제 18 회 한국정보처리학회 추계학술발표대회 논문집 제 9 권 제 2 호 (2002.11)

이동 단말을 위한 HTML 문서의 변환 기법

신희숙, 조수선, 이동우, 마평수
한국전자통신연구원
컴퓨터소프트웨어 연구소
e-mail : hsshin8@etri.re.kr

A HTML Document Transcoding Technique for Mobile Devices

Hee-Sook Shin, Soo-Sun Cho, Dong-Woo Lee, Pyeong-Soo Mah
Lab. of Computer Software, ETRI

요 약

본 논문에서는 일반 데스크탑 PC 의 디스플레이 성능에 적합하도록 작성된 유선의 웹 문서를 무선 인터넷 환경의 핸드헬드 계열 소형 단말에서도 효율적으로 표현하기 위한 변환 기법을 제시한다. 이는 기존의 단순한 텍스트 위주의 추출 및 요약 형식의 변환과는 달리, 분석 및 변환을 위한 최소 내용 단위를 설정하고, 이들의 재배치를 통하여 원본 웹 문서의 정보를 보다 정확히 반영한다. 또한 새로운 인덱스 형식으로의 재표현을 통하여 기존의 페이지 조각과 계층적 구조의 인덱스 링크를 이용한 인터페이스보다 편리한 검색 및 페이지 이동을 제공한다. 이 기법은 보다 많은 정보를 복잡한 구조로 표현하는 현재의 웹 문서 특징을 반영하고, 이동 단말들의 고성능화 추세와 함께 화려한 무선 인터넷을 요구하는 사용자들을 고려한 것이다. 전체 변환 과정은 Layout-Forming Tag Analysis Algorithm, Component Grouping Algorithm, Component Block Classification Method, Index Generation Method 로 구성된다. 변환 시스템의 구성 모듈별 설계와 프로토타입의 구현을 통하여 제안하는 알고리즘 및 변환 방법을 평가하였고, 실제 웹 문서에 대한 실험 결과에서 단말의 소형 화면에 적합하게 변환된 모습을 확인하였다.

1. 서론

언제 어디서든지 웹을 사용하고자 하는 사람들의 욕구는 무선 인터넷 환경을 창출하였고, 1990 년대 중반 이후 소형 단말기 시장의 성장은 무선 인터넷의 접속을 증가시켰다[1][2].

하지만 현재의 웹은 그 자체가 유선 환경의 인터넷을 위해 설계되었고, 기존의 유선 웹 상에 존재하는 수많은 정보 또한 데스크탑 환경의 사용자를 위해 제작되어진 것이다. 따라서 이를 소형 화면의 단말을 통하여 브라우징 할 경우, 단말에서 제대로 표현하지 못하는 문제가 발생하게 된다.

이러한 문제를 해결하고자 많은 연구가 선행되었으나, 간단한 텍스트 형식의 추출 및 요약으로의 변환 또는 수작업을 동반한 변환 방식을 사용하는 경우가 대부분이다[3][5][8][11]. 게다가 자동 변환 기법들은, 복잡한 구조에서 많은 정보를 표현하는 현재의 웹 문서의 특징과 보다 화려한 웹 문서의 모습을 이동 단말을 통하여 보기를 원하는 사용자들의 욕구를 만족시키기에는 어려운 점이 있다.

따라서 본 논문에서는 내용 블록 단위와 Semi-Semantic 정보를 이용하여 복잡한 웹 문서에 대해 보다 정확한 변환을 시도하고, 원본 웹 문서의 모습을 최대한 반영하면서 좌우스크롤이 없는 편리한 인터페이스로 소형 화면에 적합하게 표현되어지는 자동 변환 기법을 제안한다.

본 논문의 구성은 다음과 같다. 1 장의 서론에 이어서 2 장에서는 웹 문서의 변환과 관련된 기존의 연구에 대해 살펴보고, 3 장에서는 제시하고자 하는 웹 문서 변환 기법에 대해 구체적으로 설명한다. 4 장에서는 변환 시스템의 구현 및 제시하는 기법의 성능 평가에 대해 다룬다. 마지막으로 5 장에서는 향후 연구 과제에 대한 언급과 전체 논문 내용의 요약으로 결론을 맺는다.

2. 관련 연구

무선 환경에서 웹 접속을 시도하는 단말의 유형은 크게 세 가지로 구분되어진다[1].

첫째는, 노트북 계열로 기존의 데스크탑 PC 환경과 유사한 단말들이다. 최소 800x600 이상의 해상도를 가지므로 소형 화면을 위한 웹 문서 변환이 필요치 않다.

