무선 단말기를 위한 웹 페이지의 자동 재구성

송 동 리[†] · 황 인 준[††]

요 약

최근 들어 인터넷이 광범위하게 보급되고 무선 이동통신 기술이 눈부시게 발전하면서 무선 단말기를 통한 인터넷 상에서의 정보 검색이 시간과 장소에 구애받지 않고 가능하게 되었다. 하지만 무선 단말기를 통한 인터넷의 접근은 단말기 자체의 제한과 무선망의 제한으로 인해 기존 데스크 탑 컴퓨터의 유선 상에서 보다 웹 페이지 탐색에 많은 어려움을 가진다. 본 논문은 이러한 제약을 완화시키기 위해서 웹 페이지로부터 페이지 정보를 축소한 페이지 맵을 생성하여 웹 정보를 쉽게 검색할 수 있게 하는 페이지 재구성 시스템을 제안한다. 그리고 대표적인 웹 사이트에 대한 실험을 통해 제안한 시스템이 무선 단말기 상에서 빠른 웹 페이지 로딩 속도를 보장하며, 작은 스크린을 통해서도 효율적인 웹 탐색이 가능함을 보인다.

Automatic Reconstruction of Web Pages for Mobile Devices

Dongrhee Song[†] · Eenjun Hwang[††]

ABSTRACT

Recently, with the wide spread of the Internet and development of wireless network technology, it has now become possible to access web pages anytime, anywhere through devices with small display such as PDA. But, since most existing web pages are optimized for desktop computers, browsing web pages on the small screen through wireless network requires more scrolling and longer loading time. In this paper, we propose a page reconstruction scheme called PageMap to make it feasible to navigate existing web pages through small screen devices even on the wireless connection. Reconstructed pages reduce the file and page size and thus eventually reduce resource requirements. We have implemented a prototype system and performed several experiments for typical web sites. We report some of the results.

키워드 : 재구성(Reconstruction), 무선 인터넷(Mobile Internet), 페이지 구역(Page Region), 무선 단말기(Mobile Device)

1. 서 론

정보화 시대를 맞이하여 인터넷은 이제 사회 전반에서 보편적인 정보 취득의 수단이 되었다. 최근에는 무선 호출기로부터 시작하여 휴대용 전화기를 거쳐 IMT-2000 기반의 통신까지 엄청난 속도로 발전한 무선 이동 통신 환경에서 인터넷에 접속하여 정보를 처리하고 취득하려는 요구와 노력이 증가하고 있다. 한편 무선 단말기는 시공간을 초월한 정보 접근의 가능성을 제시하지만 단말기 하드웨어와 무선망 대역폭(bandwidth)의 제한으로 인해 데스크 탑 컴퓨터를 이용하는 수준의 사용이 어렵다[1-5]. 현재 이러한 무선 단말기들의 프로세서, 메모리, 배터리 성능 등에는 많은 발전이 있었지만, 여전히 작은 스크린은 사용자에게 부담으로 남는다[6].

본 논문에서는 이러한 점을 극복할 수 있는 자동 페이지 맵 추출 시스템을 제안한다. 이 시스템에서는 원래 웹 페이지를 무선 단말기 스크린 크기에 알맞게 형태 그대로 축소한 간단한 페이지 맵을 제공함으로써 페이지 맵의 각 구역을 통해 사용자가 원하는 웹 페이지의 부분만을 선택해서 볼 수 있게 한다. 페이지 맵 시스템의 과정은 크게 두 단계로 나누어진다. 첫 번째 단계는 추출 단계로서 단말기 스크린 크기에 알맞은 웹 페이지의 맵 구조를 추출한 다음 각 맵 구역 속에 웹 페이지의 내용을 요약하여 완성한다. 두 번째 단계는 재구성 단계로서 추출한 페이지 맵을 보고 사용자가 원하는 구역만으로 웹 페이지를 재구성한다. 이런 방법은 웹 페이지를 탐색하는 동안에 가독성을 향상시킬 수 있고, HTML(Hypertext Markup Language)을 기반으로 하기 때문에 컨텐츠 개발이 용이하다는 장점을 가진다. 추가로 페이지 맵에 즐겨찾기 구역(Favorite Regions) 기능을 제공하여 사용자가 원할 때는 매번 접속 시 따로 웹 페이지의 각 구역을 선택하지 않고도 언제나 그 부분만으로 웹 페이지를 재구성해서 볼 수 있게 해준다.

본 논문의 나머지 구성은 다음과 같다. 다음 장에서는 무선 단말기에서의 웹 페이지 접근을 위해 웹 페이지를 요약하는 관련 연구들을 살펴보고, 3장에서는 웹 페이지 코드를 기

※ 본 연구는 한국과학재단 목적기초연구 R05-2002-000-01224-0 지원으로 수행되었음.
† 준 회 원 : 아주대학교 정보통신전문대학원
†† 종신회원 : 아주대학교 정보통신전문대학원 교수
논문접수 : 2002년 7월 27일, 심사완료 : 2002년 10월 29일

반으로 페이지 맵을 자동 추출하는 알고리즘을 기술한다. 4장에서는 페이지 맵 시스템의 전체적인 구조와 웹 페이지 재구성 과정을 살펴보고, 5장에서는 무선 단말기에서 현재 일반적으로 보여지고 있는 웹 페이지와 제안된 방법에서의 웹 페이지 성능을 비교한 실험 결과를 보여준다. 끝으로 6장에서는 결론과 함께 향후 과제에 대해 서술한다.

2. 관련 연구

현재 작은 무선 단말기 스크린에 효과적으로 웹 페이지를 표현하기 위한 다양한 연구가 진행중이지만 웹 페이지를 단말기 스크린에 표현함에 있어 지금까지의 해결책에는 여러 가지 문제가 대두된다. 첫째, 웹 상에 존재하는 웹 페이지를 배재한 채 무선 단말기만을 위한 새로운 페이지의 재생성은 많은 부담을 초래한다. 최근에 이와 같은 방법의 하나로 WML (Wireless Markup Language)[7]이 등장 하였지만 이는 데스크 탑 컴퓨터와 무선 단말기의 작은 스크린에 맞는 페이지를 각각 별도로 준비해야 하는 부담과 함께 WWW를 이원화 시키는 위험을 가진다. 그러므로 현재 존재하는 웹 페이지를 이용하는 방법이 요구된다.

