웹 2.0 기반 동기식 웹 브라우저 화면 분할

이우기 ○ 윤선주 최중호 장용
인하대학교 산업공학과
kanonjoy@gmail.com, wookeylee@gmail.com, cjoongho@gmail.com, idbluel@gmail.com

Synchronous Web Browser Fragmentation with Web 2.0 Technologies

Wookey Lee○, SunJu Yoon, JungHo Choi, Yong Jang
*Dept. of Industrial Engineering, Inha University

요 약

인터넷을 통한 검색은 사용자가 찾으려는 정보를 일반적으로 브라우저를 통해 표현해주고 있다. 본 연구서는 웹 사용자에게 기존의 브라우저가 아닌 화면 분할을 통하여 인터넷 사용의 효율을 높이는 방식을 제공하였다. 그 중에 WEB 2.0의 Ajax를 사용하여 웹 페이지를 분할 후 특정 브라우저와 상관없이 복수 페이지의 접근을 가능하게 하고 검색의 효율성을 높이기 위해 하나의 브라우저에서 서로 다른 검색엔진의 Open API가 작동되도록 하였다. 한번의 키워드 입력과 한번의 클릭으로 검색엔진 마다 방문 할 필요 없이 하나의 브라우저에서 각각의 검색 결과를 사용자에게 제공해 주도록 하여 검색 시간을 감축하고, 결과를 비교 분석 하므로 웹 서핑의 효율성을 높여주는 시스템을 구현하였고 이를 입증하였다.

1. 서 론

웹(WWW)에서 가장 빠르게 발전된 것은 검색 서비스이며 사용자가 던지는 몇 가지의 키워드를 가지고 원하는 정보를 빠르고 간편하게 제공해주는 역할을 해주고 있다. 이러한 인터넷의 정보를 읽을 수 있도록 변환하는 역할을 브라우저가 담당하고 있다. 즉, 사용자는 대부분의 브라우저를 통해 웹에 접근하여 검색을 하고 정보를 얻는다. 브라우저는 웹 사용에 있어 중요한 기능 중에 하나이며 브라우저 기능도 확장되고 있으며 계속 새로운 웹 브라우저가 생겨나고 있다[1]. 한 때 웹 브라우저 시장은 넷스케이프 네비게이터의 독무대였다가, 얼마 가지 않아 Microsoft의 Internet Explorer(IE)가 시장의 지배자가 되었고, 지속적으로 새롭고 다양한 차세대 웹 브라우저들이 등장하고 있다. 새로운 브라우저들은 웹 서핑에 많은 도움을 준 북마크의 기능을 뛰어넘어 사용자의 기호에 따라 디자인으로 테마를 바꿀 수도 있고 웹 서핑에 도움을 주는 수많은 보조기능에서 간단한 게임까지 포함된 개인화된 웹 브라우저를 제공해 준다.

본 연구의 동기는 사용자가 검색을 했을 경우 하나의 브라우저에는 하나의 결과를 보기 보다는 여러 결과를 한번에 보면서 정보의 습득에 있어 시간을 절약 할 수 있다는 점에서 이와 관련하여 처음 선보인 탭 브라우저는 오페라(Opera)라는 웹 브라우저로 한가지 이상의 웹 페이지를 지나갈 때 하나의 창으로 보여주는 편리한 기능이었다[2]. 현재의 대다수 브라우저가 이를 제공해 주고 있다. IE 7.0은 빠른 Tab기능을 사용하였지만, 기존 방법과 큰 차이가 없고 페이지를 열어본 순서대로 정렬하는데 머물고 있다. Mozilla는 확장성이 뛰어난 웹 브라우저로써 Tab 방식과 Split Browser 확장기능을 통하여 브라우저에서 화면을 나눠 여러 웹 페이지를 실행하는 다분할 화면을 제공하였지만, 특정 브라우저를 사용해야 한다는 제약이 있다.