둘째로 셀룰러 폰 계열의 단말들을 들 수 있다. 이는 90x60 정도의 해상도와 20 줄 정도의 텍스트 위주로 된 제한된 정보의 표현만이 가능하므로 대부분이 WML, HDML, cHTML 등의 마크업 언어를 통하여 정보를 표현한다. 따라서 자동 변환 기법보다는 수작업을 동반한 변환 기법이 보다 정확하게 동작하는 경향을 보인다.

세째는 핸드헬드 계열로 Palm-Size PC, Hand-Held PC 등 일반적으로 PDA(개인정보단말기)로 통칭되는 단말들이다. 이들은 보통 3-5 인치 크기의 화면으로 320x240 정도의 해상도를 지원하나, 최근에는 화면 크기와 해상도가 높아지는 경향을 보이면서 640x480 의 해상도를 지원하기도 한다. 핸드헬드 계열 단말의 고성능화 추세와 이를 이용한 무선 인터넷의 접속이 증가하면서 이런 단말을 위한 웹 문서의 적절한 변환이 요구되어지고 있다. 따라서 본 연구에서는 HTML 브라우저를 탑재한 핸드헬드 계열의 이동 단말을 대상으로 하는 웹 문서의 변환 기법에 대해 다루고자 한다. 여기서 이와 관련된 기존 연구들을 웹 문서가 변환되는 단계 또는 시점을 기준으로 하여 분류하면 다음과 같다.

첫째, 웹 서버 측에서의 변환: Non-Automatic 기법으로, 여러 단말을 지원하기 위해 별도로 제작된 문서를 가지거나 또는 별도의 표현 기법에 대한 정의를 단말별로 이미 가지고 있는 경우이다. 실시간 자동 변환 기법을 이용할 수도 있으나, 대부분이 수작업을 동반하는 변환 툴 형식으로 지원되고, 따라서 가장 정확한 변환이 이루어지는 장점을 가진다. 하지만 서버측에서의 변환은 제한된 웹 문서에 한해서 변환 서비스가 제공되는 단점이 있다. IBM 의 WebSphere Transcoding Publisher[11]가 대표적인 예이다.

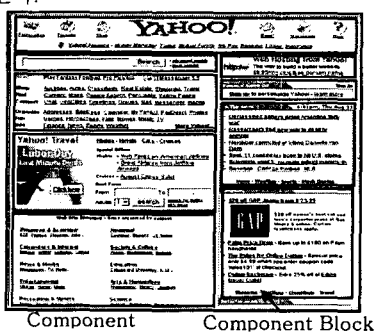
둘째, 클라이언트 측에서의 변환: 웹 문서를 전송받은 클라이언트 측에서 적절한 변환을 수행하는 것으로 사용자의 요구 사항을 반영한 개인 특성화가 쉽게 구현될 수 있는 장점이 있다. 하지만 유선과 동일한 형태의 정보를 클라이언트가 받은 후에 적절한 변환 과정을 수행하므로 네트웍 자원의 비효율적인 사용과 클라이언트 단말의 높은 컴퓨팅 파워를 요구하게 된다. 대표적인 예로는 단말에서 특정 부위의 줌인/줌아웃 인터페이스를 제공하는 SmartView[6], Pad++[7] 등이 있다.

세째, 프락시 측에서의 변환: 대부분의 자동 변환 시스템은 프락시로 동작하면서 보다 많은 웹 문서를 대상으로 적절한 변환을 수행하여 다양한 단말을 지원한다[9][10]. 관련 연구로는 팜파일럿 단말의 브라우저를 위한 변환 프락시를 선보이는 Top Gun Wingman[8]과, 상용 제품으로 자동 변환을 수행하는 Sypglass Prism 등이 있다. Top Gun Wingman 의 경우 브라우저에서의 변환도 부분적으로 수행되기도 하나 대부분의 변환은 프락시에 의해서 주도되고, Prism 의 경우는 변환 툴도 함께 제공하고 있다. 대표적인 프락시 기반 자동 변환 시스템으로 Digester[3], WebAlchemist[4]가 있다. 이들은 실험적 경험을 통해 얻은 다양한 휴리스틱 변환 기법과 이들의 적절한 적용 규칙을 제시한다. 특히 WebAlchemist[4]의 경우는 기존의 자동 변환 기법이 복잡한 웹 문서에 대하여 정확성이 떨어지게 변환되는 점을 보완하고자 Semi-Semantic 정보를 변환에 활용하고 있다. 하지만 이 기법도 텍스트의 추출 및 요약 또는 페이지 조각 나눔과 이들을 연결하는 링크의 생성 등의 기법을 주로 사용함으로써 복잡한 웹 문서에 대해 원본 문서의 의미를 명확하게 전달하기에는 어느 정도의 한계가 나타난다.