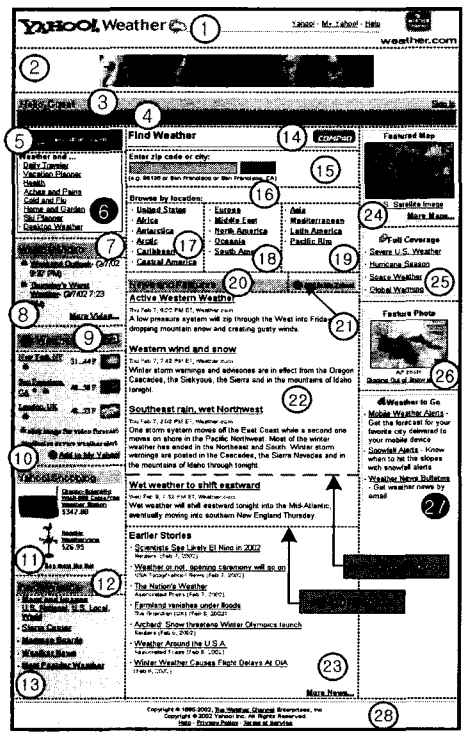
둘째, 현재의 웹 페이지를 사용하지만 웹 페이지의 구조를 무선 단말기에 맞게 변형하는 방법으로서 다음과 같은 것들이 있다. WEST[8]는 각 페이지를 초점 + 문맥관계(focus + context) 방식을 이용해서 작은 화면에 알맞은 카드 형식의 요약을 제공한다. Digestor[9]는 구조적 페이지 변형(structural page transformation)과 문장 탈락(sentence elision)을 이용해서 점증적인 요약을 제공한다. Power Browser[3]는 웹 페이지를 의미적 텍스트 단위(semantic textual units, STUs)에 의해 나누어서 요약한 다음 그 요약을 3 단계에 걸쳐 나누어 보여준다. 주석기반의 코드 변형 시스템(Annotation-based transcoding system)[4]은 저작 도구를 사용하여 각 페이지의 내용에 외부 주석 처리를 함으로써 웹 페이지의 정확한 나누기와 요약이 가능하다. 하지만 이런 방법들은 사용자의 웹 탐색을 비효율적으로 만든다. 왜냐하면 동일한 웹 페이지에 접속시 데스크 탑 컴퓨터에서 보여지는 웹 페이지 내용의 순서와 위치가 무선 단말기 화면에서는 다르게 일률적으로 정해진다면 사용자는 전체 흐름을 알수가 없어 웹 탐색에 불편함을 가지기 때문이다. 따라서 본래 웹 페이지의 구조를 변형하지 않고 웹을 탐색할 수 있는 방법이 필요하다.

셋째, 여러 가지 방법으로 진행되었던 무선 단말기 사용자 인터페이스를 위한 기존의 연구 방법은 저 해상도 텍스트를 기반으로 구현[1, 2, 10, 11]하고 있어, 최근 급속히 발전하는 무선 단말기들의 성능을 충분히 이용하지 못한다. 예를 들어 현재 개발되어 판매되고 있는 PDA 제품들 중 200Mhz 이상의 CPU와 64MB의 메모리, 65000가지 이상의 색을 나타낼 수 있는 성능의 제품이 출시되고 있어서 예전과는 다르게 무선 단말기들에도 다양한 멀티미디어의 환경 구축이 가능해졌다. 따라서 이런 환경을 충분히 활용한 좀 더 효율적인 웹 탐색 방법이 필요하다.

3. 페이지 맵 추출 과정

3.1 페이지 맵의 필요성

웹 페이지 저자들은 사용자들의 시각적 편의를 위해 한 페이지에 최대한 많은 정보를 넣으려고 한다. 따라서 이러한 정보들이 정해진 순서없이 비 규칙적으로 나열되어 있으면 사용자에게 혼란을 줄 수 있으므로, 웹 저자들은 글자체(font) 및 배경색 등 시각적 효과와 함께 테이블 구조를 사용해서 웹 페이지를 동질의 종류별로 분류시키며, 웹 페이지내의 오류 방지와 명확성을 위해서 이미지는 "ALT" 값으로, 폼(form)은 "TITLE" 이나 "NAME" 값을 명시한다. 이런 점에 착안하여 본 논문에서는 웹 페이지를 테이블 코드 흐름에 따라서 동질의 내용으로 자동 분류할 수 있는 방법을 제안한다. 여기서 코드에 따라 나누어지는 각 동질의 내용 부분을 구역(region)이라 정의한다. 나누어진 각 구역들은 페이지 맵 구성을 위해서 그 구역의 짧은 요약과 구역 번호가 할당되며, 사용자는 이렇게 구성된 페이지 맵의 구역 선택을 통해서 웹 페이지를 좀 더 쉽고 편하게 탐색할 수 있게 된다. 따라서 페이지 맵 시스템은 개인 홈페이지 같은 간단한 구성의 일률적 웹 페이지보다는 신문, 방송, 기업 등의 홈페이지처럼 다양한 내용과 복잡한 구성으로 이루어진 웹 페이지에 더 효율적으로 적용될 수 있다. (그림 1)은 실제 야후 날씨 웹 페이지에서 페이지 맵 시스템에 의해 28개로 나누어지는 구역의 도식을 보여준다.



(그림 1) 야후 날씨 웹 페이지의 구역 인식

3.2 페이지 맵 추출 알고리즘

현재 웹 상에 존재하는 웹 페이지는 간단하고 배우기가 쉬운 장점 때문에 대부분 HTML을 사용하여 제작되었다. 이러한 HTML 페이지 구조는 특성상 <HEAD> 부분의 내용을 같게 한다면 <BODY> 부분의 내용을 물리적 구조에 변화를 주지않는 태그 단위로 추가, 삭제해도 원래 웹 페이지 형태 그대로의 재구성이 가능하다[13]. 또한 <TABLE> 태그는 현재 HTML 웹 페이지에서 사용되는 구조적 태그로서 웹 페이지의 내용을 정렬할때 가장 일반적으로 많이 사용된다. 따라서 HTML문서에서 페이지 맵을 추출하기 위해 <TABLE> 태그를 기준으로 사용되는 태그를 물리적 구조에 영향을 주는 여부에 따라 간소화(simplify) 시킨다. HTML 문서는 크게 <HEAD>와 <BODY>로 구성이 되는데, 이들 각각의 부분에 대한 간소화는 <표 1>과 같이 진행한다. 여기서 웹 페이지의 내용 일부분이 스크립트를 사용한 템플릿(template)으로 구성되어 있다면 이것도 하나의 구역으로 인식된다((그림 1)의 구역 11). 이렇게 웹 페이지가 구역을 기준으로 재구성될때 재구성된 페이지의 코드는 전체 웹 페이지 코드의 부분집합으로 구성되어 사용자에게 보여지게 된다.