본 논문은 사용자의 참여가 지속적으로 높아지는 웹에 있어 인터넷의 효율성을 높여 주기 위해 Ajax를 사용하여 웹 페이지를 분할하여 기존의 브라우저와는 달리 하나의 화면에서 복수의 화면을 제공함으로써 불필요한 페이지를 제거 하며 최적의 결과를 볼 수 있는 기능을 제공해 준다는 점이다.

이를 효과적으로 지원해주는 기능으로서 기인하였다. 웹 어플리케이션은 새로운 상호작용이 가능하도록 매 순간 요청에 따른 응답이 발생될 때 불필요한 웹 인터페이스는 생성하지 않고 사용자 요청에 따른 데이터만 포함하는 응답을 수신해야 하므로 태그를 사용하여 데이터의 연동을 가능하게 하여 사용의 효율성을 높여주었다. 웹 페이지의 분할을 통하여 하나의 쿼리를 각기

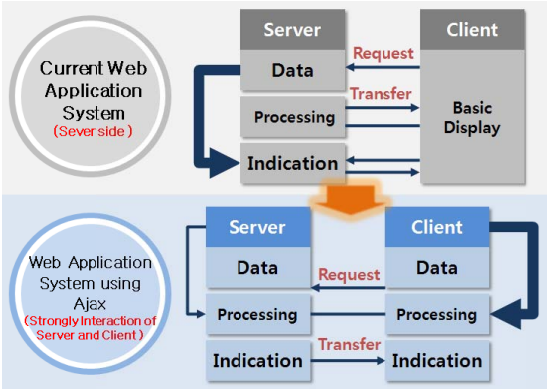
다른 여러 개의 검색엔진에서 한번에 검색결과를 한 페이지에서 볼 수 있다. WEB2.0의 기술에서 외부에서 사용할 수 있도록 공개하여, 응용 프로그램에서 운영 체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있도록 만든 인터페이스로 HTTP와 XML을 처리 할 수 있는 대부분의 언어에서 사용이 가능한 각 검색엔진의 Open API(Open Application Programming Interfaces)를 사용하여 분할된 창에 검색의 결과를 보여주게 하였다[3]. 사용자가 하나의 쿼리로 각각의 검색엔진 페이지를 방문하지 않으므로 정보 검색의 시간이 획기적으로 감축될 수 있다는 장점이 있다.

2. AJAX 기술

Ajax (Asynchronous JavaScript And XML)는 비동기 통신방식을 위해 사용되는 JavaScript, XML등의 기술을 조합이다. JavaScript는 어플리케이션 프로그램에 내장해 사용하기 좋은 범용 스크립팅 언어로 브라우저가 제공하는 여러 기능을 직접 제어할 수 있다[4]. XML (Extensible Markup Language)은 웹에서 구조화된 문석의 전송이 가능한 표준화된 텍스트 형식으로 기존의 HTML에 사용자가 새로운 태그를 정의할 수 있는 기능이 추가되었다. XML을 기반으로 RDF, RSS, MathML, XHTML, SVG등의 언어가 있다. Ajax에서 중요하게 다루는 조합은 표현 정보에서<그림 1>처럼 사용하는 XTHML과 CSS, 객체지향 모델로서 구조화된 문서를 표현하는 형식으로 JavaScript로 DOM을 제어하여 사용자 인터페이스를 동적으로 변경하여 상호교환을 위한 DOM (Document Object Model), HTTP를 사용하여 클라이언트와 서버 간 통신을 위한 기술로 비동기방식의 전송이 Ajax기술의 핵심적인 부분을 담당하는 XMLHttpRequest 등을 의미한다[5].

