

웹에서 효율적인 텍스트 문서 브라우징

김원중 조이기 손철수^U

순천대학교 컴퓨터학과

{kwj.lgcho.mrb}@cs.sunchon.ac.kr

The Efficient Text Documents Browsing on World Wide Web

Won-Jung Kim Lee-Gi Cho Cheol-Su Son^U
Dept. of Computer Science, Sunchon National University

요 약

웹의 등장으로 우리는 현재 멀티미디어 정보까지도 쉽게 이용할 수 있게 되었지만, 웹에서의 텍스트 문서정보의 브라우징(Browsing)은 많은 마우스 클릭과 스크롤링을 필요로 하고 있다. 또한 현재의 웹 텍스트 문서의 브라우징은 문서의 내용이 클 경우 검색자가 현재 브라우징하고 있는 문서가 본인이 필요로 하는 것인지, 아닌지를 판단하는데 많은 시간과 노력을 요구한다. 본 논문에서는 문서의 제목 부분과 내용 단락의 전체가 아닌 1-2줄만으로도 문서의 전체 윤곽을 쉽게 파악할 수 있다는데 착안하여 웹에서의 효율적인 텍스트 문서 브라우징 시스템을 개발하였다.

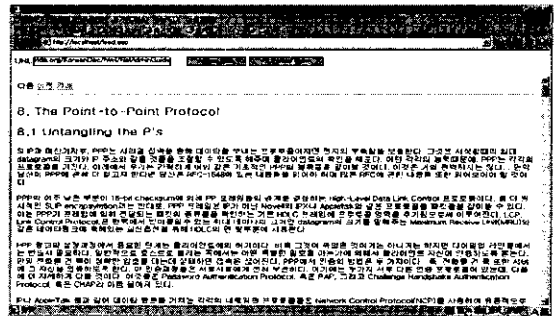
1. 서론

인간은 종류와 분량은 다르지만 끊임없이 자신의 직업 및 생활에 필요한 정보를 습득하면서 살아간다. 이러한 정보습득의 대상과 방법은 시대에 따라 많은 변화를 겪어왔다. 농경사회에서는 인간과 인간 즉, 구전을 통하여 생활에 필요한 지식을 습득하였고, 산업사회에 접어들면서 인쇄매체와 방송매체가 이용되었다.

산업사회에서의 정보습득방법은 지식전달에 있어서 세대간의 단절을 해결할 수 있었으나 지리적, 공간적, 시간적 문제는 해결할 수 없었다.

컴퓨터와 통신의 발달과 함께 시작된 정보화시대에서는 정보의 분배와 획득수단이 획기적으로 변화하였다. 대용량의 컴퓨터 저장장치에 정보를 저장하고, 고속의 통신망을 통하여 서로 주고 받음으로써 이러한 문제점들을 해결할 수 있었다. 90년대 초에 웹의 등장으로 현재 우리는 멀티미디어 정보까지도 손쉽게 이용할 수 있게 되었지만, 연구자료, 논문정보, 연구보고서, 일반도서 등과 같은 텍스트 문서(Documents) 정보를 효율적으로 브라우징할 수 있기를 바라고 있다. 그러나 현재의 웹 브라우저는 우리가 책을 읽는 것처럼 텍스트 문서의 전체 내용을 쉽게 파악하기가 매우 어렵다. 따라서 웹 브라우저에서 텍스트 문서 정보를 효율적으로 브라우징할 수 있는 방법에 대한 연구가 필요하다. 현재 웹에서 텍스트의 KLDP(Korean Linux Document Project) 사이트와 형태의 문서 정보를 제공하는 사이트들은 대부분 [그림1]