3. 이동 단말을 위한 HTML 문서의 변환 기법

앞선 연구에서 나타난 문제점을 보완하면서, HTML 브라우저를 내장한 핸드헬드 계열의 소형 단말을 위한 새로운 변환 기법을 제안한다.

웹 문서 특징 선행 조사에서 대부분의 웹 문서는 명확한 내용 전달을 위해서 의미상의 차이를 가지는 컨텐츠를 레이아웃 및 구조적 태그를 사용하여 시각적인 분리가 이루어지도록 구성함을 알 수 있었다[5]. 따라서 이러한 특성을 기반으로 하여, 시각적 분리 표현을 주도하는 태그들의 분석으로부터 페이지 레이아웃의 Semi-Semantic 정보를 추출하고 이를 변환에 활용한다. 이것은 WebAlchemist[4]에서 제시한 바와 같이 기존의 Syntactic 자동 변환 기법이 가지는 불명확한 변환을 보완할 수 있다. 또한 이러한 시각적인 내용 조각 단위로부터 분석을 위한 최소 내용 단위(*Component*)를 설정하고 이를 기본으로 변환을 수행함으로써 정보 전달이 보다 명확한 변환 결과를 얻을 수 있다. 여기에 단말의 화면 크기를 고려하여 조각 단위들의 묶음(Component Block)을 재배치하고, 새로운 인덱스 형식의 표현을 통하여 편리한 인터페이스로 좌우스크롤이 없는 웹 페이지를 제공할 수 있다. 다음의 (그림 1)은 웹 페이지의 시각적인 분리 표현의 예를 나타낸다.

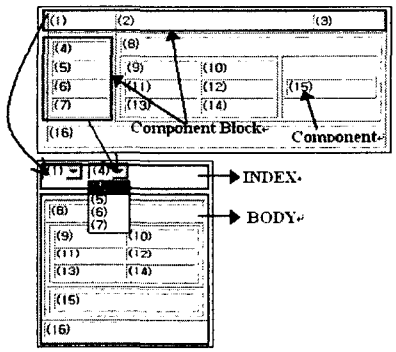


(그림 1) 웹 페이지의 시각적인 분리 표현 예

변환 과정은 다음의 두 가지 분석 알고리즘과 이후

의 재표현 및 재배치 기법을 통해서 수행된다.

먼저 Layout-Forming Tag Analysis Algorithm 을 이용하여 원본 웹 문서에서 Component 단위를 정의하고 분석에 필요한 구조적 정보를 얻는다. 이후 Component Grouping Algorithm 을 통하여 정의된 Component 를 클라이언트 화면 성능에 따라 적절한 width 를 가지도록 Component Block 으로 묶는데, 이때 각 Block 은 표현 및 배치상의 유사성을 기본으로 가지게 된다. 추출된 Component Block 은 포함하는 컨텐츠의 특성에 따라 INDEX 또는 BODY 형으로 분류되고, 이는 새로운 선택 리스트 형식으로 재표현 되어지거나 또는 테이블 블록 단위로 재배치되어진다. 다음의 (그림 2)는 간단한 HTML 예제를 통한 변환 전과 후의 모습을 나타낸다.



(그림 2) 변환 전(상)과 후(하) 모습의 예제

(그림 3)과 (그림 4)는 Layout-Forming Tag Analysis Algorithm 과 Component Grouping Algorithm 을 차례대로 기술한 것이다.

```
input tag node tree;
REPEAT {
  extract next tag node in preorder traversal order;
  IF (tag == <TABLE>) {
    IF (TableDepth > threshold & Width <= MAX_WIDTH) {
      define Width of all elements inside <TABLE>;
    }ELSE {
      increase TableDepth;
      define Width of <TABLE> element;
    }
  }ELSE IF (tag == <TR>) {
    IF (tag is not in the first row of nested table)
      increase RowNum;
    IF (tag's parent is root table) ColNum = 0;
    define Width of <TR> element;
  }ELSE IF (tag == <TD>) {
    IF (tag is not in the first row of nested table)
      increase ColNum;
    IF (tag has <TABLE> as child node) define CompNum
      with (0, child<TABLE>.first<TD>.CompNum,
      child<TABLE>.last<TD>.CompNum);
    ELSE define CompNum with (sequence number, 0, 0);
    define Width of <TD> element;
  }ELSE IF (tag == <IMG>) {
    convert image format;
    set Width of image;
    IF(<MAP> is used for the image)
      modify COORDS attribute value of <AREA>;
    }ELSE IF (other tags) trivial functions for other tags;
}UNTIL (end-of-tag of HTML)
```

(그림 3) Layout-Forming Tag Analysis Algorithm

이 알고리즘을 통하여 <표 1>의 구조적 Semi-Semantic 정보를 추출할 수 있으며, 이 정보는 이후의 변환 과정에서 이용된다.