〈표 1〉 페이지 맵을 위한 태그의 간소화

HEAD 부분			팝업 창을 위한 스크립트인 "window.open" 부분을 제거
BODY 부분	전 체		주석(comment) 처리된 부분은 추출 과정에서 제외
	테이블	<TABLE> <TBODY>	• 속성(attribute)은 그대로 적용 • 테이블 사이의 간격을 위한 속성인 "cellpadding", "cellspacing"은 0값으로 조정 • 테이블 경계선(border)는 모든 선을 나타낼 수 있도록 일정한 값으로 조정
		<TR> <TD>	• 속성은 그대로 적용 • 태그 안의 내용은 모두 제거
	템플릿	<SCRIPT>	템플릿 전체를 하나의 구역으로 인식

페이지 맵을 추출하기 위해 사용자의 요청을 통해서 프록시 서버(proxy server)로 넘어온 웹 페이지는 다음에 기술할 네 가지의 알고리즘에 적용된다. 우선 웹 페이지를 파싱(parsing)하여 트리(tree) 구조로 표현하면 트리 구조의 각 태그에 따른 깊이(level)를 알 수가 있다. 그리고 페이지에서 <HTML><HEAD>~</HEAD><BODY> 부분을 추출하여 기본 골격을 가지는 새로운 페이지를 생성한다. 이 페이지의 <BODY> 부분에 들어가는 내용은 페이지 맵 각 구역에 해당되는 내용으로 "SplitPage"에서 처리한다. 따라서 웹 페이지는 본래의 구조에서 페이지 형태를 그대로 유지하면서 일부가 더해지고 생략되므로 유효한 HTML(valid HTML)의 형태를 계속 유지할 수 있다(알고리즘 1).

```
Algorithm PageReconstruct
Parameter : URL for HTML page

  parse (URL, HTML page, page [n]) // parse HTML page into page [n]
  save away page [n].header_part     // <html><head>~</head><body>
  if (popup window is exist)
     remove popup window script  // "window.open" part
  call SplitPage ;                   // divides page into regions
  cut away page [n].footer_part      // </body>~</html>
  create valid HTML file structure
```

(그림 2) 알고리즘 1 : Page Reconstruct

```
Function SplitPage
Parameter : HTML element,
            current page n,
            marked point i

  if (tag ≠ table>) then          // contents before first table tag
     set page [n].point [i].start
     while (tag ≠ table>) do
           read code
     set page [n].point [i].end
     call MakeRegion ;
     reset page [n].point [i]
  set page [n].point [i].start       // tag == <table>
  while (tag == </table>) do
        if (tag == <table>) then // inner table
           call InnerTable ;
  set page [n].point [i].end
  call MakeRegion ;
  reset page [n].point [i]
  set page [n].point [i].start
  while (tag == </body>) do    // contents after last table tag
        read code
  set page [n].point [i].end
  call MakeRegion ;
  reset page [n].point [i]
```

(그림 3) 알고리즘 2 : Split Page

페이지 맵 각 구역의 내용을 구성하기 위해 먼저 웹 페이지를 구역으로 나누어야 한다. 구역으로 나누기 위해서 파싱된 HTML문서에서 웹 페이지의 <TABLE> 태그 위치를 찾고, 일단 그 위치가 파악되면 구역의 경계는 크게 다음과 같이 분류되어 페이지의 내용을 나눌 수 있게 된다.

- <BODY>~<TABLE>
- <TABLE>~</TABLE>
- </TABLE>~</BODY>

이렇게 경계에 따라 나누어진 페이지의 내용에 시작점(start point)과 끝점(end point)을 부여한 후 이를 구역으로 인식한다. 이 구역의 생성은 "MakeRegion"에서 처리한다(알고리즘 2, 4).

```
Function InnerTable

 set page [n].point [i].end
 call MakeRegion ;
 reset page [n].point [i]
 set page [n].point [i].start
 while (tag == </table>) do
     if (tag == <table>) then
         call InnerTable ;
 set page [n].point [i].end
 call MakeRegion ;
 reset page [n].point [i]
 set page [n].point [i].start
```

(그림 4) 알고리즘 3 : Inner Table

```
Function MakeRegion
Parameter : page[n].point[i].start,
            page[n].point[i].end,
            current region j

 for (k ← page [n].point [i].start to page [n].point [i].end) do
    if (there is no region number)
       insert (region [j].region_number)
    extract (region [j].contents_text)
    add (page [n], region [j].contents_text)
    // extract simplified contents in table data field
    if (image is inserted) then     // output [I]
       extract (region [j].contents_image)
       add (page [n], region [j].contents_image)
    // identify image and its name by "alt" or hyperlink value
    else if (form is inserted) then // output [F]
       extract (region [j].contents_form)
       add (page [n], region [j].contents_form)
    // identify form and its name by form "name" or "title" value
    else if (video (or flash) is inserted) then  // output [W]
       extract (region [j].contents_video)
       add(page [n], region [j].contents_video)
    // identify video (or flash) and its name
       // video (or flash) file name
 if (there are no contents) then      //don't make the region
    cancel region [j].region_number
```

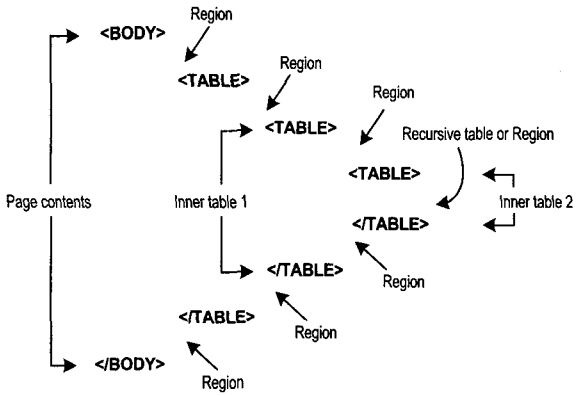
(그림 5) 알고리즘 4 : Make Region

또한 <TABLE> 태그안에 또 다른 <TABLE> 태그를 포함하는 경우를 고려해야 한다. 이를 내부 테이블(Inner table)이라 부르고 반복적인 알고리즘으로 검색하여, 위에서 나누어진 분류에 다음과 같은 세부적인 분류를 추가한다.

- <TABLE>~<TABLE>
- <TABLE>~</TABLE>
- </TABLE>~</TABLE>

즉 내부 테이블을 포함한 모든 웹 페이지의 <TABLE> 태그를 깊이에 따른 순서에 따라 차례로 분류하여 (그림 6)과 같이 웹 페이지를 구역화 시키게 된다(알고리즘 3).

