모바일 환경을 위한 웹 컨텐츠 추출기법 설계 및 구현

박지선 · 김창수† · 송하주

(† 부경대학교)

Design and Implementation of a Filtering Technique of Web Contents for the Mobile Environment

Ji-sun PARK · Chang-soo KIM† · Ha-joo SONG

† Pukyong Natioal Univeristy

(Received January 13, 2005 / Accepted February 22, 2005)

Abstract

Mobile devices compared with personal computers on the desktop have low bandwidths, small screens, and relatively slow speeds. These systems have practical problems with information searches in a web environment. Information searching of various web contents on the small size screen has especially severe limitations.We propose a filtering technique of web contents which can overcome the limitations of small size screens and meet the user requirements in a PDA environment. For these constructions, we first divide the screen into segment blocks and then extract content blocks according to the user requirements, so that only filtered web contents will be shown. The performance evaluation of the proposed technique saves an average time of about 30% by displaying only the extracted information instead of the whole web page.

***Key Words**: Web contents, Mobile environment, Filtering technique*

I. 서 론

정보화 시대를 맞이하여 인터넷은 이제 사회 전반에서 보편적인 정보 취득 수단이 되었다. 언제 어디서든 웹을 사용하고자 하는 사람들의 욕구는 무선 인터넷 환경을 창출하였고, 1990년대 중반 이후 소형 단말기 시장의 성장은 무선 인터넷의 접속을 증가시켰다.

무선 단말기의 보급이 일반화되고 모바일 환경에서 인터넷을 이용할 기회가 많아짐에 따라 웹 페이지의 개인화 연구가 활발히 진행되고 있다. 개인화는 맞춤 정보 제공이라는 측면에서 인터넷상에 과도하게 많은 정보를 요약하는 특성을 가지고 있다. 기존 논문에서는 검색엔진과 같은 텍스트 위주의 서비스들의 개인화 기법을 많이 다루지만 사용자들은 더 이상 텍스트 위주의 서비스에 만족하지 않으며 다양한 멀티미디어 서비스를 제공받기를 원한다. 그러나 현재 모바일 단말기를 이용한 인터넷 서비스 이용은 단말기의 인터페이스 및 성능 때문에 상당히 제한적이다. 인터넷상에는 무수히 많은 정보와 서비스들이 제공되고 있지만, 대부분 데스크 톱 사용자들을 위해 최적화되어 있다.

† Corresponding author : 051-620-6394, cskim@pknu.ac.kr

최근 출시되고 있는 모바일 단말기들은 기존 제품보다 더욱 많은 메모리와 강력한 프로세서를 탑재하고 있다. 하지만 제한된 화면 크기와 입력 장치는 무선 단말기의 가장 큰 장점인 휴대성과 상충관계에 있기 때문에 쉽게 개선되지 못하고 있다. 이런 단점을 보완하여 제한된 무선 환경 하에서 웹 컨텐츠를 제공하기 위해 많은 연구들이 진행되어 왔다. 그러나 컨텐츠 변환 과정에서 과도한 부하가 발생하거나 변환이 수동으로 이루어지는 등 크고 작은 단점들을 안고 있다.

모바일 단말기는 데스크 탑에 비해 상대적으로 낮은 대역폭, 작은 화면, 느린 중앙처리장치(CPU)를 가지고 있기 때문에 원활한 웹 검색이 어렵다. 특히 모바일 단말기의 작은 화면은 데스크톱에 최적화 된 웹 페이지를 그대로 보기에는 상당한 제약이 따른다. 이러한 제약들을 극복하고자 Power Browser[1], Digestor[2]등 웹 페이지의 정보를 요약하고 재구성해서 무선 단말기에 표시하고자 하는 연구들이 활발히 진행되어왔다. 그러나 이러한 연구들은 웹 컨텐츠나 배치의 요약 및 재배치에만 주안점을 두고 있으며, 모바일 사용자의 서비스 위주 웹 검색 패턴을 반영하지 못하고 있다.

본 논문에서는 이러한 단점들을 극복하고 개인화의 장점을 얻고자 원래 웹 페이지에서 제공되는 서비스들을 사용자에 따른 맞춤형 서비스로 제공하고자 한다. 즉 사용자가 접속했던 사이트를 리스트 형태로 보여줌으로써 재 접속시 번거로움을 줄였다.