같은 형태의 서비스를 제공하고 있다. 화면 상단의 '다음', '이전', '차례'는 웹 브라우저에서 사용되는 툴바와 비슷



[그림1] 기존 웹 문서 브라우징 화면

한 기능을 제공하는 것으로서 다음 페이지, 이전 페이지 그리고 문서의 목차 페이지로 이동할 수 있도록 한다. 문서의 내용들은 일반적으로 대제목, 중제목, 소제목, 많은 라인의 내용 단락 형태로 표현된다. 화면 오른쪽에 있는 스크롤 바의 위치에 의해 유추하면 절반 정도의 페이지가 사용자에게 제공되어져 있다. 만약 사용자가 화면 하단 부분의 내용을 확인하기 위해서는 옆의 스크롤 바를 아래로 이동시켜야 하는 불편함을 가지게 된다. 또한 페이지의 내용이 많아지면 사용자가 화면을 스크롤

하는 횟수와 시간이 증가하게 된다. 이러한 방법은 문서 검색자가 브라우징하고 있는 문서가 본인이 찾고 있는 정보인지 아닌지를 판단하는데 많은 시간과 노력을 요구한다. 본 논문에서는 문서에서의 한 페이지가 웹 브라우저의 한 화면에 모두 나타나도록 하여 검색자가 손쉽게 전체 내용을 파악할 수 있도록 하는 효율적인 텍스트 정보 브라우징 방법에 대해 연구하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해 살펴보고, 3장에서는 Web Document Outliner의 개념과 구현에 대해 설명하였다. 4장에서 결론과 향후 연구과제에 대해 논하였다.

2. 관련연구

웹에서 텍스트 문서의 내용을 효과적으로 파악하기 위하여 사용되는 기법으로는 Zooming 기법, Focus+Context 기법, Focus+Context Display 기법 등이 있다[1,4].

(1) Zooming 기법

Zooming 기법은 Pad++ 브라우저 처음의 본문에서 사용자가 원하는 정보가 나타날 때, 그 선택되어진 정보를 화면의 앞쪽으로 확대시킴과 동시에 처음의 본문을 화면의 뒤편으로 축소시키는 형태로 표현한다.

한번에 한 페이지가 보이는 것이 아니라, 여러 개의 페이지와 그 페이지 사이의 링크들이 커다란 줄 정보 창에 나타난다. 페이지는 컨텍스트(Context)를 제공하는 조그만 화면에 보이는 연결 페이지와 함께 보고자 하는 페이지가 명확히 보이도록 크기가 조절된다. 글꼴의 변화는 애니메이션화 되어서 포커스를 둔 페이지가 디스플레이 화면 중앙으로 움직인다. 그러는 동안 부수적인 정보가 있는 페이지들은 작아진다.

줌 브라우저는 현재의 HTML 페이지를 다양한 크기로 보여지도록 지원한다. 또한 한 페이지 안에 크기가 다양한 레이아웃을 지원하는 확장된 HTML을 보유하고 있다. 이러한 확장(MSML, 다중 규격 표시 언어)은 개발의 초기단계에 있다. 지금 현재로서는 우리의 브라우저는 내장된 Pad++ 언어로부터 한 페이지에 보이도록 지원한다. 이것은 고객 편에서 다양한 반응을 보일 수 있도록 해주며, 그것이 페이지에 첨가되도록 하여, 페이지 구성이 독립된 그래픽 객체처럼 쓰이도록 한다.

또한 Holmquist와 Ahlberg는 정보의 양이 많은 경우에 전체 문맥을 파악하면서 동시에 특정부분을 자세히 알고 싶은 경우 임의접근이 가능한 Flip Zooming 기법을 제안하였다. 그러나 이 기법에서는 문자 정보와 그림 정보를 같은 화면에 표현하지 못하는 단점이 있다.

(2) Focus+Context 기법

Focus+Context 기법은 전체 정보(Context)의 개략적인 내용을 한 눈에 파악하면서 동시에 사용자가 원하는 부분(Focus)을 자세히 볼 수 있도록 하는 것이다. 이 기법은 긴 문서에서 자신이 원하는 특정부분의 검색시간을 줄여주면서 동시에 페이지 단위로 분할 표현함으로써 리스트 형태의 문서표현이 가져오는 지루함을 극복했다. 또한 찾고자 하는 특정부분으로 건너뛰는 것을 보다 더 쉽게 한다.

