
웹 2.0을 위한 Ajax기반 RSS리더 모듈 개발에 관한 연구

권영제* · 김차중**

A Study on the Development of RSS Reader Module Based on Ajax for Web 2.0

Young-jae Kwon* · Cha-jong Kim**

요 약

현재 기존의 웹과 차별화를 의미하는 웹 2.0에 대한 관심이 높아져 가고 있다. 기존의 시스템이 클라이언트와 서버 모델에 기반을 둔 정적인 웹이 전형적이었다면, 웹 2.0은 웹이 근본적으로 변화하고 진화한다는 차원의 차세대 웹을 뜻한다. 본 논문에서는 웹 2.0규격하에서 쓰일수 있는 RSS리더에 대한 모듈을 제안한다. 제안된 모듈은 Ajax를 이용하여 개발하였다. 현재 쓰이는 RSS Feed가 브라우저에 따라 왜곡될수 있는 부분을 보정하고 한글 및 제 3세계 언어의 인코딩 문제를 해결하였고, Ajax의 유동성 있는 프로그래밍을 통한 DragBox모듈과 RSS Feed의 자동 업데이트 모듈의 설계 및 개발을 통해 웹 2.0 기반상에서의 효과적인 콘텐츠의 활용을 위한 모듈을 개발하였다.

ABSTRACT

Recently, it is increasing the interest in previous web and web 2.0 which mean differentiation. The previous system is typical Static web based on client and server model, while web 2.0 mean next generation web which web change and evolve fundamentally. In this paper, I suggest that the module on RSS reader available under web 2.0 standards. A currently used RSS feed do correct the part distortion which can be possible resolve the encoding problem of Hangeul and third-world language. The suggested module is implemented using Ajax. I developed the module for the use of effective contents on web 2.0 through the DragBox module based on the programming on mobility of Ajax and the design and implementation of automatic update module of RSS feed.

키워드

Ajax, RIA, RSS Reader, SOA, Web 2.0, W3C

I. 서 론

현재 기존의 웹과 차별화를 의미하는 웹 2.0에 대한 관심이 높아져 가고 있다. 웹 2.0의 개념은 O'Reilly와 MediaLive International의 컨퍼런스 브레인스토밍 세션에서 Dale Dougherty에 의해 시작되었다.[1] 기존의 시스템이 클라이언트-서버 모델에 기반을 둔 정적인 웹이 전형적이었다면, 웹 2.0은 웹이 근본적으로 변화하고 진화

한다는 차원의 차세대 웹을 뜻한다.

웹 2.0은 대개 세가지 요소가 결합되어 최적화 되는것을 보고 있다. Model(표준화된 데이터 전송), View(UI, Rich Internet Application), Control (공개된API)이다. 그리고 또 다른 웹 2.0을 위한 기술로 각광받고 있는 주요 키워드로는 시맨틱웹(Semantic Web)[2]이 있으며, 사용자 중심의 리치한 인터페이스 설계가 가능한 XML(Ajax), Java Script, 플렉스(Flex) 등이 있다.

* 한밭대학교 정보통신전문대학원 컴퓨터공학과

** 한밭대학교 정보통신컴퓨터공학부

이렇게 웹 2.0 기술이 각광을 받으며 많은 커뮤니티가 생겨나고 현재도 많은 블로그 사이트 혹은 개인 유저들의 커뮤니티 활동을 위한 게시판사이트와 공유사이트가 하루에도 수없이 쏟아져 나와 네트워킹에 많은 부하를 주고 있다.

이를 보완하기 위해 나온 기술인 RSS(Really Simple Syndication)[3]는 콘텐츠 신디케이션(Contents Syndication)을 위해 나온 XML 형태의 규격 중 하나로 웹 사이트끼리 서로 자료를 주고받기 위한 규격이다. 하지만 그러한 RSS는 해당 사이트에서 지원을 해야 쓸 수 있고, RSS 전용 브라우저가 아니라면 문서의 형식이 조금은 바뀌기 때문에 원본의 형식과 폼이 바뀌어져 왜곡될 수도 있다는 것이다.