또한 프락시 기반의 필터링 모듈구현을 통해 특정 리스트 선택시 전체 웹페이지 중 쿼리한 contents block만 추출하여 모바일 기기에 디스플레이 함으로써 변환 시간을 줄일 수 있도록 하였다.

본 논문의 구성은 다음과 같다. 제2장에서는 관련 연구로 웹 페이지의 개인화 연구와 무선 단말기를 위한 웹 컨텐츠 변환 시스템, 프락시 기반의 웹 문서 변환에 대해 알아본다. 3장에서는 제안하는 시스템의 전체 구성과 각 모듈의 기능에 대해 알아보고 시스템 처리과정을 기술한다. 또한 전체 시스템 구성 모듈중 필터링 모듈에 대해 자세히 알아보며 흐름도와 알고리즘을 통해 처리과정을 보인다. 4장에서는 실제 구현을 통해 처리 결과를 보이고 마지막으로 5장에서 결론을 제시하였다.

Ⅱ. 관련 연구

1. 웹 페이지의 개인화 연구

최근 들어 모바일 단말기의 하드웨어적 단점을 극복하고 동시에 모바일 통신의 특징인 개인화(personalization)를 살려 웹 페이지를 사용자에 맞게 제공하기 위한 연구들이 진행되고 있다. 웹 사이트의 개인화는 온라인 쇼핑몰 등에서 사용자의 구매 데이터를 기반으로 상품을 추천하는 동적인 웹 페이지 생성 등에 사용되고 있으며 그 중요성이 계속 증가되고 있다. 개인화는 맞춤 정보 제공이라는 측면에서 인터넷상에 과도하게 많은 정보를 요약하는 특성을 가지고 있다. 이러한 웹 사이트 개인화를 화면 크기나 대역폭이 데스크탑에 비해 상대적으로 열악한 모바일 사용자에게 제공하여 개인화와 정보요약의 두 가지 효과를 얻고자 하는 연구가 진행 되어왔다.

모바일 사용자를 위한 개인화에 대한 연구로는 내용기반 기계 학습 알고리즘(content-based machine learning algorithm)[6]을 이용하여 사용자의 뉴스 관심도를 학습한 뒤 열악한 무선환경(대역폭, 화면크기)에 적합하게 정보의 양을 축소시켜 제공해 주는 연구가 있다.[7] 또한 웹 로그 마이닝을 통해 얻을 수 있는 사용자의 접근 패턴, 선호도 등을 에이전트의 학습에 활용하는 방법이 있다. 많은 웹 사이트 개인화 관련 연구들이 위 방법을 사용하고 있다. 그러나 유선 인터넷을 이용 하는 사용자의 패턴을 분석하여 무선 인터넷 환경을 고려하지 않고 개인화에 적용하는 것

은 개인화 된 정보의 정확성이 낮은 문제점이 있다. 지금까지의 웹 컨텐츠 및 서비스의 개인화 연구들은 주로 로그 데이터 분석에 초점을 두고 있다.

모바일 단말기를 통해 웹에 접근하는 사용자들은 단순히 웹을 검색하는 것이 아니라 특정한 정보나 서비스를 찾거나 받으려는 경향을 가진다. 또한 검색 폼, 미디어 재생기, 뉴스 섹션 등의 서비스 요소들은 대부분 전체 서비스 페이지 중 일부만을 점유하고 있다. 그러므로 사용자가 특정 서비스를 사용하고자 할 때 전체 웹 페이지 중 서비스 컨텐츠가 위치한 특정 부분만이 요구된다. 본 논문에서는 사용자가 사용한 웹 페이지를 리스트로 제공하고, 선택시 해당 웹 페이지의 컨텐츠를 포함하는 segment block 리스트를 보여준다. 이중 특정 부분만 선택하여 사용자에게 서비스 하고자 한다. 사용자는 재 접속 시 번거로움을 줄일 수 있으며 원하는 데이터에 보다 쉽게 접근할 수 있는 장점을 가진다.

2. 무선 단말기를 위한 웹 컨텐츠 변환 시스템

모바일 기기는 화면 크기나 대역폭이 데스크탑에 비해 상대적으로 열악하다. 그러므로 사용자에게 제공하기 위해 대부분의 개인화 시스템은 무선 단말기를 위한 웹 컨텐츠 변환 시스템을 포함하고 있다. 하지만 지금까지의 해결책에는 여러 가지 문제가 대두된다.

