웹 2.0을 위한 Ajax기반 RSS리더 모듈 개발에 관한 연구

권영제* · 김차종**

*한밭대학교 정보통신전문대학원 컴퓨터공학과 · **한밭대학교 정보통신·컴퓨터공학부

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

Young-jae Kwon* · Cha-jong Kim**

*Graduate School of Industry & Communications, Hanbat National University

*Divi. of Information Commu. & Computer Eng., Hanbat National University

E-mail : setface@korea.com* · cjkim@hanbat.ac.kr**

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 Hanguel 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 진화한다는 차원의 차세대 웹을 뜻한다.

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

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 Simplr Sydication)[3]는 컨텐츠 신디케이션(Content Syndication)을 위해 나온 XML형태의 규격 중 하나로 웹 사이트끼리 서로 자료를 주고받기 위한 규격이다. 하지만 그러한 RSS는 해당 사이트에서 지원을 해야 쓸 수 있고, RSS전용 브라우저가 아니라면 문서의 형식이 조금은 바뀌기 때문에 원본의 형식과 품이 바뀌어져 왜곡될 수도 있다는 것이다.

본 논문은 RSS 이러한 문제점을 해결하고자 웹 2.0상의 특성을 고려하고 Ajax[4]의 유동성 있는 프로그래밍을 통한 모듈을 연구하였다. RSS Feed가 브라우저에 따라 왜곡 될 수 있는 부분을 보정하는 모듈과 한글 및 제 3세계 언어의 인코딩 문제를 해결하고, DragBox모듈과 RSS Feed의 자동 업데이트 모듈을 구현해, RSS리더 시스템이 웹2.0 기반 상에서 얼마나 효과적으로 쓰일 수 있는지를 보여주는 시스템이라 할 수 있겠다.

II. 모듈의 구성

2.1 모든 브라우저의 지원

그동안 구현자들은 인터넷 익스플로러에 맞춰서 사이트를 구현해왔다. 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 다들 익스플로러로 접속하기 때문에 구현자들도 굳이 다른 브라우저를 지원할 필요를 느끼지 못했던 것이다.

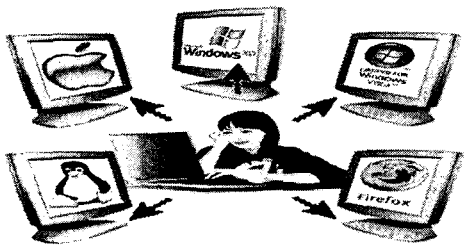

그림 4. 다양한 OS · 브라우저 사용 보장

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. 익스플로러뿐만 아니라 파이어폭스를 비롯해 모든 브라우저를 받아들이는 최선의 대안은 결국 철저하게 표준을 따르는 것이다.

2.1.1 XMLHttpRequest

XMLHttpRequest는 웹 서버와 통신하기 위한 Ajax의 핵심 컴포넌트이다. 이미 IE에서는 5.0에서부터 액티브X 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 (http://www.w3c.org)[5] 표준이 아니지만 W3C에는 XMLHttpRequest와 비슷한 DOM Level 3의 Load and Save specification이 있다. 아직 DOM Level3의 Load and Save가 구현된 브라우저는 없지만 다이나믹하게 XML문서를 읽어내서 DOM에 적용하고 DOM구조를 XML로 serialize한다는 개념은 다소 비슷한 부분이 있다.

2.1.2 XMLHttpRequest 객체

XMLHttpRequest를 얻어오는 방법은 IE와 그 이외의 브라우저가 서로 다르다.

인터넷 익스플로러의 경우에는 다음과 같은 방법으로 객체를 얻어올 수 있다.

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

그림 5. Internet Explorer의 객체 파싱

그 이외의 브라우저에서는 다음과 같은 방법으로 객체를 얻어올 수 있다.

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

그림 6. Other Browser의 객체 파싱

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 7. 일반적인 객체 파싱

2.2 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 사용해왔다. 그런데 문제는 한글이 아닌 다른 언어로 된 운영체제에서는 이 방식의 페이지를 읽지 못한다는 것이다.

이런 문제를 해결하는 대안이 바로 유니코드라고 불리는 UTF-8[6] 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 PHP파일로 송신하면 데이터는 UTF-8로 인코딩되어 전송된다.

```
var url = some_url;
var param = 'id=myid&password=mypass&name=이름';

xmlHttpPost(url,param,result function);
```

그림 8. 한글이 전송되지 않는 예

그림 5처럼 인자값에 한글이 들어갈 경우에는 제대로 전송이 되지 않는다.