[그림 1] AJAX 구성

기존의 웹 어플리케이션은 클라이언트로부터 웹 서버로 하나의 요청을 보내면, 서버는 데이터를 가공하여 클라이언트로 새로운 웹 페이지를 생성하여 응답한다. <그림 2>처럼 클라이언트는 웹 서버의 응답이 있을 때까지 대기 해야 하며 서버에서 HTML 코드 전체를 받아 웹 페이지 전체가 새로 이벤트가 일어나므로 많은 정보량의 낭비와 시간이 걸린다. 반면에 Ajax경우 웹 서버에서 페이지로 변경된 이벤트의 데이터만 XMLHttpRequest객체의 상태 체크를 통해 제공하여 처리해 줌으로써 비동기적으로 데이터를 주고받기 때문에 연속적으로 데이터를 계속 요구하지만 서버에 부하를 줄 일수 있다. Ajax는 전체 HTML 페이지를 업데이트 하지 않고 데이터를 전송, 수신하여 수많은 웹 서퍼들은 애플리케이션의 속도가 빨라지고 응답성이 증가하는 것으로 느끼게 되며 상호작용이 반복적으로 이루어지게 된다.



[그림 2] 기존 웹 어플리케이션 및 AJAX기반 어플리케이션

Ajax의 open소스 사용은 모든 사람이 쉽게 소스코드를 획득 할 수 있으므로 다른 사이트에 대한 차별성를 두기 어렵게 된다. 또한 기존의 경우 비즈니스 로직이 서버에서 모두 존재 했지만 Ajax는 클라이언트 쪽에도 존재하기 때문에 해커들의 공격에 안심할 수 없으므로 대비측면에서 개선이 필요하다.

3. 브라우저 효율성

웹 사용에 있어 브라우저의 관점은 기존의 텍스트 중심의 결과제공 기능에서 더 나아가 이미 발전된 그래픽 및 고차원 정보제공기능을 접목하는 새로운 브라우저를 제공해야 할 필요가 있다. 예컨대, 매우 수동적인 브라우저 기능에서 능동적으로 미리 연관 페이지들을 가져다 제공해주는 추천경로 기능이나, 거의 비어있는 forward버튼의 기능사용을 풍부하게 새로운 방식으로 브라우저를 제공 할 수 있다. 개별 사용자의 경우 특정 웹 사이트를 방문하여 순회할 때에 발생하는 가장 중요한 문제 중의 하나는 “브라우징 효율”이다[5]. 브라우징 효율의 개선은 웹 사용에 있어 사용자가 원하는 정보를 찾는데 걸리는 노력을 최소화 한다는 것이다. 최근 어플리케이션을 보면, 웹 서핑 시 사용자에게 시각적인 효율성을 높인 기능을 갖추기 시작했다. IE 7.0 버전에

서 Tab방식을 사용해 복수 페이지 제공을 시도하였지만 기존과 큰 차이 없이 실행된 페이지를 하나씩 차례로 정렬하는데 머물고 있다. Mozilla는 Tab 방식과 Split Browser 확장기능을 통해 화면을 나눠 여러 페이지를 실행 한 다분할 화면을 가능하게 해서 창안에 2개 이상의 페이지를 제공 하였지만 특정 브라우저에 종속적이라는 약점이 있다. WEB 2.0의 기술들은 새로운 브라우저 개발 관점에 적용할 수 있다. 그 중 하나가 Ajax의 사용인데 적절한 브라우저를 이용하는 경우에 전체 페이지를 불러오는 일 없어 응답성을 향상시킬 수 있다. 브라우저마다 속도 차이의 우위가 존재하기는 하지만 다른 여러 프로그램과 마찬가지로 웹 브라우저도 사용자가 어떻게 사용하지도 중요하다.