첫째, 웹 상에 존재하는 웹 페이지를 배재한 채 무선 단말기만을 위한 새로운 페이지의 재생성은 많은 부담을 초래한다. 이와 같은 방법의 하나로 WML(Wireless Markup Language)[5]이 등장하였지만 이는 데스크 탑 컴퓨터와 무선 단말기의 작은 스크린에 맞는 페이지를 각각 별도로 준비해야 하는 부담과 함께 WWW를 이원화시키는 위험을 가진다. 그러므로 현재 존재하는 웹 페이지를 이용하는 방법이 요구된다. 본 논문에서는 HTML을 기반으로 하기 때문에 컨텐츠 개발이 용이하다는 장점을 가진다.

둘째, 현재의 웹 페이지를 사용하지만 웹 페이지의 구조를 무선 단말기에 맞게 변형하는 방법으로서 다음과 같은 것들이 있다. Digestor[2]는 구조적 페이지 변형(structural page transformation)과 문장 탈락(sentence elision)을 이용해서 점증적인 요약을 제공한다. 본 논문에서는 Digestor[2]에서 제시한 다양한 휴리스틱 변환 기법들의 조합 적용 과정보다 간단한 알고리즘을 적용함으로써 변환 시스템의 변환 시간을 줄일 수 있도록 하였다. [3]의 논문에서는 시각적 분리 표현 근거를 기반으로 한 연구를 소개하고 있다. 하지만 내용 블록 단위의 추출을 목적으로 하는 본 논문과는 달리 시각적 유사성을 이용한 패턴 추출 기법을 제안하는 차이점이 있다.

3. 프락시 기반의 웹 문서 변환

웹 문서가 변환되는 단계 또는 시점을 분류 기준으로 하여 크게 세 가지로 나누어 보면 다음과 같다. 웹 서비스 구조를 프락시가 존재하는 3-Layer로 구성했을 때 서버-프락시-클라이언트를 변환 수행단계 또는 시점으로 하여 분류한 것이다.

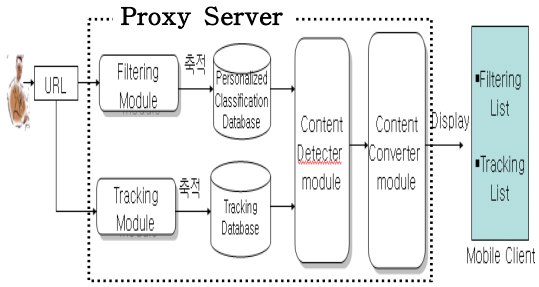
첫째, 서버 측에서 웹 문서의 변환이 이루어지는 경우이다. 이는 여러 단말을 지원하기 위해 별도로 제작된 문서를 웹 서버가 가지고 있거나, 또는 별도의 표현 기법에 대한 정의를 단말별로 이미 가지고 있다. 웹 문서의 실시간 자동 변환 시스템을 사용하여 소형 단말을 지원 할 수도 있지만, 대부분의 웹 서버 운영자는 수작업을 동반한 변환 메카니즘[8]을 이용하고, 따라서 가장 정확한 변환이 이루어지는 장점을 가진다. 하지만 서버 측에서의 변환은 제한된 웹 정보에 한해서 이루어지고, 기존의 유선 상에 존재하는 웹 정보의 방대한 양에 비하여 변환 서비스가 제공되는 문서의 범위가 너무 제한적인 단점이 있다. 둘째,

클라이언트 측에서 웹 문서의 변환이 이루어지는 경우이다. 웹 문서를 전송 받은 클라이언트 측에서 적절한 변환을 수행하는 것으로 사용자의 요구사항을 반영한 변환과 개인 특성화가 쉽게 구현될 수 있는 장점이 있다. 하지만 유선과 동일한 형태의 정보를 클라이언트가 받은 후에 변환 과정을 단말에서 수행함으로써 네트워 자원의 비효율적인 사용과 클라이언트 단말의 높은 컴퓨팅 파워를 요구하게 된다. 셋째, 프락시 측에서 웹 문서의 변환이 이루어지는 경우이다. 대부분의 자동변환 시스템은 프락시로 동작하면서 다양한 단말들을 지원하기 위한 변환을 시도한다. 이 기법은 반드시 프락시 서버를 거쳐야 한다는 조건을 가지나 이미 유선 네트웍 환경에서 대부분의 인트라넷 등이 컨텐츠의 재활용, 보안, 네트웍망의 효율적인 관리 등을 목표로 프락시 서버를 사용하고 있다. 이상으로 기존의 관련연구들은 크게 세 가지로 분류하여 살펴보았는데, 물론 모든 기법들이 장단점을 가지므로 특정 환경, 특정 서비스에 대해서 좀 더 적합한 변환 기법을 선별할 수 있을 것이다. 본 논문에서는 보다 많은 웹 정보를 모바일 브라우저에게 적합한 표현 형식으로 자동으로 변환하여 제공하고, 자원의 효율적인 사용을 위하여 프락시 기반의 변환 기법을 사용하고자 한다. 따라서 프락시의 기본기능에 웹 문서 변환을 위한 필터링 모듈을 추가하는 방법을 사용하여 프락시 기반의 웹 문서 변환 시스템을 설계하고 구현하였다.