본 논문에서는 RSS의 이러한 문제점을 해결하고자 웹 2.0상의 특성을 고려하고 Ajax[4]의 유동성 있는 프로그래밍을 통한 모듈을 연구하였다. RSS Feed가 브라우저에 따라 왜곡될 수 있는 부분을 보정하는 모듈과 한글 및 제 3세계 언어의 인코딩 문제를 해결하고, DragBox 모듈과 RSS Feed의 자동 업데이트 모듈을 구현해, RSS 리더 시스템이 웹 2.0 기반 상에서 얼마나 효과적으로 쓰일 수 있는지를 보여주는 시스템이라 할 수 있겠다.

II. 관련연구

본 장에서는 1장에서 제안한 모듈에 쓰이는 기술을 간단히 소개한다.

2.1. Ajax의 사용

Ajax는 비동기식 자바 스크립트와 XML의 합성어로 '에이잭스'라고 읽는다. 구글의 지도 서비스나 검색 사이트의 추천 검색어 서비스 등이 Ajax의 대표적인 사례다. 그리고 과거 액티브 엑스나 플래시로 구현했던 기능들은 Ajax가 모두 대체할 것으로 보인다.

기존 웹 애플리케이션 이용 행태에서는 사용자가 서버 측에 요청을 보내면 서버에서 응답이 클라이언트로 모두 전송되고 관련 리소스가 모두 로딩된 이후에 또 다른 행동을 취할 수가 있었다.

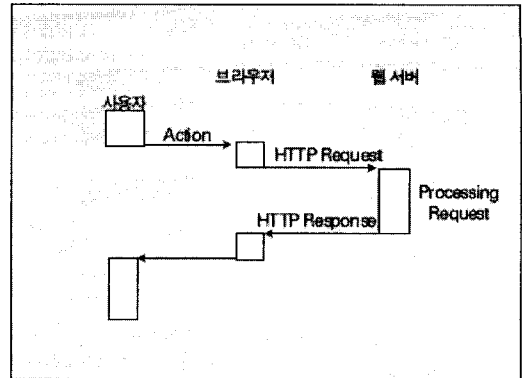


그림 1. 기존의 웹 애플리케이션
Fig. 1. Existing Web Application.

하지만 Ajax 어플리케이션에서는 서버 측에 보내는 요청이 비동기적으로 이루어지기 때문에 요청을 보낸 이후에도 응답이 완료되기까지 기다리지 않고 또 다른 요청을 보내거나 그 이외에 액션을 취할 수 있다.

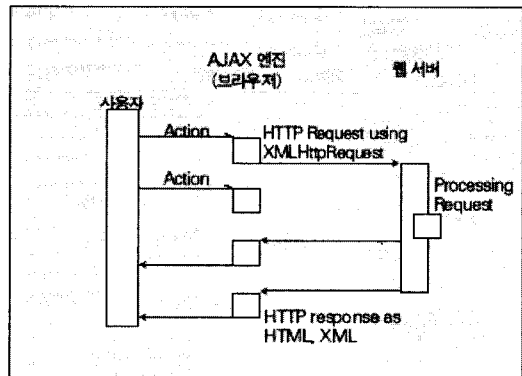


그림 2. 에이잭스 애플리케이션
Fig. 2. Ajax Application.

특히 새로거나 복잡한 기능은 아니지만 Ajax의 핵심은 사용자들에게 더 편리하고 직관적인 서비스를 제공한다는 것이다. 사용자들은 기다리는 시간이 줄어들고 대신 브라우저가 더 많은 일을 맡게 된다. 이런 간단한 기능뿐만 아니라 엑셀이나 워드 같은 복잡한 소프트웨어도 웹으로 개발되고 있다. 최근에는 프로그램을 설치하지 않고도 마이크로소프트 오피스와 거의 비슷한 기능을 이용할 수 있는 사이트도 개발되고 있다.