끝으로 위에서 정해진 구역의 시작점과 끝점을 가지고 구



(그림 6) HTML 페이지의 구역화 예제

역을 생성하게 되는데, 이 때 페이지 맵에서의 각 구역 명시를 위해 간단한 요약을 포함하게 된다. 텍스트 요약은 다음과 같이 정의된다.

[정의 1] 페이지 P에서의 구역 R은 $R_i \subset P$, $i = 1, \cdots, n$, (n은 구역의 수). 구역 R_i의 문자열 길이를 $|R_i|$라 하고, $R_i[j]$는 R_i문자열의 1부터 j까지의 문자열이라 하면 각 구역 R_i의 요약 S_i(기호 $\in S_i$, 공백 $\notin S_i$)는

(1) $|R_i| > 15$ 일때,

$S_i = R_i$ [15문자열] + "…"

만약 $S_i <$ 3단어 이면 $S_i = R_i$[2단어]

$S_i \geq$ 3단어 이면 $S_i = R_i$[3단어]

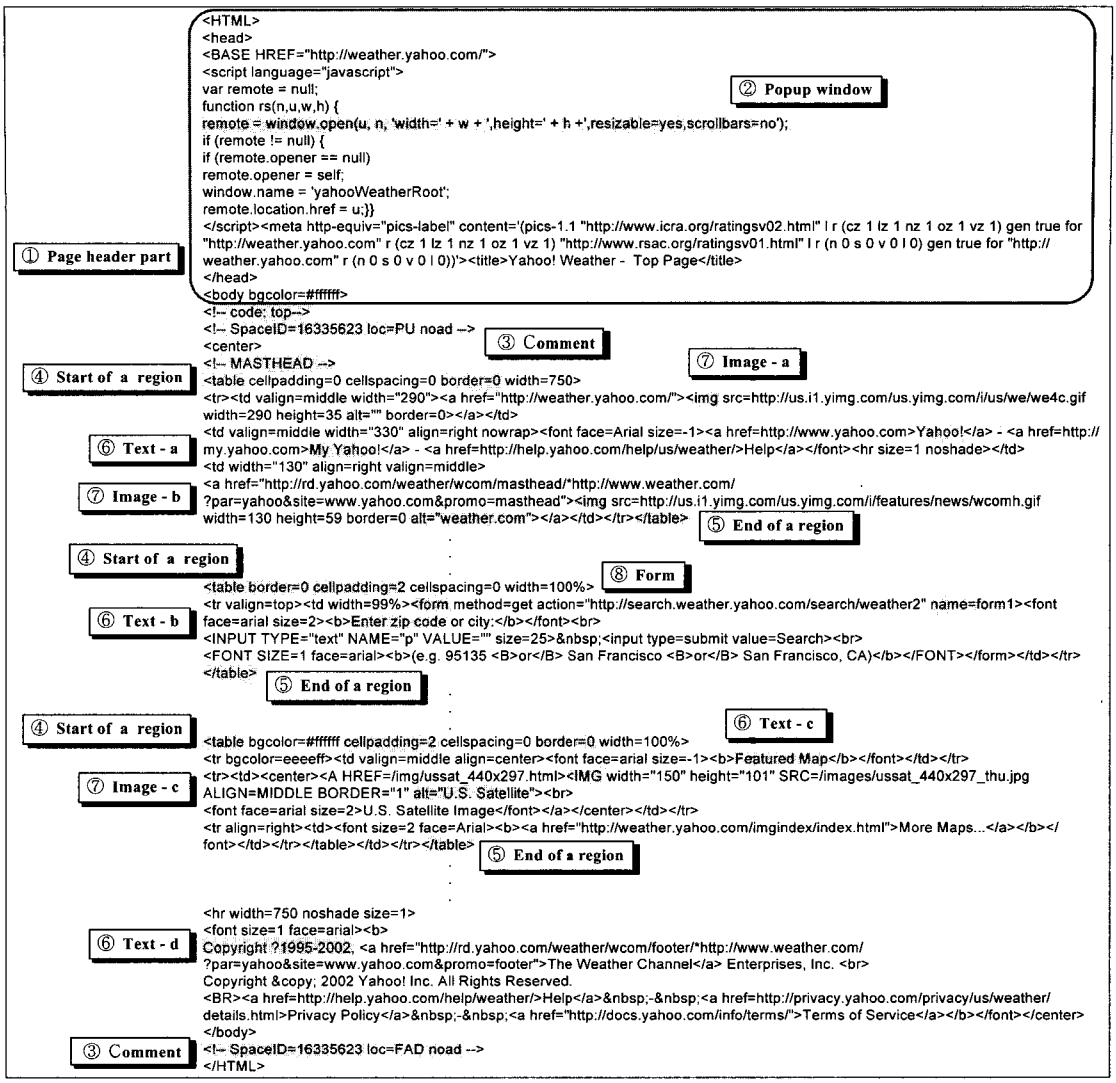
(2) $1 \leq |R_i| \leq 15$ 일때,

$S_i = R_i$ 전체 문자열

이미지의 경우에는 이미지 태그 속성의 하나인 "ALT" 값으로 대체되며, "ALT" 값이 있다 하더라도 링크가 존재하지 않으면 대체하지 않는다. 또한 폼은 "TITLE"이나 "NAME"의 속성값을 이용하여 대체되며, 최근 웹 페이지에서 자주 사용되는 동영상과 플래시(flash)는 소스 경로에서 파일의 이름으로 추출된다. 이는 대부분의 기업 사이트에서 웹 페이지에 파일을 삽입할때 이러한 파일들을 가장 잘 나타낼 수 있는 이름으로 명명하기 때문에 추출된 파일 이름을 가지고 추측하는 것이 가능하다. 또한 웹 페이지 구성의 여백을 위한 빈 테이블 태그는 위에서 구역으로 정해 졌더라도 그 테이블안에 내용이 없음을 인식한 후에는 구역으로 생성하지 않는다(알고리즘 4).

3.3 페이지 맵 생성 예제

무선 단말기 사용자는 3.2절에서 기술한 웹 페이지의 페이지 맵 추출 알고리즘을 통해 생성된 페이지 맵으로 웹 페이지를 쉽게 탐색할 수 있다. 여기서는 실제 야후 날씨 페이지



(그림 7) 페이지 맵 생성 예제

(http://weather.yahoo.com)의 알고리즘 적용 예제를 통해 페이지 맵의 생성을 알아본다. (그림 7)은 실제 야후 날씨 페이지의 코드이며 명시된 적용 기준은 다음과 같다.