Ⅲ. 프록시 서버의 설계

1. 시스템 구조

현재 많은 웹사이트들은 다양한 서비스들을 선보이고 있다. 하지만 검색 폼, 미디어 재생기, 뉴스 섹션 등의 서비스 요소들은 대부분 전체 서비스 페이지 중에서 일부만을 점유하고 있다. 그러므로 사용자가 특정 서비스를 사용하고자 할 때 전체 서비스 페이지 중에서 서비스 요소들이 위치한 특정 부분만이 요구된다. 예를 들면 우리가 웹 검색을 위해 야후나 구글 등의 검색 페이지에 접근한다면, 우리는 단지 검색 키워드 입력을 위한 폼만이 필요하다. 본 논문에서는 이러한 특정 서비스를 한번의 접근 후에는 사용자에게 맞춤형 리스트로 제공되어 보다 접근이 편리하도록 설계한 시스템을 제안한다. 또한 모바일 환경이라는 특성을 고려해서 웹 페이지의 변환시간을 줄이고자 한다. 특히 요즘 폭발적으로 증가하는 멀티미디어 서비스는 데이터 전송량이 많은 특징이 있다. 전체 웹 페이지를 읽어오는 것보다 이러한 멀티미디어 컨텐츠를 포함하는 contents block만을 가져와 모바일 기기에서 보여줌으로 변환 시간을 줄일 수 있다. 본 논문에서는 이러한 변환 기능을 수행하는 필터링 모듈에 대해 상세히 기술하고자 한다. 시스템의 전체적인 구성은 [그림 1]과 같다.



[그림 1] 시스템의 구성.

2. 프록시 서버의 동작방식

사용자가 사용하고자 하는 URL을 입력한 후에는 두 개의 모듈이 동작하게 된다. Filtering module과 tracking module이다. Filtering module은 사용하고자 하는 서비스를 포함한 웹 페이지를 segment block으로 나누어 personalized classification DB에 저장 후 쿼리한 contents block만이 서비스 되도록 한다. Tracking module은 주기적인 tracking을 통해 개인화 된 웹 페이지 등을 추적하여 관련 페이지 등을 tracking DB

에 저장함으로써 사용자에게 추천 페이지로 제공될 수 있다. 즉 사용자가 기존에 사용하던 정보 이외에 참고 정보들도 리스트화 되어 제공되는 것이다. 이렇게 데이터베이스에 축적된 데이터들은 content detecter module을 거쳐 사용자가 접속했던 서비스 페이지에 대한 URL정보를 얻을 수 있다. 모바일 단말기에 디스플레이 하기 위해 컨텐츠 변환모듈인 content convert module로 보내지게 된다. Content convert module에서는 contents block만 리스트로 제공되게 된다. 이 리스트들이 화면에 디스플레이 되는 방법도 좀더 효과적으로 나타낼 수 있다. 위 부분은 filtering 정보 리스트를 나타냄으로써 사용자가 사용하고자 하는 고유정보를 나타내었고, 아래 부분은 tracking 정보 리스트로써 관계정보 즉 추천정보를 제공함으로써 사용자에게 보다 폭넓은 정보를 제공하고자 한다.