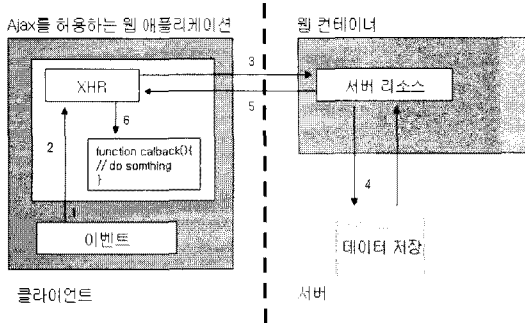


그림 3. 에이젝스의 동작방식
Fig. 3. Operatoin Mode of Ajax.

2.2 PHP를 이용한 가벼운 플랫폼의 적용

개발자들 사이에서는 몇 년 전부터 PHP를 비롯해 파이썬이나 펄 같은 스크립트 언어가 부쩍 인기가 있다. 자바나 C, C++, 비주얼 베이직 같은 전통적인 프로그래밍 언어 보다 배우기도 쉽고 개발과 유지·보수도 훨씬 간단하기 때문이다. 리눅스와 아파치, MySQL 등 공개 소프트웨어를 조합한 LAMP나 루비 온 레일즈 같은 개발 플랫폼은 기존의 자바나 닷넷 프레임을 대체할 수 있을 만큼 발전해 왔다.

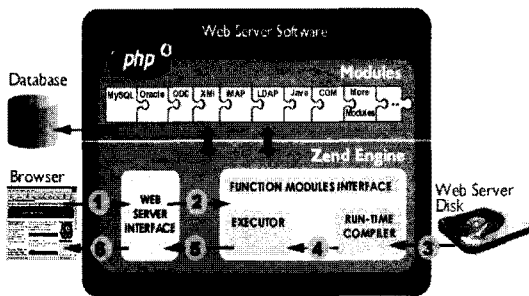


그림 4. PHP시스템의 구조
Fig. 4. PHP System Structure.

이들 공개 개발 플랫폼의 장점은 무엇보다도 가볍고 빠르는데다 라이브러리가 많고 꾸준히 업데이트되고 있다는 것이다. 과거와 달리 기능도 크게 개선됐다. 웹 2.0 시대에 들어서면서 서버·클라이언트 환경보다는 인터넷 기반 개발 환경이 더욱 중요하게 된 것도 가벼운 개발 플랫폼이 주목받는 이유이다.

2.3 RSS

웹 1.0 시대에 사용자들은 뉴스를 보려면 뉴스 사이트를 직접 찾아가야 했다. 브라우저의 검색창에 주소를 직접 입력하기도 하고 주소를 즐겨찾기에 넣어두고 클릭해서 찾아가기도 했다. 그런데 웹 2.0 시대에는 굳이 뉴스 사이트를 찾아가지 않아도 뉴스를 읽을 수 있는 방법이 많다. 특히 RSS는 Rich Site Summary로도 불리우며, XML기반의 표준 통신 포맷이다.

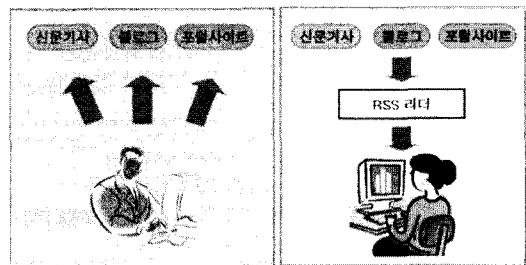


그림 5. 기존 방식과 RSS이용 방식
Fig. 5. Existing Method vs. RSS Method.

