의 크기와 처리 시간에 대한 시험을 행하였다.

5.2.1 문서의 레이아웃

프록시를 통하지 않은 웹사이트의 레이아웃과 프록시를 통해 태그와 내용의 손실이 생긴 웹사이트의 레이아웃을 비교하였다. 그림 5는 프록시를 통한 웹사이트를 Palm OS Emulator의 웹 브라우저를 통해 본 모습이다.

그림 5처럼 프록시를 통해 접근한 웹사이트와 프록시를 통하지 않고 접근한 웹사이트의 레이아웃이 크게 다

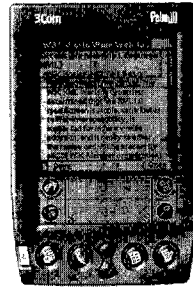


그림 5 Palm OS Emulator 상의 웹 브라우저의 실제 화면

르지 않아야 한다. 그림 6은 프록시를 통하지 않고 웹사이트에 접근한 화면이고, 그림 7은 프록시를 통해 웹사이트에 접근했을 경우 적응적으로 웹 문서가 제공된 모습이다. 각각은 왼쪽으로부터 HandheldDevice, Phone, Pager DTD를 참조해 적응적으로 HTML로 변환된 웹 문서들이 이동 단말에 제공된 화면을 나열한 것이다.

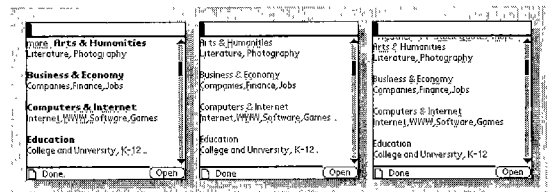


(a) Yahoo

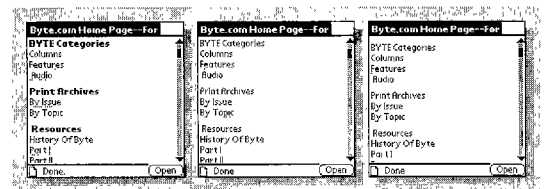


(b) Byte

그림 6 프록시 없이 접근한 시험 웹사이트의 화면



(a) Yahoo (<http://www.yahoo.com>)



(b) Byte (<http://www.byte.com>)

그림 7 프록시를 통한 시험 웹사이트의 화면

그림 7에서 보는 바와 같이 이동 단말에 각각의 적응적으로 제공된 웹 문서들의 레이아웃은 거의 변형되지 않았으며 원본 웹 문서들과 유사하다. 하지만 이것만으로는 웹 문서가 실제로 이동 단말에 적응적으로 제공되었는지 판단하기 어렵기 때문에 문서의 데이터 양과 처리시간이 감소하였는지를 검증할 필요가 있다.

5.2.2 문서 데이터 크기 및 문서의 처리 시간

5.2.1절에서와 같이 문서 내부의 태그와 내용 일부의 손실이 생겼음에도 웹 문서의 레이아웃이 크게 파괴되지 않았다. 하지만 문서의 단순화를 행한 목적은 낮은 처리 능력을 가진 이동 단말에서 문서를 손쉽게 처리해 주기 위함과 제한적인 무선망에 부담을 줄이기 위한 것이다. 따라서 실제 문서 자체의 데이터 크기도 감소해야 한다.

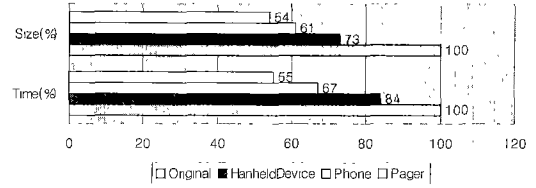
표 3은 실제 적응적으로 변환 된 문서들의 데이터 양을 측정한 것이다. 표 3에서 Yahoo의 경우 원본 문서의 데이터의 크기는 약 12202byte이지만 프록시를 거쳐 적응적으로 제공된 문서는 Pager DTD의 경우 6711byte로 약 46% 정도의 문서 데이터 크기가 감소하였다. 또한 문서 처리 시간에 있어서도 처리시간이 721ms에서 55ms로 약 45% 정도로 감소하였다.