기존의 브라우저에서 버튼을 클릭하거나 검색 용어를 입력하면 서버에 요청을 생성하고 이에 대한 응답이 브라우저도 다시 보내진다. 요청은 실제로 또 다른 완전 HTML페이지를 만들어내며, 새로운 HTML 페이지가 웹 브라우저 스크린에 다시 나타날 때 플래시 현상이 나타나기도 한다. 이를 동기식이라고 한다. 브라우저는 트리거된 요청이 응답되기 전까지는 애플리케이션을 사용할 수 없다. Ajax는 기존의 경우와는 다르게 요청과 응답은 계속적으로 이루어지지만 비동기식 요청은 서버가 응답할 때까지 대기하지 않으므로 대화성 있고 빠른 애플리케이션을 얻게 되어 사용자로서의 브라우저로 데스크탑 애플리케이션 상에 있는 것과 동일한 것이다.

4. 웹 브라우저 화면 분할

4.1 화면 분할

Ajax을 사용하여 페이지 내의 DOM을 제어하며 페이지 내의 분할을 가능하게 하였다. 기본화면으로 접속 할 사이트의 3개 URL를 입력하는 텍스트박스, 새로운 창을 생성, 웹 페이지 내의 전체 창을 초기화 해주는 기능을 포함하는 linkForm을 구현한다. 3개의 접속 창을 배치하고 Google Ajax Search API를 사용하여 하나의 접속 창에 사용하였다[6]. URL을 입력 후 send 버튼을 클릭하면 해당 사이트에 접속할 수 있다. 태그를 통하여 사이트에 접속 하므로 접속 창의 정보를 소유할 수 있다. JavaScript에서 새로운 속성을 생성하여 창 번호의 값, 창 겹침 기능 설정, 창 위치 설정, 창과 마우스 거리, 드래그 여부를 확인할 수 있다. 창마다 다른 사이트의 접속이 가능하므로 화면 내에 복수의 페이지 표시가 가능하다. 마우스 이벤트가 태그에 발생하면 창의 위치 정보와 마우스 위치를 활용하여 각각의 창 화면에 대해 자유롭게 이동이 가능하다.

<표 1> div 태그

	div 속성	속성 내용
div속성	div.setAttribute	창 번호의 값
	div.style.position	창 겹침 가능하게 설정
	div.style.left div.style.top	창 위치 설정
	div.offLeft div.offTop	창과 마우스의 거리
	div.draging	드래그여부 확인
마우스 이벤트	onmousedown	draging속성의 변경 창 위치와 마우스 위치 간격 계산 후 저장
	onmousemove	마우스 위치를 검사하고, draging속성이 true일 경우 창의 위치를 변경
	onmouseup	Dragging 속성을 false
	onmouseout	마우스 위치 오류 검사

Http://jsgt.org/mt/01에서 제공하는 jsgt_dragfloat.js 파일을 통해 페이지를 분할 한다[7]. dragableFloat(id, x, y) 함수를 통해서 새로운 태그를 생성하여 속성값과 마우스 이벤트 값을 통해 드래그 앤 드롭 기능을 구현하고 id로 창들의 번호를 지정하여 x, y의 위치에 창을 표현한다.

```
1  function on_loadedLink(url, Box) {
2  if(!dragableFloatId[selectBox(Box)] ||
       divArray[selectBox(Box)].style.visibility=="hidden"){
3          return alert("해당 박스가 없습니다.")  }
4      else if(url=="" || url=="NULL")  {
5          return alert("URL 을 입력하세요")  }
6      else {
7          if(url.substr(0,7)!="http://")  {
8                  url="http://"+url;
                   linkInputForm.linkURL1.value=
9                  "http://"+linkInputForm.linkURL1.value;  }
   document.getElementById(selectBox(Box)).innerHTML=
   "<div class='boxname' align='right'>Box"+selectBox(Box)+
   "<img src=./img/icon_del.gif type='button' onClick=
10 deleteDragableFloat("+selectBox(Box)+");
    style='cursor=hand' alt='닫기'></div>" +
   "<iframe src='"+url+"' width='550' height='380'
   border='0' frameborder='0'></iframe>"
```