신문을 보려고 신문사까지 찾아가야 할 필요가 없는 것처럼 이제 뉴스를 보려고 굳이 뉴스 사이트를 찾아가야 할 필요가 없게 됐다. 뉴스 사이트의 RSS 주소만 알면 실시간으로 최신 뉴스를 확인할 수 있다. 자주 찾아가는 블로그의 RSS 주소를 모두 모아놓고 새로 올라온 글만 찾아 읽을 수도 있다. RSS는 콘텐츠의 발행과 구독이라는 새로운 정보 유통 방식을 만들어냈다.

이러한 네이버 뉴스에서는 특정 주제로 검색을 하고 그 검색 결과를 RSS로 만들 수 있다. 이 주소를 RSS 구독기에 집어넣으면 새로운 뉴스가 뜰 때마다 바로바로 확인할 수 있게 된다. 굳이 뉴스를 찾으러 가지 않아도 찾아와서 알려주는 시대가 된 것이다. RSS 주소를 모아서 링크를 뽑아내 조합하면 전혀 다른 뉴스 사이트를 만들 수도 있다. 이 사이트는 특정 주제의 뉴스만 모아 실시간으로 업데이트하게 된다. 그리고 RSS는 텍스트뿐만 아니라 멀티미디어까지 포괄해 콘텐츠 유통의 광범위한 표준으로 자리 잡고 있다.

2.4 CSS

CSS(Cascading Style Sheets)[6]는 콘텐츠와 디자인을 분리하는 방식이다. 별도의 파일로 스타일 양식을 만들

어 지정해두고 그 파일만 고쳐주면 페이지 디자인이 바뀌게 된다. 테이블을 쓰는 것보다 직관적이지는 않지만 전체적으로 디자인 원칙을 잡는 데 도움이 된다. 수많은 페이지의 디자인을 하나의 CSS 파일로 관리할 수도 있다. CSS는 월드 와이드 웹 국제 컨소시엄인 W3C (<http://www.w3c.org>)[5]가 권장하는 표준 웹 디자인 방식이다.

표준화된 CSS 방식을 도입하면 디자인을 손쉽게 바꾸는 것은 물론이고 필요에 따라 그림 파일을 뺀 텍스트 파일만 불러올 수 있고 휴대전화나 PDA 버전으로 만들 수도 있다. 개발과 유지·보수에 드는 비용과 시간도 크게 줄어든다. 무엇보다도 데이터의 가공과 활용이 훨씬 자유롭게 된다. 콘텐츠와 디자인이 분리될 때 데이터 디자인이 가능하게 된다.

III. 모듈의 구성

본장에서는 본 논문에서 사용될 핵심 모듈의 기능을 설명하였다.

3.1 모든 브라우저의 지원

그동안 개발자들은 인터넷 익스플로러에 맞춰서 사이트를 개발해 왔다. 그래서 익스플로러에서는 보이지만 다른 브라우저에서는 보이지 않는 사이트도 많았다. 우리나라의 경우 최근까지 익스플로러의 시장 점유율이 99%에 이를 정도였다. 익스플로러가 아니면 보이지 않기 때문에 다들 익스플로러를 쓰게 되고 익스플로러로 접속하기 때문에 개발자들도 굳이 다른 브라우저를 지원할 필요를 느끼지 못했던 것이다.

그런데 10만명의 1%는 1천명이지만 1천만명의 1%는 10만명이다. 이 10만명을 결코 무시해서는 안 된다. 게다가 세계적으로 익스플로러의 시장 점유율은 갈수록 떨어지고 있다. 익스플로러를 버리고 파이어폭스를 쓰는 사용자들이 이미 10%를 넘어섰다. 이 비율은 앞으로 더욱 늘어날 것으로 보인다. 핵심은 이제 익스플로러가 유일한 브라우저가 아니라는 있다는 것이다. 익스플로러 뿐만 아니라 파이어폭스를 비롯해 모든 브라우저를 받아들이는 최선의 대안은 결국 철저히하게 표준을 따르는 것이다.