Byte 사이트에 대한 시험 결과 문서 데이터의 크기가 27082byte인 원본 문서가 프록시를 통해 적응적으로 제공되었을 때 Pager DTD의 경우 문서 데이터의 크기가 7739byte로 약 71% 정도가 감소되는 결과를 나타냈다. 또한 Byte의 경우 가장 웹 문서 내의 태그 및 내용의 손실이 적은 HandheldDevice DTD를 통해 적응적으로 문서를 제공하더라도 약 50%정도의 큰 데이터 감소를 보였다. 또한 문서 처리 시간에 있어서도 원본 문서의 경우 822ms에서 Pager DTD의 경우 340ms까지 처리시간이 줄어들어 약 59% 정도의 처리 시간이 감소되는 것을 알 수 있다.

특히 Byte 사이트에 대한 시험 결과의 경우 문서 자체가 매우 복잡한 레이아웃 구조를 가지기 때문에 다양한 태그가 사용되고 있으며 이와 같은 태그들이 실제 문서 내부의 많은 부분을 차지하고 있다. 따라서 태그 단순화를 하게 되면 급격하게 문서의 데이터 크기가 감소되게 된다. 그러나 복잡한 레이아웃을 가지지 않는 사이트 일 경우 태그 단순화의 효과가 크게 나타나지 않는다. 이에 대한 결과는 그다지 복잡한 레이아웃을 가지지 않는 Lycos, Wall Street Journal과 WAP forum 사이트에 대한 시험 결과에서도 명확한 차이를 나타낸다. 이와 같은 경우 태그 단순화의 장점은 문서를 처리하는 시간이 감소된다는 데 있다.

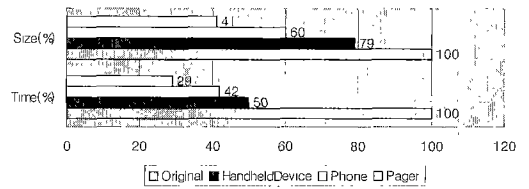
표 3 문서 데이터 크기 감소와 처리 시간의 감소

	Original	HandheldDevice	Phone	Pager
Size (byte / %)	12202 / 100	8960 / 73	7561 / 61	6711 / 54
Time (ms / %)	721 / 100	611 / 84	490 / 67	400 / 55



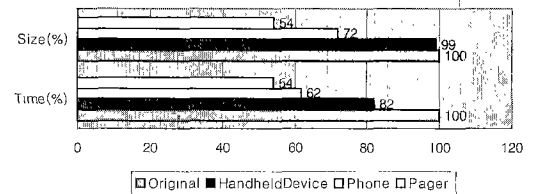
(a) Yahoo (<http://www.yahoo.com>)

	Original	HandheldDevice	Phone	Pager
Size (byte / %)	27082 / 100	13512 / 50	11254 / 42	7739 / 29
Time (ms / %)	822 / 100	651 / 79	491 / 60	340 / 41



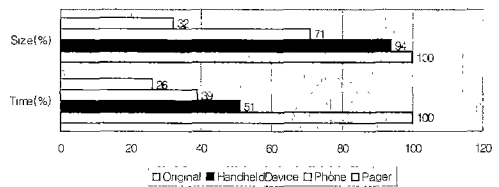
(b) Byte (<http://www.byte.com>)

	Original	HandheldDevice	Phone	Pager
Size (byte / %)	15907 / 100	13116 / 82	9801 / 62	8621 / 54
Time (ms / %)	811 / 100	801 / 99	581 / 72	441 / 54



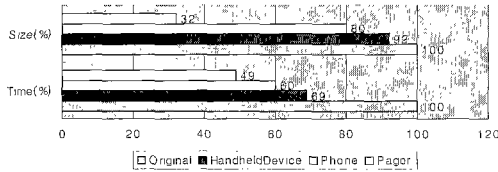
(c) Lycos (<http://www.lycos.com>)

	Original	HandheldDevice	Phone	Pager
Size (byte / %)	9296 / 100	4751 / 51	3617 / 39	2442 / 26
Time (ms / %)	341 / 100	320 / 94	241 / 71	110 / 32



(d) Wall Street Journal (<http://interactive.wsj.com>)