검색엔진을 이용하여 검색된 문서의 내용을 보다 효과적으로 시각화하기 위해서 Focus+Context 기법을 이용한

줌 브라우저(Zoom Browser)를 Java로 구현하였다. Java에서 제공하는 Socket을 사용하여 클라이언트/서버 환경을 구축하였고, 서버에서 일반 검색엔진들의 검색결과를 받아온다. 줌 브라우저에서는 검색된 결과들을 페이지 단위로 나누어서 전체정보의 양을 한 화면에 표현하도록 하여 친절한 사용자 인터페이스를 제공하고, 사용자가 읽기 쉬운 크기로 확대된다.

(3) Focus+Context Display 기법

Brown 등은 HTML의 헤드 태그에 지퍼(zipper)라 불리는 아이콘을 첨부하고, 사용자가 이 지퍼를 클릭 함으로써 문서의 각 부분을 볼 수 있도록 구현했다. 지퍼를 구현하기 위한 세 가지 방법을 제시하고 있는데, 첫 번째 방법은 지퍼를 삽입한 custom-built 웹 브라우저를 사용하는 것이고, 두 번째 방법은 지퍼를 웹 프록시(proxy)에 의해서 HTML문서에 삽입하고, 지퍼의 변환된 상태를 반영해주는 수정된 HTML을 생성한다. 마지막으로 세 번째 방법은 Java script를 이용하여 지퍼를 구현하는 것이다. 이 기법은 복잡한 계층구조를 갖는 문서에서는 효과적이거나 한 섹션(section)이 긴 문서의 경우에는 적합하지 못하다.

3. Web Document Outliner

본 논문에서는 웹 브라우저에서 텍스트 문서의 내용 단락을 부분용 1-2줄만 보이도록 축소하여 문서의 전체 페이지가 브라우저의 한 화면에 나타나도록 한다.

KLDP(Korean Linux Document Project)와 같은 문서 정보 제공 사이트들은 서론에서 언급한 것처럼 텍스트 위주의 많은 단락을 가진 웹 페이지를 구성한다. 그리고 각 문서들은 문서 내용 단락을 대표하는 소제목과 가지고 있는 것이 대부분이다. 이 소제목은 그 단락에 대한 내용의 핵심어가 대부분이기 때문에 그 소제목만을 보고서도 단락에서 주고자 하는 정보를 요약할 수 있다. 이에 착안하여 이 시스템을 개발하게 되었다.

3.1 구현

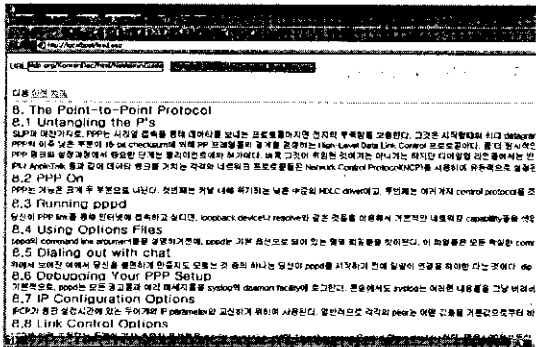
본 논문에서 설계 구현한 시스템은 텍스트로 구성되어 있는 Browser의 Width를 최대한 늘임으로써 각 문서의 내용 단락이 1-2라인에 표현되도록 한다. 그 결과 웹 문서 페이지의 구성은 소제목과 그에 해당하는 내용 단락의 시작부분이 하나의 그룹으로 묶이게 된다. 따라서 문서 검색자들은 문서의 한 페이지에 표현된 전체 소제목들과 1-2라인의 내용 단락을 읽음으로써 해당 페이지의 전체 간략 정보를 쉽게 파악할 수 있게 된다.

[표1]은 본 시스템에서 구현한 자바 스크립트의 구조이다. 구현 시스템은 Netscape 5.x, Explorer 5.x 버전에서 작동되도록 구현하였다.