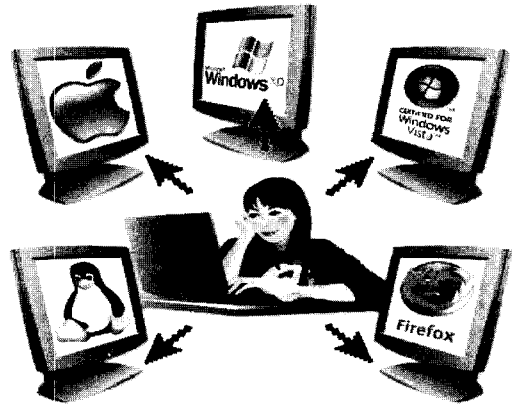


그림 6. 다양한 OS와 브라우저 사용
Fig. 6. Using of Various OS and Browsers.

3.1.1 XMLHttpRequest

XMLHttpRequest는 웹 서버와 통신하기 위한 Ajax의 핵심 컴포넌트이다. 이미 IE에서는 5.0에서부터 액티브엑스 오브젝트 형태로 제공되고 있고 그 이외의 브라우저에서는 XMLHttpRequest라는 윈도우 객체에 속해 있는 객체의 형태로 제공된다. Ajax를 이용하여 스크립트를 작성할 때 가장 큰 차이를 보이는 부분이 XMLHttpRequest 객체를 가져오는 부분인데, 브라우저에 따라 차이를 보인다. XMLHttpRequest는 아직 W3C 표준이 아니지만 W3C에는 XMLHttpRequest와 비슷한 DOM Level3의 Load and Save specification이 있다. 아직 DOM Level3의 Load and Save가 구현된 브라우저는 없지만 다이나믹하게 XML문서를 읽어내서 DOM에 적용하고 DOM구조를 XML로 차례로 나열한다는 개념은 다소 비슷한 부분이 있다.

3.1.2 XMLHttpRequest 객체

XMLHttpRequest를 얻어오는 방법은 IE와 그 이외의 브라우저가 서로 다르다.

인터넷 익스플로러의 경우에는 다음과 같은 방법으로 객체를 얻을 수 있다.

```
var xmlhttp = new ActiveXObject("Microsoft.XMLHTTP")
```

그림 7. 인터넷 익스플로러의 객체 파싱
Fig. 7. Object Parsing by Internet Explorer.

그 이외의 브라우저에서는 다음과 같은 방법으로 객체를 얻어올 수 있다.

```
var xmlhttp = new XMLHttpRequest()
```

그림 8. 다른 브라우저의 객체 파싱
Fig. 8. Object Parsing by Other Browser.

좀 더 일반적인 방법으로는 다음과 같이 XMLHttpRequest 객체를 얻어올 수 있다.

```
var xmlhttp = false
if(window.XMLHttpRequest){
    xmlhttp = new XMLHttpRequest()
} else {
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP")
}
```

그림 9. 일반적인 객체 파싱
Fig. 9. General Object Parsing.

3.2 문자 인코딩의 UTF-8화

컴퓨터의 이진수를 문자로 바꾸는 과정을 인코딩이라고 하는데 당연히 쓰는 언어에 따라 인코딩 체계가 다르다. 우리나라 사이트는 그동안 대부분 EUC-KR이라는 방식을 사용해 왔다. 그런데 문제는 한글이 아닌 다른 언어로 된 운영체제에서는 이 방식의 페이지를 읽지 못한다는 것이다.

이런 문제를 해결하는 대안이 바로 유니코드라고 불리는 UTF-8[7] 방식이다. UTF-8은 한글과 한자를 비롯해 4만자에 이르는 세계 대부분의 나라의 언어를 포함하고 있다. 만국 공통의 문자부호 체계인 셈이다. UTF-8 방식으로 인코딩된 페이지는 세계 어느 나라 어느 언어로 된 운영체제에서도 특별한 설정 없이 우리가 보는 것과 똑같은 페이지를 보여준다. 세계적으로 웹 2.0 시대를 주도하는 사이트들은 이미 UTF-8 인코딩을 적용해 한글뿐만 아니라 세계 모든 나라의 언어를 자유롭게 쓰고 읽을 수 있다.