3.3.1 Page header part - 알고리즘 1

요청된 URL에 의해 넘어온 HTML페이지의 머리 부분(header part)을 가져온다. 즉 표시된 "<html><head>~</head><body>" 태그까지의 내용이다. 그리고 페이지의 바닥 부분(footer part)은 "~</body></html>"로 자동 생성되어 유효한 HTML 페이지를 구성하게 된다.

3.3.2 Popup window - 알고리즘 1

머리 부분을 가져오는 도중 자동 팝업 창 생성을 위한 "window.open" 스크립트가 있으면 이 부분을 제거한다.

3.3.3 Comment

<body>의 내용 중 "<!--" 와 "-->"로 처리된 모든 웹 페이지의 주석 부분은 알고리즘을 적용하지 않고 건너 뛴다. 단 스크립트 구문의 일부로 사용된 것은 제외한다.

3.3.4 Start of a region - 알고리즘 2, 알고리즘 3

<table> 태그로 구역의 시작을 알 수 있으며 속성인 "cellpadding"과,"cellspacing"의 값은 전부 0으로, "border"의 값은 2이상으로 처리함으로써 구역이 명확하게 구분된 최소한의 페이지 맵 구조를 구성하게 된다.

3.3.5 End of a region - 알고리즘 2, 알고리즘 3

<table> 태그 후에 </table> 태그는 구역의 끝을 가리키며, 이를 하나의 구역으로 인식, 구역을 생성하게 된다. 또한 </table> 태그 후에 <table> 태그까지에도 내용이 있다면 구역으로 인식하여 생성한다.

3.3.6 Text - 알고리즘 4

구역 안의 텍스트 요약은 각 구역의 첫번째 테이블 필드에서만 추출되며(없을시 다음 필드), 내용은 3.2절에서 정의된 텍스트 요약에 의해서 각각 다음과 같이 추출된다.

태그만으로 이루어지는 간소화된 HTML(simplified HTML) 문서와 내용 요약자(Contents Summarizer)에서 각 구역의 내용을 제목과 같은 짧은 요약으로 생성하는 부분으로 각각 나누어지며 이들을 기반으로 페이지 맵 생성자에서 페이지 맵을 생성한다.

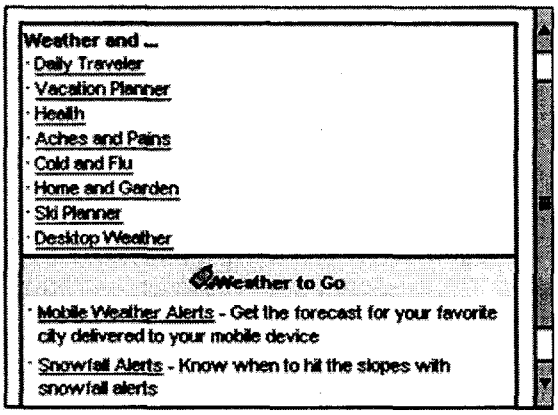
- **웹 페이지의 재구성** : 기본적으로 프록시 서버와 연동되는 데이터베이스에는 클라이언트 연결과 동시에 클라이언트 환경에 대한 단말기 정보와 사용자의 저장 유무에 따라 즐겨찾기 구역 정보를 가지고 있다. 즐겨찾기 구역이란 인터넷 익스플러어(Explorer)의 즐겨찾기(favorites)나 넷스케이프의 북 마크(bookmarks)처럼 사용자가 원할때 프록시 서버의 데이터베이스로부터 미리 저장해 놓은 구역을 불러올 수 있는 기능을 말한다. 또한 단말기 정보중에는 스크린의 크기가 가장 중요한 요소로서 이는 페이지 맵의 폰트 크기, 각 구역 내용의 요약 수준을 정하는 기준이 된다. 일단 사용자가 무선 단말기를 통해 웹 사이트에 접근할때 프록시 서버는 그 웹 페이지의 즐겨찾기 구역 정보가 있는지를 조사한다. 만약 정보가 있고 사용자가 원한다면 그 구역만을 불러와서 재구성이 되며, 그렇지 않다면 요청된 웹 페이지를 가지고 페이지 맵을 자동 추출한다. 사용자는 이렇게 추출된 페이지 맵의 구역 선택을 통해서 웹 사이트를 탐색하게 되는데, 즉 사용자가 구역을 선택하게 되면 프록시 서버는 사용자에게 그 구역에 해당하는 내용만으로 페이지를 재구성해서 돌려준다.

4.2 페이지 맵을 통한 웹 페이지의 탐색

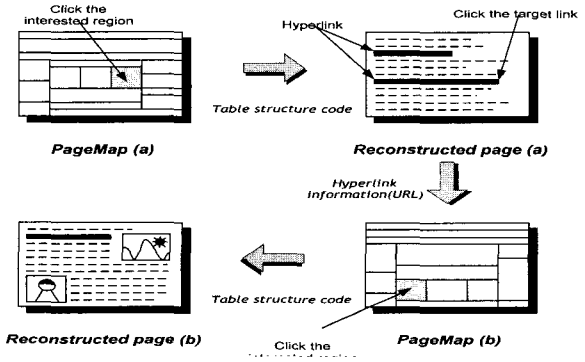
대부분의 웹 페이지의 해상도는 최하 800×600 해상도로 디자인 되어있는 반면 작은 무선 단말기는 일반적으로 320×240의 스크린을 가지고 있다. 이런 환경은 웹 페이지의 가독성과 상호 작용을 제한할 수 있으며[2, 12], 사용자로 하여금 많은 스크롤링(scrolling)을 요구하게 된다. 이런 이유로 페이지 맵 시스템에서는 사용자에게 페이지 로딩 시간과 스크롤링을 줄일 수 있는 웹 페이지의 맵을 제공하며, 이 맵은 사용자에게 좀더 웹의 내용을 이해하기 쉽고 사용하기 편리하게 한다.

또한 무선 단말기의 사용자 인터페이스 부분은 데스크 탑 컴퓨터와는 다른 무선 단말기의 화면 특성 때문에 터치 스크린 기능을 포함한 간단한 키 조작에 맞게 따로 고려되어야 한다. 작은 키패드(keypad)를 이용해 모든 조작이 이루어진다는 것은 단말기 사용에 익숙하지 않은 사용자들에게는 불편함으로 다가올 수 있기 때문이다. 여기서 적용될 수 있는 다중 뷰 상호작용(Multi-view interaction) 방법은 상대적으로 불편한 무선 단말기의 인터페이스 상에서 불필요한 탐색을 최소화할 수 있도록 하나의 화면을 여러 화면으로 분할하여 표현하는 방법으로 PAD++[10]와 CZ Web[11]에서의 방법과 유사하게 무선 단말기의 웹 페이지를 분할, 적용하기 위해 사용한다. 즉 웹 페이지에서 여러 부분으로 구분된 구역을 하나의 페이지 맵으로 생성한 후 각 부분에 대해 사용자가 접근 가능한 인터페이스를 구현하는 것이다. 이 부분의 인터페이스에 대한 것은 추후 과제로 남겨져 있으며 현재는 각 구역의 번호를 입력받아 구역을 선택할 수 있게 하였다.