<표 1> Parameters of Structural Information

Parameter	Description
CompNum	Component ID General Component has (sequence number, 0, 0). Inclusive Component, which includes <TABLE> tag, has (0,first figure of first child's CompNum, first figure of last child's CompNum).
RowNum	Row number in total layout
ColNum	Column number in total layout
TableDepth	Number of ancestor <TABLE>
Width	Re-calculated width value using pixel unit

```
input Component node tree;
REPEAT {
  extract next Component node in preorder traversal order;
  IF (Component has sibling nodes) {
    make Component group with the sibling nodes;
  }
  make table block with Component group/Component;
  IF (table block has last <TD> & parent <TABLE> of current
    Component is nested table & first ancestor <TD> of the
    Component is inclusive Component) {
    make table block for each two groups of child nodes;
    make table block with first ancestor <TD>;
  }
  rearrange Component Block;
}UNTIL(end-of-Component node)
```

(그림 4) Component Grouping Algorithm

위의 알고리즘을 통하여 추출된 Component Block 은 자신이 포함하는 컨텐츠의 패턴에 따라서 INDEX 형과 BODY 형으로 분류된다. 이러한 분류 과정은 Text Length, Image Width, Link Number, Row Number, Column Number 등과 같은 변수들의 패턴 비교를 통하여 수행된다.

분류 과정 이후, INDEX 형은 새로운 HTML 문서의 상단에 메뉴 선택을 위한 인터페이스로 재표현 되어지고, BODY 형은 <BODY> 안에 그대로 재배치되어진다. 이 때 INDEX 형의 경우는 스크립트 파일과 <SELECT>를 생성하여 각 인덱스 값이 option 속성값으로 매핑되도록 구현할 수 있다.

이상의 과정을 통하여 본 논문에서 제시하는 소형 화면의 단말을 위한 웹 문서의 변환 과정이 수행되고, 제안하는 기법의 특징을 요약하면 다음과 같다.

- Web page analysis based on visual separation
- Definition of minimum unit for transcoding
- Using semi-semantic information of page layout
- Grouping and rearrangement of Component
- Re-expression specific blocks in indexing format

4. 시스템의 구현 및 실험 평가

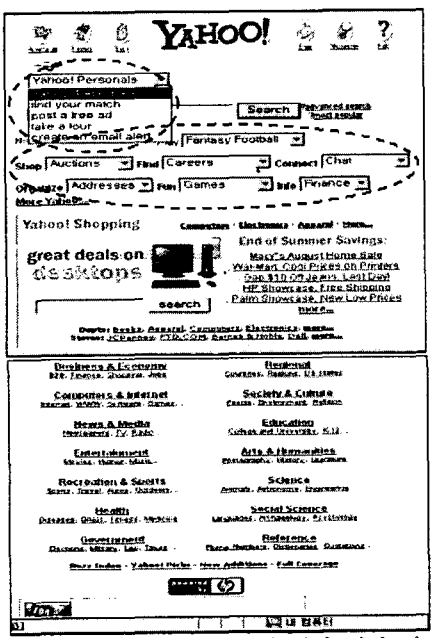
시스템의 구현 및 실험은 알고리즘의 실현과 성능 평가에 초점을 두었고, W3C HTML Tidy 와 Jigsaw 를

활용하여 HTML4.0 과 HTTP1.1 을 따르도록 프로토타입을 구현하였다. 한국 인터넷 정보센터의 웹사이트 분석/평가 전문 사이트에서 제시한 순위별 50 개의 사이트를 선정하여 테스트하였고, <표 2>에서 그 결과를 요약한다.