3. 필터링 모듈

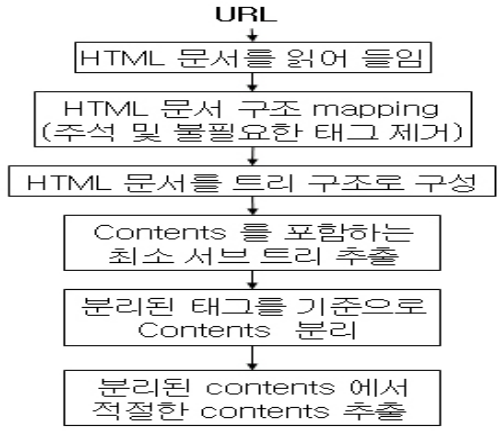
가. 웹 페이지의 시각적인 분리

오늘날 웹 저작자는 다양한 시각적 효과를 사용하여 가능한 많은 정보를 한 페이지에 표현하고자 한다. 이 정보는 시각적으로 단체화하여 조각나 있다. 대부분의 웹사이트가 <TABLE> 요소를 사용하여 전체적인 웹 문서의 구조를 형성하고, 색상 값, 이미지, 공백 표현 등을 사용하여 컨텐츠 내용의 논리적인 구분을 시각적으로 표현하고 있다. 즉 전체 사이트를 시각적인 구분에 따라 나눈다면, 그것이 내용에 근거한 상이한 조각으로 분리가 되는 것이다. 여기서 코드에 따라 나누어지는 각 동질의 내용 부분을 segment block이라 한다.

나. HTML 태그에 의한 segment block추출

Segment block은 HTML문서 등을 작성할 때 나타나는 몇 가지의 일반적인 규칙들을 이용한다. 컨텐츠를 분리하기 위해 경계가 되는 태그를 추출하여 추출된 태그를 기준으로 분리할 수 있다. 즉 segment block이란 코드 <TABLE>에 따라 나누어지는 동질의 부분이며 이렇게 분리된 segment block들 중에서 쿼리한 contents block을 추출해 낸다.

Contents block을 추출하는 전체 흐름도는 그림2와 같다.



[그림 2] Contents block을 추출하는 흐름도.

읽어들인 HTML 문서는 주석과 같은 불필요한 태그를 제거한 다음 [그림 3-a]와 같은 알고리즘을 이용해 트리 구조로 구성한다. 본 논문에서 HTML 문서를 트리 구조로 구성하는 이유는 트리 구조가 HTML 문서 구조를 분석하기가 좀 더 용이하기 때문이다.

다. Contents block 추출 알고리즘

[그림 3-a]는 HTML문서를 트리 구조로 구성하는 알고리즘이고, [그림 3-b]는 contents를 포함하는 최소 서브 트리 추출하는 알고리즘이다.

그림[3-a]는 단어를 토큰화하여 tag, contents로 구분한 후에 <TABLE>,<TR>, <TD> 태그를 노드로 가지는 트리 데이터 구조로 표현한다. 웹 페이지를 파싱(parsing)하여 트리(Tree) 구조로 표현하면 트리 구조의 각 태그의 깊이(Level)를 알 수 있다. 이 페이지의 <TABLE> 부분에

들어가는 내용은 그대로 구분된 태그에 포인터를 설정한다. 따라서 웹 페이지는 본래의 구조에서 내용은 그대로 유지하면서 구조의 일부가 더해지고 생략되므로 유효한 HTML(Valid HTML)의 형태를 계속 유지한다.

[그림 3-b]는 원본 HTML이 트리 데이터 구조로 표현되었을 때, 각 노드를 전위순회(preorder traversal)로 방문하면서 알고리즘을 수행한다. parameter 에 명시된 index값을 이용하여 원하는 내용(contents)이 포함되어 있는 <TABLE>태그와 최소의 서브 트리를 추출한다.

```
/*
Function ConstructTree
Parameter : current page            n
                    word key_value
*/
DO {
        Classification word each tag and contents using string tokenize
        IF( tag == <TABLE> ) {
        add node <TABLE>;
        IF (node is root table)
                    TableDepth = 1;
        IF (Contents in block has key_value)
                    set page.index[n]          }
        ELSE IF( tag == <TR> ) {
        add node <TR>;
                    IF (node is not in the first row of table) {
                                increase sibling_node
                                define <TR> element pointer;}
                    IF (node's parent is root table) {
                    sibling_node = 0;
                                increase TableDepth; } }
        ELSE IF ( tag == <TD> ) {
                    add node <TD>;
                    IF (node is not in the first col of table) {
                                increase sibling_node;
                                define <TD> element pointer;}
                    IF (node has <TABLE> as child node) {
                                sibling_node = 0; //
                                increase TableDepth;
                    } }
        ELSE IF( other tags )
                    trivial functions for other tags;
        } WHILE ( end-of-tag of HTML )
```