(그림 10)은 사용자가 (그림 8) (b)의 페이지 맵을 보고 (그림 1)에서의 구역 6과 27만을 선택했을때, 자동적으로 그 구역만을 추출하여 재구성한 웹 페이지의 모습이다. 이 재구성된 페이지는 충분히 작으므로 무선 단말기의 화면에 알맞게 보여질 수 있다. 이렇게 추출된 페이지 맵을 통해 사용자는 구역을 선택하게 되고 그 구역만으로 새로운 웹 페이지가 재구성된다. 재구성된 웹 페이지의 구역 내용이 다른 페이지로의 링크를 가지고 있다면 그 페이지로의 이동이 가능하며 이렇게 연결되는 페이지는 또 다시 페이지맵으로 추출되어 보여진다. (그림 11)은 사용자가 이러한 방법으로 웹 페이지를 효율적으로 탐색하는 것을 보여준다.



(그림 10) 웹 페이지의 재구성



(그림 11) 페이지 맵을 이용한 웹 페이지 탐색

5. 성능 평가

우리는 웹 브라우징 기능을 가지는 PDA와 같은 무선 단말기를 실험하기 위해 데스크 탑 컴퓨터에서 단말기와 같은 크기의 브라우저 사용이 가능한 프로토타입 시스템을 만들고 이것을 이용하여 성능을 평가하였다. 실험에서는 다양한

- Text-a : "Yahoo! - My Yahoo!" , [정의 1] (2) 문자열 전체
- Text-b : "Enter zip code…", [정의 1] (1) 3 단어이상
- Text-c : "Featured Map", [정의 1] (2) 문자열 전체
- Text-d : "Copyright ?1995-2002, … ", [정의 1] (1) 3단어 이하

3.3.7 Image- 알고리즘 4

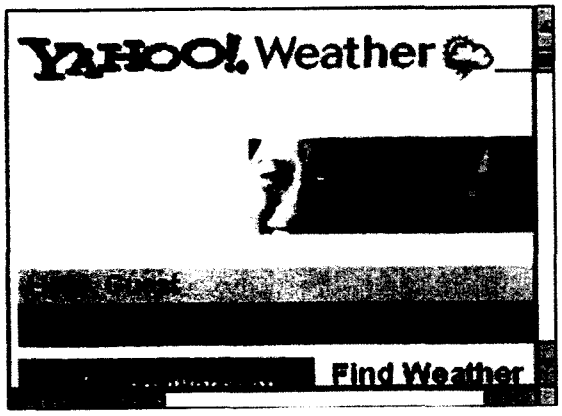
구역안의 이미지는 "< img" 태그로 인식되며 "[I]" 표시와 함께 속성인 "alt" 값으로 요약된다. 따라서 이미지의 요약은 다음과 같다.

- Image-a : "[I]" (alt값이 없음)
- Image-b : "[I] weather.com"
- Image-c : "[I] U.S. Satellite"

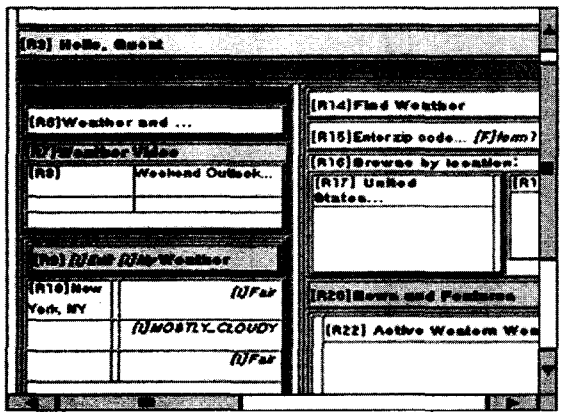
3.3.8 Form - 알고리즘 4

구역 폼은 "< form" 태그로 인식되며 "[F]" 표시와 함께 속성인 "name" 값으로 요약된다. 즉 "[F] form1"으로 요약된다.

이런 과정을 통해 사용자는 (그림 8) (b)와 같은 완성된 페이지 맵을 이용할 수 있게 된다.



(a) 일반적인 야후 웹 페이지의 모습

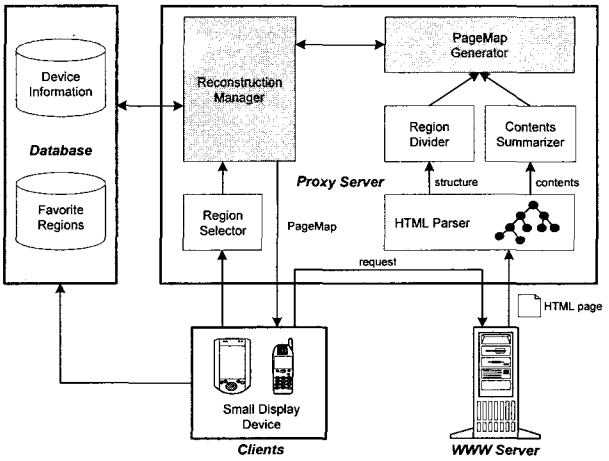


(b) 야후 웹 페이지의 페이지 맵 추출

(그림 8) 320×240의 해상도를 가지는 무선 단말기 스크린에서의 비교

4. 페이지 맵 시스템 구조 및 재구성 과정

본 시스템의 클라이언트는 무선 단말기로서 하드웨어 성능이나 네트워크 연결, 사용자 환경 등에 쉽게 영향을 받을 수 있기 때문에[5], 모든 변형(transform) 모듈을 프록시 기반으로 구현한 Digestor[9]에서와 같이 좀 더 효율적인 통신을 위해서 클라이언트와 서버사이에 프록시 서버를 두고 이곳에서 재구성을 수행한다. (그림 9)는 시스템의 전체적인 구조를 보여준다.