	Original	HandheldDevice	Phone	Pager
Size (byte / %)	15169 / 100	10479 / 69	9135 / 60	7478 / 49
Time (ms / %)	411 / 100	380 / 92	330 / 80	130 / 32

(e) WAP Forum (<http://www.wapforum.org>)

6. 결 론

본 논문에서는 이동 단말에 적응적인 웹 문서를 제공하기 위한 효율적인 문서 변환 방법과 문서 단순화를 위한 RHTML을 제안하였다. 제안한 문서 변환 방법을 통해 원본 문서의 마크업 오류를 제거하고 RHTML의 3가지 DTD의 집합으로부터 적응적으로 경량의 웹 문서를 제공하였다. 본 논문에서는 구현한 프록시 서버를 통해 실제로 웹 문서가 원본과 거의 유사한 레이아웃을 유지하면서도 문서의 데이터 양이 감소하고 문서를 처리하는 시간이 감소하는 것을 확인하였다.

향후 연구과제로는 실제의 웹사이트에 사용되고 있는 태그 및 내용을 분석하여 좀 더 체계적으로 태그 집합을 정의하고 이를 통해 정량적인 웹 문서의 단순화를 행하는 방법에 관한 연구가 필요하다. 또한 문서를 적응적으로 변환하는 데 있어서 지능형 에이전트의 개념을 도입하여 무선망의 상황이나 이동 단말의 요청 등을 분석하여 지능적으로 적응화 된 문서를 제공해 주는 방법에 대한 연구가 필요하다.

참 고 문 헌

- [1] WAP Forum, Wireless Markup Language (WML) 1.1 Specification, Jun. 1999.
- [2] Unwired Planet, Handheld Device Markup Language (HDML) 2.0 Specification, Apr. 1997.
- [3] ISO 8879:1986, Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML), 1986.
- [4] W3C Recommendation, Extensible Markup Language (XML) 1.0 Specification, Feb. 1998.
- [5] W3C Proposed Recommendation, XHTML 1.0 : The Extensible HyperText Markup Language - A Reformulation of HTML 4.0 in XML 1.0, Aug. 1999.
- [6] W3C Note, HTML 4.0 Guidelines for Mobile Access, Mar. 1999.
- [7] WAP Forum, <http://www.wapforum.com>

- [8] W3C Recommendation, Document Object Model (DOM) Level 1 Specification Version 1.0, Oct. 1998.
- [9] ISO/IEC 15445 : 1998 HyperText Markup Language (HTML), 1998.
- [10] W3C Recommendation, HTML 3.2 Reference Specification, Jan. 1997.
- [11] W3C Recommendation, HTML 4.01 Specification, Aug. 1999.
- [12] 강성천, 정광수, "이동 통신 환경에서 효율적인 웹 문서 변환에 관한 연구", 한국정보과학회 추계 학술발표논문집 Vol. 26, No. 2 pp. 667~669, 1999.
- [13] 함경선, 정성신, 양서민, 이혁준, "CDMA 무선망에서의 WWW 응답속도 향상을 위한 무선망 적응 프록시의 구현", SK Telecom Telecommunications Review Vol. 9, No. 3 pp. 358~368, 1999.
- [14] A. Fox, S. D. Gribble, Y. Chawathe, E. A. Brewer, "Adapting to Network and Client Variation Using Infrastructural Proxies: Lesson and Perspectives," IEEE Personal Communications, 5(4): 10~19, Aug. 1998.



강 성 천

1998년 2월 광운대학교 전자통신공학과 졸업(공학사). 2000년 2월 광운대학교 대학원 전자통신공학과 졸업(공학석사).



정 광 수

1981년 한양대학교 전자공학과 학사. 1983년 한국과학기술원 전기 및 전자공학과 석사. 1991년 미국 University of Florida 전기공학과 박사(컴퓨터공학전공). 1983년 ~ 1993년 한국전자통신연구원 선임연구원. 1991년 ~ 1992년 한국과학기술원 대우교수. 1993년 ~ 현재 광운대학교 전자공학부 부교수(정보통신연구원 연구원). 관심분야는 컴퓨터통신, 인터넷, 분산처리, 멀티미디어