[그림 3]브라우저 화면 분할 알고리즘

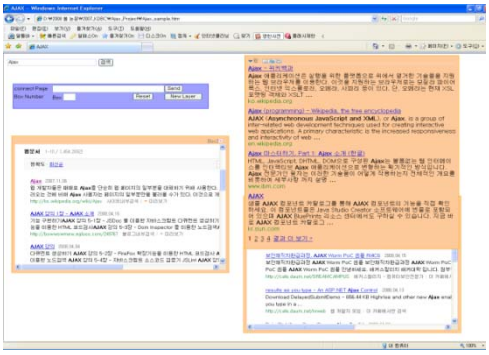
사이트 접속과정은 <그림 3>에 알고리즘으로 표현하였다. linkForm 텍스트 박스에 URL을 입력 받아 창 번호를 검색 후 해당 창에 사이트를 접속한다.

linkForm 창에 URL을 입력 후 버튼을 클릭하면 on_loadedLink(url)함수가 발생하고 해당 창의 존재 여부를 확인한다. 창이 존재하지 않을 경우 “해당 박스가 없습니다.”라는 경고 창을 호출한다. 접속요청의 URL입

력 여부를 확인하여 URL이 입력되지 않으면 “URL을 입력하세요.”라는 경고 창을 호출한다. URL과 박스 번호가 올바르게 입력되면 http://사용여부 확인하고 선택된 창에 해당 URL 사이트 접속한다.

4.2 동기식 화면 갱신

Mozilla와 IE 7.0은 브라우저의 툴에서 키워드를 입력하고 Tab방식으로 원하는 검색엔진을 선택하여 검색의 결과를 브라우저로 보여주고 있지만, Tab안에 여러 검색엔진의 등록은 가능하지만 한번의 클릭으로 복수의 검색엔진 결과를 볼 수는 없다.



[그림 4] 각 검색엔진의 검색

본 논문에서는 페이지 분할을 사용하여 각 창마다 각기 다른 검색엔진을 연결하여 하나의 키워드를 입력 받아 창에 서로 다른 검색엔진 결과를 가져오게 한다. 또한 서버에 수동으로 업로드 하지 않고 브라우저에서 동적으로 페이지를 빠르게 볼 수 있게 해준다. 각 검색엔진의 계산은 다르므로 한번의 키워드 입력과 한번의 검색 버튼의 클릭으로 각 검색엔진의 특정 있는 검색의 결과를 보여준다. Google은 PageRank 알고리즘을 사용하여 웹 페이지의 중요도를 객관적으로 평가하며, 특정 페이지의 중요도를 고려하여 중요한 페이지부터 링크를 받은 경우 링크된 페이지에 더 큰 값을 메긴다. 중요하고 질이 좋은 사이트 들은 높은 페이지 순위의 점수를 받아 Google 검색엔진 사용시 검색결과가 상단에 표시된다[8]. Naver는 키워드형 검색엔진이라고 할 수 있다. 찾고자 하는 정보와 관련된 핵심어를 입력하여 정보를 찾는 방법으로 인터넷에 있는 홈페이지의 내용과 URL을 자체적으로 보유한 데이터베이스로 구축해 두어 키워드 입력 시 그 데이터베이스에서 의존하여 결과를 보여주는 검색 체계이다[9]. Daum 검색엔진은 문서검색에 있어 Google 검색엔진을 사용하고 있다. 각 검색엔진의 사용에 있어서 WEB 2.0의 대표적인 기술 중 Open API를 사용하였다. Open API는 자사의 API를 외부에 웹 서비스(Web Services) 형태로 공개한 것으로 개방지향적인 성격을 나타내고 있다. 본 논문에서는 Google의 검색 Ajax Open API[6]와 Naver 웹 문서 검색 API[10], Daum 카페검색 API[11]를 사용하였다.

텍스트 박스에서 키워드를 입력 받아 검색 버튼을 클릭하면 동적으로 각 분할된 페이지마다 주어진 검색엔진의 Open API를 통하여 페이지에 주어진 키워드의 검색 결과를 제공해준다.