(그림 9) 전체 시스템 구조

4.1 시스템 구조

페이지 맵 시스템은 크게 (1) 웹 사이트와 통신하기 위한 클라이언트 부분, (2) 웹 사이트의 WWW 서버 그리고 (3) 웹 페이지의 재구성을 위한 프록시 서버로 구성된다. 프록시 서버는 HTTP 프록시 서버의 확장과 같이 구현되며, 재구성 모듈들을 통합, 관리한다. 이 서버는 크게 페이지 맵 생성자(Page Map Generator)와 재구성 관리자(Reconstruction Manager)의 두 부분으로 이루어지며, 특히 페이지 맵 생성자는 재구성 시스템의 핵심 부분이 된다. 사용자가 무선 단말기를 통해 웹 페이지를 요청할 때 다음의 세 단계를 통해서 웹 페이지 탐색이 가능해 진다 : (1) 요청 받은 HTML 문서의 분석과, (2) 분석 결과로 이루어지는 페이지 맵의 추출, (3) 사용자의 선택에 의한 웹 페이지의 재구성이다.

- **요청받은 HTML 문서의 분석** : 사용자 요청에 의해 WWW 서버로부터 넘겨받은 HTML 문서는 프록시 서버에서 HTML 파서를 통해 트리 구조로 표현되는 구조화된 HTML 문서(structured HTML)로 표현된다. 즉 본래의 웹 내용을 그대로 유지함과 동시에 트리 구조의 각 태그에 따른 깊이를 가지게 된다.
- **페이지 맵의 추출** : 구조화 된 HTML 문서는 태그의 간소화에 따라 구역 구분자(Region Divider)에서 구조적인

웹 사이트에서의 평가를 위해 신문, 방송, 기업, 쇼핑, 정부 등을 포함하는 10개의 대표적인 웹 사이트를 고려하였으며 그 리스트는 <표 2>에 나타나 있다.

〈표 2〉 성능 실험을 위한 10개의 사이트

	Site	Page	URL
1	USA-TODAY	Auto Track	http://www.usatoday.com/money/autos/autofront.htm
2	Washington Times	America's Newspaper	http://www.washingtontimes.com/
3	CNN	Education	http://fyi.cnn.com/fyi/teachers.ednews/
4	ABC	World Index	http://abcnews.go.com/sections/world/
5	White House	Homeland Security Actions	http://www.whitehouse.gov/homeland/
6	Yahoo	Weather	http://weather.yahoo.com
7	IBM	IBM Small business center home	http://www-1.ibm.com/businesscenter/us/smbusapub.nsf/detailcontacts/SBCenter59BB
8	SUN	Consulting	http://www.sun.com/service/sunps/
9	AOL	Health Wellness	http://www.aol.com/community/health.html
10	ACM	ACM Store	http://store.acm.org/acmstore/

〈표 3〉 웹 페이지의 파일 용량 크기와 화면 크기의 실험 결과

(a) 파일 용량 크기 비교

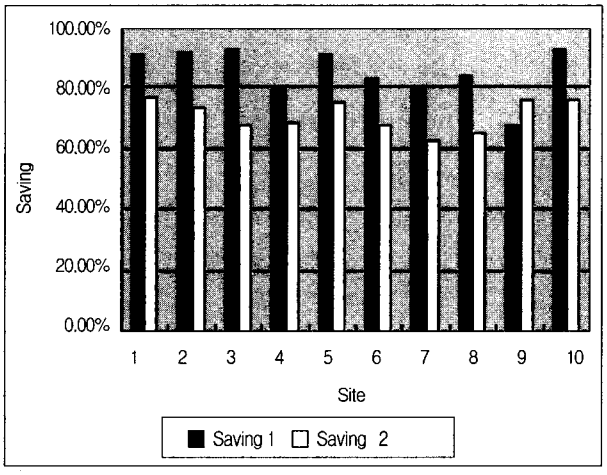
Site	Original page File size (A)	PageMap File size (B)	Saving 1 (A-B)/A*100
1	115.4 Kb	10.9 Kb	90.6 %
2	131.6 Kb	11.3 Kb	91.4 %
3	148.3 Kb	11.6 Kb	92.2 %
4	123.9 Kb	25.0 Kb	79.8 %
5	128.6 Kb	12.1 Kb	90.6 %
6	66.2 Kb	11.1 Kb	83.2 %
7	136.8 Kb	27.2 Kb	80.1 %
8	78.8 Kb	12.6 Kb	84.0 %
9	54.0 Kb	17.5 Kb	67.6 %
10	85.3 Kb	6.3 Kb	92.6 %

(b) 화면 크기 비교

Site	Original page Display size (A)	PageMap Display size (B)	Saving 2 (A-B)/A*100
1	651×1514	580×396	76.7 %
2	769×1925	711×547	73.7 %
3	772×1333	708×468	67.8 %
4	778×1494	723×504	68.6 %
5	754×1366	683×370	75.5 %
6	763×1263	701×441	67.9 %
7	765×1012	706×416	62.1 %
8	1005×1066	894×422	64.8 %
9	676×1856	596×509	75.8 %
10	738×1334	639×367	76.2 %

웹 페이지의 접근에 있어 무선 단말기 사용자들은 낮은 대역폭으로 인하여 데스크 탑의 사용자들보다 더 오래 기다려야 하고 작은 스크린으로 인해 더 많은 스크롤링을 해야만 한다. <표 3>은 이러한 제약을 실험하기 위해 각 페이지의 용량 크기와 화면 크기를 분석, 비교하였다. 페이지 용량 실험은 무선 단말기의 대역폭에 대응하고, 화면 크기 실험은 스크롤링의 빈도에 대응한다. 즉 페이지의 적은 용량은 더 빠른 통신 속도를 보장하고, 작은 페이지의 크기는 스크롤링의 감소를 나타낸다. 실험 결과에 의하면 평균적으로 파일 용량의 크기는 85.2%, 화면 크기는 70.9%가 감소됨을 알 수가 있다.

(그림 12)는 위에서 실험한 각 사이트들의 절약율을 비교하였다. 특히 페이지 1, 2, 3, 5, 10은 웹 페이지 내용의 대부분이 큰 용량의 이미지로 구성되어 있어서 페이지 맵으로의 구성시 많은 페이지 용량의 감소를 보인다. 또한 다른 페이지들과는 달리 페이지 9에서는 페이지 용량의 감소보다는 화면 크기의 감소가 더 큰 것을 볼 수 있는데, 이는 이 페이지의 내용이 이미지보다는 주로 다수의 텍스트로 구성되어 있었기 때문이다. 즉 적은 용량을 차지하는 텍스트는 전체 페이지 용량의 감소에는 크게 영향을 미치지 못했으나 긴 텍스트가 요약되면서 화면 크기가 현저하게 줄어든 것이다.