3.2 구현 화면

[그림2]는 구현 시스템을 앞의 [그림1]의 웹 문서에 적용한 결과 화면이다. 기존의 웹 문서 브라우징에서는 한 화면에 하나의 대체목과 소제목 그리고 내용단락을 볼 수 있었지만, 구현 시스템에서는 하나의 대체목과 8개의 소제목 및 내용 단락의 시작 부분을 볼 수 있어서 검색자는 단 한번의 마우스 클릭으로 쉽게 전체 내용의 윤곽을 파악할 수 있다.

구현된 시스템은 특정한 프로그램을 사용하지 않고도 웹 브라우저에서 곧 바로 해석되는 자바 스크립트를 사용함으로써 [그림1]와 [그림2]의 차이처럼 HTML Code 중에 Browser의 Width 값을 10,000으로 준다면 단순히 Browser의 Width의 크기를 늘리면 되는 것이므로 시스템의 성능이 우수하다. 본 논문에서 구현한 시스템은 웹 문서에 추가시킬 수 있는 코드를 제공하며, 웹마스터들이 문서에 추가를 시킴으로써 적용할 수 있다.



[그림2] 구현 시스템에서 웹 문서 브라우징

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="javascript">
<!--
function LinedView() {
  if (document.OUT.Line.value=="OutLined View") {
    document.OUT.Line.value="UnOutLined View";
    OutLine();
  } else {
    if (document.OUT.Line.value=="UnOutLined View") {
      document.OUT.Line.value="OutLined View";
      UnOutLine();
    }
  }
}
function OutLine() {
  window.top.resizeTo (10000, screen.availHeight);
}
function UnOutLine() {
  window.top.resizeTo
  (screen.availWidth, screen.availHeight);
}
function Action_URL() {
  document.OUT.target = "Temp";
  document.OUT.action="Sample.asp";
}
-->
</SCRIPT>
</HTML>
```

Continue-

-Continue

```
document.OUT.submit();
}
--></SCRIPT>
</HEAD>
<BODY>
<TABLE>
<FORM NAME="OUT" METHOD="POST">
<TR><TD>
<INPUT TYPE="TEXT" NAME="URL_VALUE" VALUE=""
SIZE="35">
</TD><TD>
<INPUT TYPE="BUTTON" NAME="AURL"
VALUE="Action_URL" OnClick="javascript:Action_URL()">
</TD>
<TD>
<INPUT TYPE="BUTTON" NAME="Line" VALUE="OutLined
View" OnClick="javascript:LinedView()">
</TD>
</TR>
</FORM></TABLE>
</BODY></HTML>
```

[표1] 구현 자바스크립트 구조

4. 결론

본 논문에서 구현한 시스템은 연구보고서, 논문, 일반문서 등과 같은 텍스트 형태의 문서 제공 서비스를 제공하는 사이트에서 유용하게 사용할 수 있다. 특히, 앞으로 웹에서의 책의 출판이 일반화될 때, 효과적으로 웹 북(Web Book)의 내용을 파악하는데 크게 도움을 줄 수 있을 것이다. 향후 연구 내용으로는 웹 페이지의 내용 단락이 테이블 형태로 구성되어 있으며, Width가 "%"가 아닌 픽셀로 되어 있는 경우 브라우저의 Width를 키우더라도 효과를 볼 수 없는 단점을 해결하여야 한다.

5. 참고문헌

[1] Marc H. Brown, Hannes. Marais, Marc A. Najork, and William E. Weihl, Focus+Context Display of Web Pages: Implementation Alternatives, 1997., <http://gatekeeper.dec.com/pub/DEC/SRC/technical-notes/SRC-1997-010-html/>
 [2] Jui-Chieh Hsu, William J ohnston, John McCarthy, Active Outlining for HTML Documents: An X-Mosaic implementation, <http://www.ncsa.uiuc.edu/SDG/IT94/Proceeding/HCS/hsu/hsu.html>
 [3] S. Selvarajan, outliner software, <http://class.ee.iastate.edu/berleant/home/Research/DWD/Outliner>.
 [4] Benjamin B. Bederson, James D. Hollan, Jason Stewart, David Rogers, Allison Druin, David Vick, A Zomming Web Browser, <http://www.cs.umd.edu/hcil/pad+/papers/spie-96-webbrowser/spie-96-webbrowser.pdf>