<표 2> 알고리즘의 테스트 결과 요약 및 분석

비교 변수	백분율
좌우스크롤의 발생율	0% (해당 페이지 수 / 전체 테스트페이지 수)
문서 내용의 유지율	89% (해당 Component 수 / 전체 Component 수)
문서 구조상의 오류 발생율	14% (해당 테이블 블록 수 / 전체 테이블 블록 수)
에러 발생율	9% (해당 페이지 수 / 전체 페이지 수)

문서 내용의 유지율은 원본 페이지와 변환된 페이지가 가지는 컨텐츠의 차이점에 대한 비율을 나타내는 값으로 Component 단위에서 컨텐츠 유무를 비교 기준으로 하였다. 문서 구조상의 오류 발생율은 새로운 구조적 표현 즉, <TABLE>의 시각적인 표현이 적절하지 못한 경우를 나타내는 것으로 내부 <TR>, <TD>의 속성값과 매칭 여부를 비교하였다. 에러 발생율은 스크립트 에러 및 객체 랜더링에서 발생하는 에러의 횟수로 계산한 값이다. 결과적으로, 구조적 구성을 가지는 웹 문서에 대해 최적으로 동작하여 좌우 스크롤없이 내용 단위별 재표현 및 재배치가 이루어졌으며, 특히 많은 정보를 표현하면서 복잡하게 구성된 페이지에 대해 적절한 변환 결과를 보였다. 다음 그림은 Yahoo 사이트에 대한 변환 결과 예제를 보인다.



(그림 5) Yahoo.com 사이트의 변환 결과 예제

5. 결론

본 논문에서는 기존의 일반 데스크탑 PC 의 디스플레이 성능에 적합하도록 작성된 웹 문서를 소형 화면에서도 효율적으로 표현되어질 수 있도록 변환해 주기 위한 새로운 기법을 제시하였다.

웹 문서의 분석 및 변환을 위한 주요 알고리즘으로 Layout-Forming Tag Analysis Algorithm 과 Component Grouping Algorithm 을 소개하였고, Component Block 의 분류 및 인덱스 생성과 블록 단위의 재배치 방법에 대해 제안하였다.

본 논문이 제시하는 변환 기법을 통하여 기존의 웹 문서는 이동 단말의 화면 크기에 적합한 모습으로 변환되어 좌우 스크롤없이 브라우징할 수 있고, 원 문서가 가지는 정보를 명확하게 전달하는 효과도 얻을 수 있다. 이는 이동 환경의 고성능 소형 단말의 특성을 최대한 고려하여 사용자에게 편리하고 효과적으로 웹 서비스를 사용할 수 있도록 한다.

시스템의 설계와 프로토타입의 구현을 통하여 알고리즘의 성능을 평가하였고, 테스트 결과 페이지를 함께 제시하였다. 향후 과제로 시각과 청각이 동시에 지원되는 웹 문서 형식으로의 변환 기법과 실시간 자동 변환을 위한 최적 기법에 대해 계속 연구하고자 한다.

참고문헌

[1] C.G.Park, Y.C.Lee, "Trend of Mobile Devices", ETRI Weekly Technology Trend, 2001, vol.1027.

[2] C.W.Bae, "Trend of Information and Communication Industry: ch.7 PDA", KISDI, 2001.

[3] T.Bickmore, A.Girgensohn and J.W.Sullivan, "Web Page Filtering and Re-Authoring for Mobile Users", The Computer Journal, vol.42, no.6, 1999, pp.534-546.

[4] Y.H.Whang, C.H.Jung, J.H.Kim and S.K.Chung, "WebAlchemist: A Web Trtanscoding System for Mobile Web Access in Handheld Devices", SPIE ITCOM 2001, Aug. 2001.

[5] Y.D.Yang and H.J.Zhang, "HTML Page Analysis Based on Visual Clues", IEEE ICDAR 2001, Sept. 2001, pp.859-864.

[6] N.Milic-Frayling and R.Sommerer, "SmartView: Flexible Viewing of Web Page Contents", World Wide Web Conference 2002, 2002.

[7] B.Bederson and J.Hollan, "Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics", ACM User Interface Software and Technology, 1994, pp.17-26.

[8] E.Brewer, A.Fox, I.Goldberg, D.Lee and A.Polito, "Experience with Top Gun Wingman: A Proxy-Based Graphical Web Browser for the 3Com PalmPilot", IFIP Middleware'98, pp.407-424.

[9] B.Zenel, "A General Purpose Proxy Filitering Mechanism Applied to the Mobile Environment", Wireless Networks Journal, vol.5, 1999, pp.391-409.

[10] A.Joshi, "On Proxy Agents, Mobility, and Web Access", Mobile Networks and Applications Journal, vol.5, 2000, pp.233-241.

[11] IBM, WebSphere Transcoding Publisher, http://www-3.ibm.com/software/webservers/transcoding/index.html.