(그림 12) 파일 용량 크기와 화면 크기 절약 비율의 비교

또한 페이지 6의 실험 도중에 의도한 바와 달리 구역이 잘못 나누어지는 경우가 있었는데((그림 1)의 잘못된 구역), 이는 극히 드문 경우로서 웹 저자들의 잘못된 코드 쓰기에 기인한다. 경계선(border)이 없는 테이블 사용으로 인해 이러한 점는 브라우저를 통해 볼때는 나타나지 않으나 이는 웹 저자들이 같은 종류의 내용을 테이블 단위로 나누어 사용하는 과정에서 오류를 범한 것이다. 실험 결과에서 보듯이 페이지

맵 시스템은 무선 단말기에서 웹 페이지를 탐색하는데 상당히 효율적임을 알 수 있다.

6. 결 론

현재 웹의 내용들은 데스크 탑 컴퓨터에서 뿐만아니라 무선 단말기에서도 사용된다. 이런 이유로 현존하는 웹 페이지들은 각각의 기기에 맞게 표현되어야 하지만 무선 단말기는 데스크 탑 컴퓨터에 비해 제한된 자원을 가지므로 사용자에게 많은 불편함을 초래하게 된다. 따라서 본 논문은 무선 단말기의 이러한 제한된 특성을 해결하기 위해 원래의 페이지 모양 그대로 축소된 페이지 맵을 자동으로 추출할 수 있는 방법을 제안하였으며, 특히 별도의 페이지 준비없이 현존하는 웹 페이지에서 생성되게 하였다. 페이지 맵은 또한 제한된 대역폭의 환경에서 웹 페이지 로딩 시간을 줄여주며, 무선 단말기 스크린에 적당한 크기로 줄여 적은 스크롤링 만으로도 웹 페이지의 탐색이 가능하게 해준다. 그리고 궁극적으로는 즐겨찾기 구역 기능으로 편리성과 가독성을 높일 수도 있다.

현재 페이지 맵 시스템은 무선 단말기의 인터페이스 부분에 대해서 연구중이며 구역의 직접 선택을 통해 페이지의 탐색이 가능하게 될 것이다. 또한 여기에 부적절한 서버 또는 프록시의 클라이언트 정보의 접근과 가로 채기(snatching)를 막기 위해서 재구성 시스템에 보안 프로토콜을 정의해서 추가할 계획이다.

참 고 문 헌

[1] T. Kamba, S. A. Elson, T. Harpold, T. Stamper, and P. Sukaviriya, "Using small screen space more efficiently," Proceedings of CHI '96, ACM Press, 1996.

[2] J. Trevor, D. M. Hilbert, B. N. Schilit, and T. K. Koh, "From Desktop to Phonetop : A UI For Web Interaction On Very Small Devices," ACM Symposium on User Interface Software and Technology, 2001.

[3] O. Buyukkokten, H. Garcia-Molina, and A. Paepcke, T. Winograd, "Power Browser : Efficient Web Browsing for PDAs," In Proceedings of CHI'2000, ACM Press, Amsterdam, 2000.

[4] M. Hori, G. Kondoh, K. Ono, S. Hirose, and S. Singhal, "Annotation-Based Web Content Transcoding," Proceedings of the 9th International World Wide Web Conference, 2000.

[5] A. Fox, E. A. Brewer, "Reducing WWW Latency and Bandwidth Requirements by Real-Time Distillation," Proceedings of the 5th International World Wide Web Conference, Paris, France, 1996.

[6] D. Song and E. Hwang, "PageMap : Summarizing Web Pages for Small Display Devices," *Proc. of Int'l Conf. on* Internet Computing, Las Vegas, June, 2002.

[7] Wap Forum, White paper (Wireless Internet Today Overview), June, 2000, http://www.wapforum.org.

[8] S. Bjrk, L. E. Holmquist, J. Redstrm, I. Bretan, R. Danielsson, J. Karlgren, and K. Franzn, "WEST : A Web Browser for Small Terminals," ACM Symposium on User Interface Software and Technology, 1999.

[9] T. W. Bickmore, B. N. Schilit, "Digestor : Device-independent Access to the World Wide Web," Proceedings of the 6th International World Wide Web Conference, Santa Clara, CA, 1997.

[10] B. B. Bederson, J. D. Hollan, "Pad++ : A Zooming Graphical Interface for Exploring Alternate Interface Physics," ACM Symposium on User Interface Software and Technology, 1994.

[11] B. Fisher, M. Agelidis, J. Dill, P. Tan, G. Collaud, and C. Jones, "CZWeb : Fish-eye Views for Visualizing the World-Wide Web," Proceedings of the 6th International World Wide Web Conference, 1997.

[12] M. Jones, G. Marsden, N. Mohd-Nasir, K. Boone, and G. Buchanam, "Improving Web Interaction on Small Displays," Proceedings of the 8th nternational World Wide Web Conference, Toronto, Canada, 1999.

[13] K. Oh and E. Hwang, "Automatically Generating XML Documents from Web Data with Similar Pattern," to appear in the Journal of Computer and Information Science, Sep., 2002.

[14] A. Marcus, J. V. Ferrante, T. Kinnunen, K. Kuutti and E. Sparre, "Baby Faces : User-Interface Design for Small Displays," Proceedings of CHI '98, ACM Press, pp.96-97, 1988.

[15] K. L. Jones, "NIF-T-NAV : A hierachical navigator for WWW pages," Proceedings of the 5th International World Wide Web Conference, 1996.

[16] D. Nation, C. Plaisant, G. Marchionini and A. Komlodi, "Visualizing Web Sites using a Hierarchical Table of Contents Browser : WebToc," Proceedings of the 3rd Conference, Human Factors and the Web. 1997.

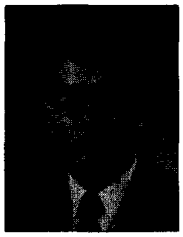
송 동 리

e-mail : harley@madang.ajou.ac.kr

2001년 아주대학교 정보 및 컴퓨터공학과 (학사)

2001년~현재 아주대학교 정보통신전문대학원 석사과정

관심분야 : 데이터베이스, 무선 인터넷, 홈네트워킹, XML 응용



황 인 준

e-mail : ehwang@madang.ajou.ac.kr

1988년 서울대학교 컴퓨터공학과(학사)

1990년 서울대학교 컴퓨터공학과(석사)

1998년 Univ. of Maryland at College Park 전산학과(박사)

1998년~1999년 Bowie State Univ., Assistant Professor

1999년~1999년 Hughes Research Lab. 연구교수

1999년~현재 아주대학교 정보통신전문대학원 조교수

관심분야 : 데이터베이스, 멀티미디어 시스템, 정보 통합, 전자상거래, XML응용