[그림 3-a] HTML을 트리 구조로 구성.

```
/*
Function MakeBlock
Parameter : HTML page,
            current block[n],
            marked point block start,
            marked point block end,
*/
DO {
        FOR (page.block[n].start to page.block[n].end)
            extract( block[n].contents_text)
        add (page, block[n].contents_text)
                    IF (image is inserted) {
                    extract (block.contents_image)
                    add (page, block[n].contents_image)}

                    ELSE IF (form is inserted) {
                    extract (block.contents_form)
            add (page, block[n].contents_form)}

                ELSE IF (video(or flash) is inserted) {
                    extract (block.contents_video)
                    add     (page,     block[n].contents_video)}
        }
        IF (there is no contents)
            cancel ExtractBlock
        } WHILE ( end-of-tag of HTML )
```

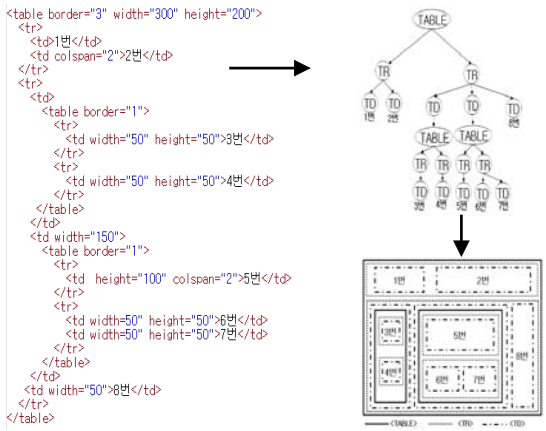
[그림 3-c] Contents block 설정.

추출된 태그와 내용은 block화되고 block의 시작 주소가 부여된다.

[그림 3-c] 알고리즘은 segment block으로 설정된 노드(node)와 노드가 가리키는 내용을 바탕으로 contents block을 생성한다. 태그(tag)에 따라 나누어진 페이지의 내용에 시작점(start point)과 끝점(end point)을 부여 한 후 이를 구역으로 인식하여 추출한다. 구조 분석 단계를 통하여 그 구역 내부에 존재하는 텍스트, 이미지, 비디오 포맷에 적절한 index를 부여하여 컨텐츠의 특성별로 화면에 디스플레이 한다. 예를 들어

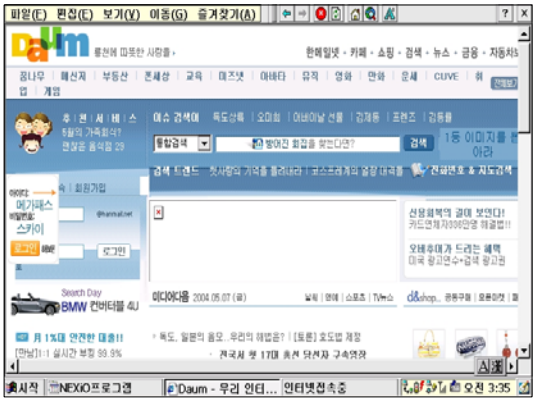
```
/*
Function ExtractBlock
Parameter : current page,
            tag node tree,
                    marked index   n
*/
DO {
                extract next node with preorder traversal
                IF( tag == <TABLE> ) {
                IF( Contents in block has marked index )
                    set page.block[n].start
                ELSE {
                }   }
                ELSE IF( tag == <TR> ) {
                IF (tag is not in the first row of nested table)
                                        increase RowCnt;
                                IF (tag has <TABLE> as child node)
                                        define Width of <TR>;}
                ELSE IF( tag == <TD> ) {
                    IF (tag is not in the first row of nested table)
                    increase ColCnt;
                IF (tag has <TABLE> as child node)
                    define Width of <TD>;                   }

                ELSE IF( other tags ) {
                                trivial functions for other tags;}
} WHILE ( end-of-node of tree )
CALL ExtractBlock;
```

[그림 3-b] Contents를 포함하는 최소 서브트리추출.