한글 인코딩에는 XMLHttpRequest 객체를 사용하면 간단하다. Ajax의 핵심이 바로 이 객체가 아닌가 생각한다. XMLHttpRequest 객체를 사용하여 데이터를 PHP 파일로 송신하면 데이터는 UTF-8로 인코딩되어 전송된다.

```
var url = some url;
var param = 'id=myid&password=mypass&name=이름';
xmlHttpPost(url,param,result function);
```

그림 10. 한글이 전송되지 않는 예
Fig. 10. Example in which Hangul is not transmitted.

그림 9처럼 인자값에 한글이 들어갈 경우에는 제대로 전송이 되지 않는다.

```
var param = 'id=myid&password=mypass&name=' +
escape(encodeURIComponent('이름'));
```

그림 11. 한글을 전송가능하게 하는 예
Fig. 11. Example which can Hangul be transmitted.

encodeURIComponent,escape 두 함수를 이용해서 공을 한번 해줘야 제대로 전달이 된다.

다음으로 PHP에서 위 인자값을 받을 때는 그림 11과 같이 쓰인다.

```
$id = $_POST['myid'];
$pass = $_POST['password'];
$name = iconv('UTF-8','EUC-KR',urldecode($_POST['name']));
```

그림 12. PHP 인자값 받기
Fig. 12. Receiving a PHP argument.

자바스크립트에서는 함수를 적용한 역순으로 Decoding을 먼저 해주고 문자셋을 EUC-KR로 변경해주면 된다.

3.3 일반 컴퍼넌트Box의 RIA화

일반적인 RIA(Rich Internet Application)[8]란 기존의 웹 애플리케이션 기술이 가진 평면적인 표현과 순차적인 프로세스를 다이나믹한 사용자 인터페이스와 데이터 베이스의 연동을 통해 저렴한 비용으로 하나의 인터페이스에서 모든 프로세스가 처리 가능 하도록 해주는 기술을 말한다. 즉 하나의 화면에서 구현 가능한 웹 애플리케이션이라고 할 수 있다. 그리고 자신이 하고자 하는 모든 것을 한 페이지 안에서 모두 할 수 있다는 것이 바로 RIA의 장점이다.

본 논문에서는 Ajax를 활용한 DragBox처럼 유동적인 박스를 만들어 RIA처럼 한 화면에서 구현할 수 있는 인터페이스를 개발했다. 그림 13은 DragBox의 function을

정의한 것이다.

```
function autoScroll(direction, yPos) /* 박스의 스크롤 기능 */
function initDragDropBox(e) /* 박스의 위치를 잡아주는 기능 */
function initDragDropBoxTimer() /* 박스의 로딩타이머 */
function moveDraggableElement(e) /* 박스의 끌기 기능 */
function stop_dragDropElement() /* 박스의 끌기 멈추는 요소 */
function createColumns() /* 박스가 만들어질 배열을 만든다 */
function mouseoverBoxHeader() /* 헤더 위에 마우스가 올라가있을때 행동 */
function mouseoutBoxHeader(e,obj) /* 헤더 위에 마우스가 없을때의 행동 */
function showHideBoxContent(e,inputObj) /* 숨겨져 있는 박스의 정보를 보여줌 */
function mouseover_CloseButton() /* 마우스가 위에 있을시에 닫히는 버튼 */
function highlight_CloseButton() /* 마우스가 위에 있을시에 닫히는 버튼 */
function closeDragableBox(e,inputObj) /* 끌수있는 박스를 닫는 기능 */
function addBoxHeader(parentObj,externalUrl,notDragable)
/* 박스의 헤더를 불러들임 */
function addRSSEditContent(parentObj) /* 박스안의 정보를 편집 */
function addBoxContentContainer(parentObj,heightOfBox) /* 박스안의 정보생성 */
function addBoxStatusBar(parentObj) /* 박스의 헤더부분의 기능 */
function createABox(columnIndex,heightOfBox,externalUrl,uniqueIdentifie
r,notDragable) /* 박스의 생성 */
```

그림 13. DragBox Components
Fig. 13. DragBox Components.