4.3 웹 페이지 화면 분할의 활용

현재까지의 브라우저에서 Mozilla는 복수의 페이지를 가장 효율 적으로 제공해 주며[8], IE 7.0은 Tab방식으로 브라우저에서 화면 분할을 제공해 주지 않는다[12].

본 논문에서는 Mozilla와 비교하여 Ajax 기반 웹 페이지 분할의 장점과 활용방안을 제안한다. Mozilla와 큰 차이점은 화면을 나누는 방법에 있다. Mozilla 경우 Tab사용으로 화면 전체 공간을 나눈다. 나눠진 공간마다 다른 사이트에 접속이 가능하므로 사이트간 비교가 용이 하다.

[그림 5] 창의 이동, 생성, 겹침 기능

Ajax 웹 페이지 분할은 Mozilla와 같이 화면이 나눠져 보이지만, 태그를 통해 접속 창의 삽입이 가능하다. 페이지를 태그를 통해 화면을 나눴다는 것은 사용자가 사용자 환경에 영향을 받지 않는 다는 장점이 있다. 사이트 접속 창 생성방법을 Ajax 기반으로 한 것은 Ajax가 현재 사용되고 있는 대부분의 브라우저에서 활용이 가능하기 때문이다. Mozilla 경우 전체화면에서 나누어 정해진 공간에서만 사용하는 것에 그치지만, 태그를 사용한 것은 태그 속성을 통해 창의 겹침이나 이동을 자유롭게 할 수 있어 사용자에게 분할 된 페이지의 시각적인 효율성 높여주고 사이트 접속 창간의 데이터 연동이 가능하다. Mozilla 경우 화면을 나눈 효과만 가져오므로 분할된 공간은 개별적인 브라우저로 의미한다. Ajax 웹 페이지 분할은 창으로 다수의 사이트에 접속해 있지만 모두 하나의 페이지다. 같은 페이지에 창이 있다는 것은 창들 간의 데이터 연동이 가능하므로 복수 페이지 사용의 효율성을 높일 수 있다.

5. classic web 과 Ajax web 속도측정

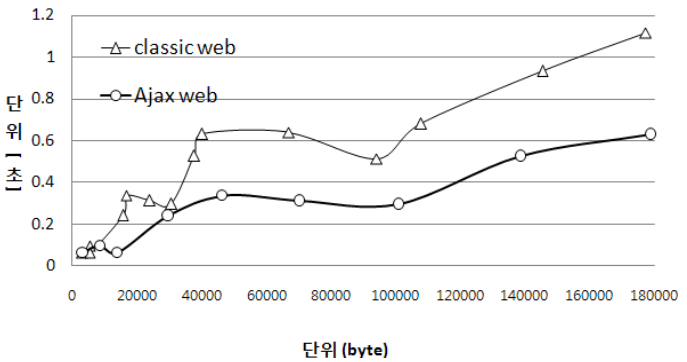
기존 웹 기술과 Ajax의 차이점이 데이터 전송 측면에서 Ajax는 부분적인 데이터만 전송함으로서 전송량을 줄일 수 있다고 하였다. 기존 웹 방법과 Ajax 기반 페이지 분할의 접속 시간을 측정함으로서 두 기술의 차이점을 분석한다.

실험은 네트워크 분석기인 Ethereal을 사용하여 페이지 요청시점과 완료시점의 시간 간격을 계산하여 페이지로의 접속 시간을 측정한다[13].기존 웹의 데이터 용량별 접속 시간을 보기 위해 3202~177414(byte) 용량에 해당하는 임의의 페이지에 접속한다. <표2>은 접속한 사이트의 목록이며, <그림6>의 classic web이 측정 결과를 나타낸다.