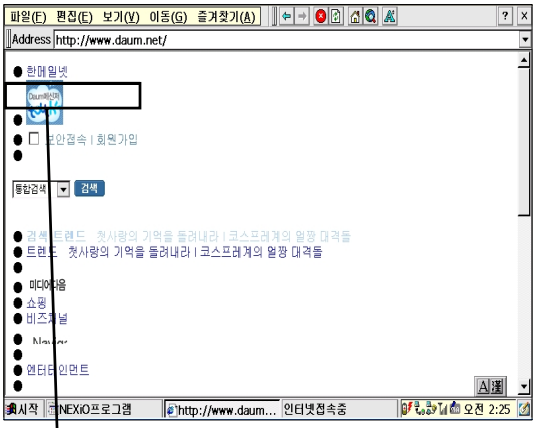


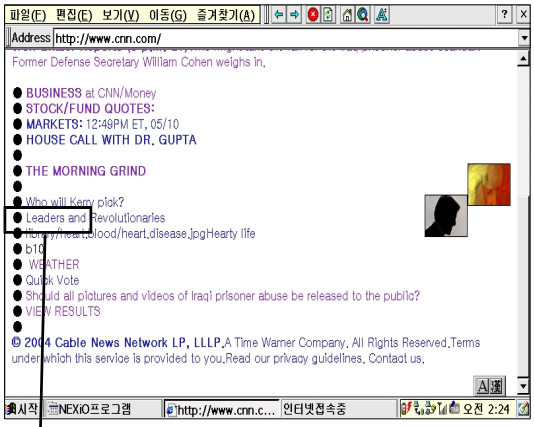
[그림 3-d] HTML 예제 코드에 대한 트리 형태와 구조적 태그 segment block.

Image Index를 생성해야 할 경우, 이미지의 크기 정보를 첨부하고, Text Index는 그 개수만큼 Select Form을 생성한다. [그림 3-d]는 contents block 추출 흐름도에 의해서 HTML 코

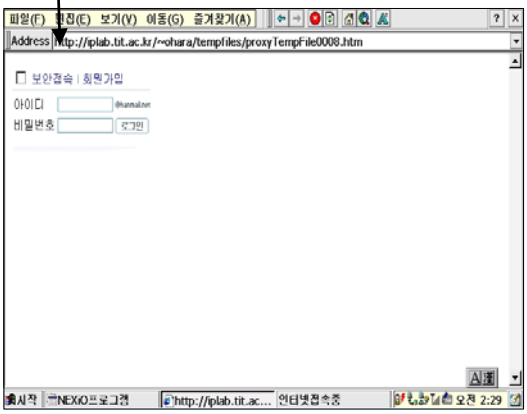


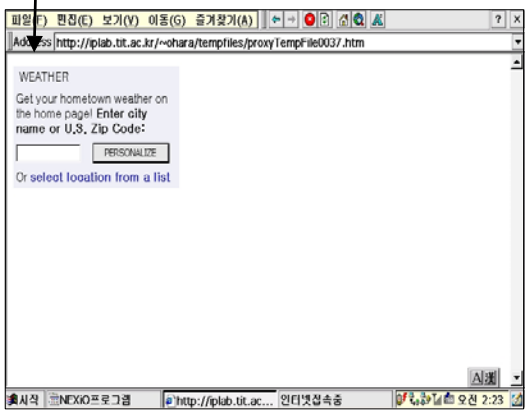
(a) PDA에서 기본적으로 보여주는 전체 웹 페이지





(b-1) Segment block 단위 리스트





(b-2) 리스트 중 쿼리한 contents block

[그림 4] 예제 사이트를 변환한 결과 (Daum, CNN).

드가 트리 형태와 구조적 태그의 시각적 구성인 segment block으로 구성됨을 보이는 예제이다. 아래 HTML의 예제 코드에서 <HEAD>, <BODY> 등의 태그는 생략하고 유효한 <TABLE> ~ </TABLE> 부분만 예제로 보인다.

Ⅳ. 구현 및 성능평가

1. 개발환경

본 논문은 Linux (Red Hat version 8.0) 운영체제 환경에서 개발되었으며, 웹 서버는 Apache 2.0.40을 사용하였다. Compiler는 gcc 3.2이며, Language는 표준 C를 사용하였다. 데이터베이스는 MySQL 3.23.40 환경이며, 사용된 클라이언트는 Nexio S-150 이다. 통신은 CDMA 환경에서 개발되었다.

2. 구현결과

본 논문에서는 사용자가 구현하고자 하는 URL을 이미 알고 있다는 가정 하에 다음 두 개의 예제 사이트를 통해 모바일 환경을 위한 웹 컨텐츠 추출기법을 보이고자 한다.

[그림 4]에서는 Daum과 CNN사이트를 예제로 보인다. (a)항목은 모바일 단말기에서 기본적으로 전체 웹페이지를 읽어오는 화면이며 (b-1)항목은 전체 페이지를 segment block 단위 리스트로 보여주는 화면이다. (b-2)는 이 리스트 중 쿼리한 contents block만 화면에 디스플레이 한다.