3.4 RSS의 자동 업데이트

RSS 기술은 시간과 장소에 관계없이 사용자가 원하는 부문의 콘텐츠를 자동으로 보내주는 웹2.0 관련 핵심 기술이다. 일반적으로 RSS Feed를 추가하는것으로는 RSS를 활용할 기반이 안되는것이다. RSS는 자동으로 업데이트가 되어야 한다.

```
function initDragDropBoxTimer() {
    if(dragDropCounter>=0 && dragDropCounter<10){
        dragDropCounter++;
        setTimout('initDragDropBoxTimer()',10);
        return;
    }
    if(dragDropCounter=10){
        mouseoutBoxHeader(false,dragObject);
    }
}
```

그림 14 Timer Component
Fig. 14. Timer Component.

그림 14는 DragBox가 생성될때 시간을 받아서 시간에 맞추어 RSS Feed를 다시 불러들이는 역할을 한다.

IV. 시스템의 구현

본 모듈은 사용자가 별도의 환경설정 후 사용할수 있도록 PHP를 사용하여 구현하였다.

4.1 시스템의 구현

본 시스템은 그림 15의 입력폼에서 입력받은 RSS주소와 RSS의 개수, 업데이트 시간, 박스의 크기등을 추출하고 입력받은 정보중 RSS Feed 정보만을 제외한 유동적인 박스를 생성하고, RSS Feed와 입력받은 정보를 그림 16의 박스안에 보여준다.

새로운 RSS feed 추가

RSS 주소 :

RSS 갯수 : 업데이트 : 분

박스 크기 :

그림 15 RSS Feed정보 및 기타정보 입력 인터페이스

Fig. 15. RSS Feed information and other information input interface.

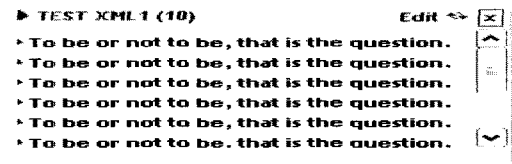


그림 16. DragBox의 초기 화면
Fig. 16. Display of initial DragBox.

그림 17에서 보여지는 DragBox내에서 보여지는 Edit 모드는 RSS Feed정보 및 보여지는 RSS의 개수와 크기 업데이트시간 등 그림 15에서 보여주었던 입력 인터페이스에서 생성하였던 내용을 볼 수 있으며, 입력내용을 바꿀 수 있다.

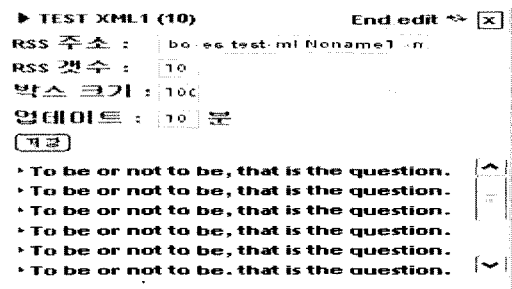


그림 17. DragBox의 Edit모드 화면
Fig. 17. Display of Edit Mode of DragBox.

그림 18에서 보여주는 DragBox의 비활성화 화면은 그림 16의 왼쪽 위쪽에 보이는 “▶”를 클릭했을때 비활성화 되어서 화면을 잠시 닫거나 할때 쓰일수 있다.그림 18의 왼쪽 위쪽에 보이는 “▼”을 클릭시에 다시 그림 16처럼 펼쳐져서 내용을 볼수 있게 해준다.



그림 18. DragBox의 비활성화 화면
Fig. 18. Display of Deactivation of DragBox.