<표 2> 접속 한 사이트 목록

사이트	데이터 용량	사이트	데이터 용량
www.7-eleven.co.kr	3202	www.lge.co.kr	37708
www.google.com	5349	java.sun.com	40200
www.tnccompany.com	5429	www.ebay.com	66987
www.fox.com/home.htm	15649	www.cjmall.com	94128
code.google.com	16794	www.yahoo.com	107946
ajax.asp.net	23991	news.google.com	145410
www.kaist.ac.kr	30612	www.joins.com	177414

<그림 6>의 실험은 기존 웹의 경우 하나의 사이트를 방문 했을 경우의 접속시간을 나타낸 것이고, Ajax 기반 페이지 분할의 경우는 다수의 페이지에 접속한 경우를 나타내는 차이점이 있다. 하지만 같은 데이터 크기일 경우 Ajax 방법이 접속 시간이 적게 드는 것을 볼 수 있다. 이것은 페이지에서 표시하는 데이터의 전체 크기는 같지만 Ajax 기반 페이지 분할의 경우 표시 된 복수의 사이트 중 접속 요청된 페이지의 데이터 크기만 전송되기 때문에 접속 시간의 단축이 일어난다.



[그림 6] Classic web과 Ajax web 속도

Ajax 기반 페이지 분할을 사용하면 1 개의 사이트에 접속하는 시간은 차이가 없지만, 복수 사이트에 접속할 경우 접속된 사이트의 화면 변화 없이 추가로 접속 요청된 사이트의 데이터 전송 및 부분적인 화면 변화로 다수 사이트의 누적된 데이터 크기에 상관없는 접속 속도를 유지할 수 있다.

6. 결론 및 향후 연구과제

웹 페이지를 Ajax 기반으로 분할 하여 자유로운 복수의 페이지를 사용을 제공해 하였으며, 동적으로 키워드 입력 후 한번의 클릭으로 여러 검색엔진을 한 브라우저에서 사용하므로 웹 서핑에 있어 시간적 효율성을 높여주었다. 이 방법을 웹 쇼핑에 적용을 한다면 여러 쇼핑몰을 브라우저 화면 분할에 기반한 비교가 가능하다. 정보 검색 시 여러 검색엔진을 방문 하지 않아도 검색엔진의 특징에 따라 여러 검색 결과제공 받을 수 있으며, 중복 정보를 필터링 할 수 있다.

향후 연구 과제로는 복수의 페이지에 사용자가 원하는 검색엔진의 선택하는 기능이 필요하다. 하나의 특정 검색엔진을 사용 할 경우 키워드 입력 시 검색 결과의 Top-K를 선정하여 각 페이지로 바로 연결을 하여 사용자로 하여금 최적화 검색을 가능하게 하는 기능으로 확장할 수 있다.

7. 참고문헌

[1] N.Hanakawa, N.Ikemiya, “A new web browser including a transferable function to Ajax codes”, In Proc. ASE, pp.351-352, 2006.

[2] http://firefoxinside.tistory.com/entry

[3] D.E. Millard, M.Ross, “Web 2.0: hypertext by any other name”, In Proc. Hypertext, pp. 27-30, 2006.

[4] R. Atterer, A. Schmidt, "Tracking the interaction of users with ajax applications for usability testing," In Proc. CHI, pp. 1347-1350, 2007.

[5] K.Smith, “Simplifying Ajax-Style Web Development”, IEEE Computer, 39(5), pp.98-101, 2006.

[6] http://code.google.com/apis/ajaxsearch

[7] http://jsgt.org/mt/01

[8] S.Brin, L.Page, “The anatomy of a large-scale hypertextual web search engine”, InWWW Conference, volume 7, pp.107-117, 1998.

[9] http://photouni.tistory.com

[10] http://openapi.naver.com/index.nhn

[11] http://dna.daum.net/apis/GettingStarted

[12] explorer 7, http://www.microsoft.com/windows/products/winfamily/ie

[13] http://www.ethereal.com