3. 성능 평가

이 장에서는 본 논문에서 구현한 모바일 환경을 위한 웹 추출기법의 성능을 평가해 본다. 실험에서는 다양한 웹사이트의 평가를 위해 4개의 대표적인 웹사이트를 고려하였다. 다음 <표 1>에서는 모바일 단말기에서 해당 사이트의 전체 웹페이지를 읽어오는 시간과 본 논문의 추출기법을 통해 읽는 시간을 비교하여 본다. 단, 여기에서의 '평균시간' 이란 하나의 사이트를 5번 측정 후 이것들의 평균시간을 의미한다.

<표 1> 변환 시간 비교를 이용한 성능 평가

URL	전체 웹페이지 읽는 평균시간(s)	추출 기법을 통해 읽는 평균시간(s)	시간 단축 비율
www.daum.net	48	17	35%
kr.yahoo.com	47	18	38%
www.nate.com	46	13	28%
www.joins.com	44	16	36%

위 <표 1>과 같이 전체 웹 페이지를 읽는 평균시간보다 본 논문의 추출기법을 이용하여 읽는 평균시간이 약34% 단축됨을 알 수 있다.

Ⅴ. 결 론

무선 단말기 사용자의 인터넷 접속이 증가하면서 점차 웹 페이지는 유선 사용자 뿐 아니라 무선 사용자의 요구를 만족시켜야할 필요성이 증가하였다. 하지만 현재 웹은 불필요한 정보의 과잉 현상으로 개인에게 필요한 정보나 서비스를 찾기 위해서는 많은 시간과 노력이 필요하다. 또한 모바일 단말기는 데스크 탑에 비해 상대적으로 낮은 대역폭, 작은 화면, 느린 중앙처리장치(CPU)를 가지고 있기 때문에 원활한 웹 검색이 어렵다. 특히 모바일 단말기의 작은 화면은 데스크톱에 최적화 된 웹 페이지를 그대로 보기에는 상당한 제약이 따른다.

본 논문에서는 이러한 단점들을 극복하고 개인화의 장점을 얻고자 원래 웹 페이지에서 제공되는 서비스들을 사용자에 따른 맞춤형 서비스로 제공하였다. 즉 사용자가 접속했던 사이트를 리스트 형태로 보여줌으로써 재 접속시 번거로움을 줄였다. 또한 프락시 기반 필터링 모듈의 설계 및 구현을 통해 웹 전체 페이지를 segment block으로 구분하여 이 중 쿼리한 contents block을 추출

하여 제공함으로써 변환 시간을 줄이고 무선 단말기의 작은 화면의 단점을 극복하고자 하였다.

향후 연구과제로는 태그 자체의 구문 분석과 함께 semantic 정보를 활용함으로써 보다 명확한 웹 문서 분석을 수행이 필요하다.

참고 문헌

[1] Orkut Buyukkokten, et al., Power Browser : Efficient Web Browsing for PDAs, Digital Libraries Lab Stanford University, 2000.

[2] Timothy W. Bickmore, Bill N. Schilit, Digestor : Device-independent access to the World Wide Web, In Proceedings of the 6th International World Wide Web Conference, 1997.

[3] Y. D. Yang and H. J. Zhang, HTML Page Analysis Based on Visual Clues, IEEE International Conference on Document Analysis and Recognition (ICDAR 2001), pp.859~864, 2001. 9.

[4] 신희숙, 마평수, 조수선, 이동우, 소형 화면 단말기를 위한 웹 문서 변환 기법, 2002년 정보처리학회논문지 D, 제9-D권 제6호, 2002. 12.

[5] Wap Forum, White paper, Wireless Internet Today Overwiew, June, 2000.

[6] D. Bilsus, M. J. Pazzani, A Personal News Agent that Talks, Learns and Explains, In Proc, 3rd Int'1. Conf. on Autonomous Agents, pp.268~275, 1999.

[7] D. Bilsus, M. J. Pazzani, J Chen, A learning agent for wireless news access, In Proceedings of the 2000 Conference on Intelligent User Interfaces, 2000.

[8] M. Hori, G. Kondoh, K. Ono, S. Hirose and S. Singhal, An motation-Based Web Content Transcoding, 9th World Wide Web Conference, 2000.

[9] 전영효, 황인준, 모바일 사용자를 위한 웹 서비스 페이지 개인화 기법, 2003년도 한국정보과학회 봄 학술발표논문집 Vol. 30, No.1.