그림 19에서는 RSS Feed Reader시스템을 3개 OS의 대표적인 브라우저(Apple Mac-Safari, RedHat Linux-Mozilla, Microsoft XP-Internet Explorer)를 대상으로 테스트를 해보았고, 보여지는 RSS Feed데이터가 정상적으로 보여지는 것을 볼 수 있다.



그림 19. 각 브라우저별 실행 화면
Fig. 19. Each browser display.

V. 결 론

현재 다양한 웹 애플리케이션 기술이 나와있다. 기술을 선택할 때 가장 중요시 해야 되는 것은 바로 사용자 접근성이다. 웹 브라우저 지원 범위와 운영체제, 기타 디바이스 지원에 대한 것은 매우 중요한 일이다. 웹 2.0의 경험적 요소 중에는 웹을 더욱 동적으로 만들고 풍부한 UI를 선보이는 특징이 있다. 이것이 웹 서비스와 데스크톱 애플리케이션과의 경계를 모호하게 해서 웹 애플리

케이션이 발전 되도록 한 요인이기도 하다.

이러한 요인으로 인해 본 논문에서는 웹 2.0규격하에서 쓰일 수 있는 RSS리더에 대한 모듈을 개발하였다. 좀더 유동적인 웹 2.0의 구현을 위해 Ajax를 이용하여 구현하였고, 현재 쓰이는 RSS Feed가 브라우저에 따라 왜곡될 수 있는 부분을 보정하는 모듈과 한글 및 제3세계 언어의 인코딩 문제를 해결하는 모듈을 개발하였다. 그리고 끌어다넣기가 가능한 DragBox모듈과 RSS Feed의 자동 업데이트 모듈의 개발을 통해 웹 2.0 기반상에서의 효과적인 콘텐츠의 활용을 위한 모듈을 구현하였고, 마지막으로 좀더 풍부한 사이트의 구현을 위해 Ajax와 Flex의 연동과 SOA(Site Open API)를 이용한 사이트간의 정보 교류에 관련한 연구가 계속해서 진행되어야 할 것이다.

참고문헌

- [1] Tim O'Reilly, "What is Web 2.0", <http://www.oreil-lynet.com/public/a/oreilly/tim/news/2005/09/30/What-is-web-20.html>, 2005.
- [2] 권혁철, "시맨틱 웹의 가능성과 한계", 지식정보인프라, 통권 15호, PP · 15-18, 2004. 7.
- [3] adaptivepath, "Ajax: A New Approach to Web Applications", <http://www.adaptivepath.com/publications/essays/archives/000385.php>, 2005.
- [4] Ben Hammersley, RSS and Atom, O'Reilly, 2005.
- [5] W3C, "DOM Level 1 Specification", <http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001>, 1998.
- [6] W3C, "W3C Recommendation:Cascading Style Sheets, (CSS) level1", <http://www.w3c.org/TR/1999/REC-xml-19990111>, 1999
- [7] The Unicode Condortium, The Unicode Standard, Version 2.0, 1996.
- [8] Michal Malaj, "RIA - Rich Internet Applicati on", <http://flex2.blogspot.com/2006/09/ria-rich-internet-app-lication.html>, 2006

저자소개



권영제 (Young-jae Kwon)

2005년 우송대학교 컴퓨터과학과
(공학사)

2007년 한밭대학교 컴퓨터공학과
(공학석사)

※ 관심분야: 웹 2.0, 시멘틱 웹, RIA, Ajax, Flex



김차종 (Cha-Jong Kim)

1984년 광운대학교 컴퓨터공학과
(공학사)

1986년 광운대학교 컴퓨터공학과
(공학석사)

1991년 광운대학교 컴퓨터공학과 (공학박사)

1997년 7월~1998년 7월 Pittsburgh of University, USA

초빙교수

한밭대학교 정보통신·컴퓨터공학부 교수(현)

※ 관심분야: XML기반 Web Service, Semantic Web,
컴퓨터그래픽스, Mobile Programming