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

그림 9. 한글을 전송가능하게 하는 예

그림 6의 encodeURIComponent,escape 두 함수를 이용해서 가공을 한번 해줘야 제대로 전달이 된다.

다음으로 PHP에서 위 인자값을 받을때는 그림 7과 같이 쓰인다.

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

그림 10. PHP인자값 받기

자바스크립트에서는 함수를 적용한 역순으로 decoding을 먼저 해주고 문자셋을 euc-kr로 변경해주면 된다.

2.3 일반 컴퍼넌트Box의 RIA화

일반적인 RIA(Rich Internet Application)[7]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 할수 있다는 것이 바로 RIA의 강점이다.

본 논문에서는 Ajax를 활용한 DragBox처럼 유동적인 박스를 만들어 RIA처럼 한 화면에서 구현할 수 있는 인터페이스를 구현했다. 그림 8은 DragBox의 function을 정의한 것이다.

```
function autoScroll(direction,yPos) /*박스의 스크롤 기능*/
function initDragDropBox(e)         /*박스의 위치를 잡아주는 기능*/
function initDragDropBoxTimer()     /*박스의 로딩타이머*/
function moveDragableElement(e)     /*박스의 끌기기능*/
function stop_dragDropElement()     /*박스의 끌기및놓기의 요소 */
function createColumns()            /*박스가 만들어질 배열을 만듦 */
function mouseoverHideHeader()      /*헤더위에 마우스가 위에 올라가있을때 행동*/
function mouseoutBoxHeader(e,obj)   /*헤더 위에 마우스가 있을때의 행동*/
function showHideBoxContent(e,inputObj)/*숨겨져 있는 박스의 정보를 보여줌*/
function mouseover_CloseButton()    /*마우스가 위에 있을시에 닫히는 버튼*/
function highlightCloseButton()     /*마우스가 위에 있을시에 닫히는 버튼*/
function closeDragableBox(e,inputObj)/*활수있는 박스를 닫는 기능*/
function addBoxHeader(parentObj,externalUrl,notDrabable)
/*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 11. DragBox의 Components

2.4 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 되야한다.

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

그림 12 Timer Component

그림 9는 DragBox가 생성될때 시간을 받아서 시간에 맞추어 RSS Feed를 다시 불러들이는 역할을 한다.

III. 시스템의 구현

본 모듈은 사용자가 별도의 환경설정후 사용할 수 있도록 PHP를 사용하여 구현하였다.

3.1 시스템의 구현

본 시스템은 그림 10의 입력폼에서 입력받은 RSS주소와 RSS의 개수, 업데이트시간, 박스의 크기등을 추출하고 입력받은 정보중 RSS Feed정보만을 제외한 유동적인 박스를 생성하고, RSS Feed와 입력받은 정보를 그림11의 박스안에 보여준다.

그림 13. RSS Feed정보및 기타정보 입력 인터페이스

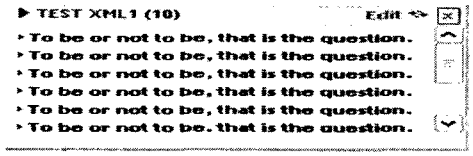

그림 14. DragBox의 초기화면

그림 12에서 보여지는 DragBox내에서 보여지는 Edit모드는 RSS Feed정보 및 보여지는 RSS의 개수와 크기 업데이트시간등 그림 10에서 보여주었던 입력 인터페이스에서 생성하였던 내용을 볼 수 있으며, 입력내용을 바꿀수 있다.

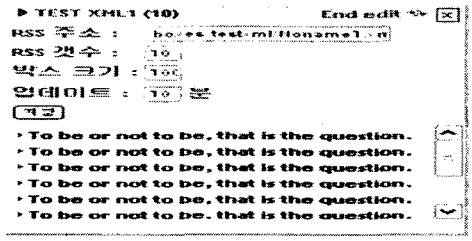

그림 15 DragBox의 Edit모드 화면

그림 13에서 보여주는 DragBox의 비활성화 화면은 그림 11의 왼쪽 위쪽에 보이는 "▶"를 클릭했을때 비활성화 되어서 화면을 잠시 닫거나 할 때 쓰일수 있다. 그림 13의 왼쪽 위쪽에 보이는 "▼"을 클릭시에 다시 그림 11처럼 펼쳐져서 내용을 볼수 있게 해준다.

그림 16. DragBox를 비활성화 화면

그림 14에서는 RSS Feed Reader시스템을 3개 OS의 대표적인 브라우저(Apple Mac-Safary, RedHat Linux- Mozilla, Microsoft XP-Internet Explorer)를 대상으로 테스트를 해보았고, 보여지는 RSS Feed데이터가 정상적으로 보여지는것을 볼수 있다.

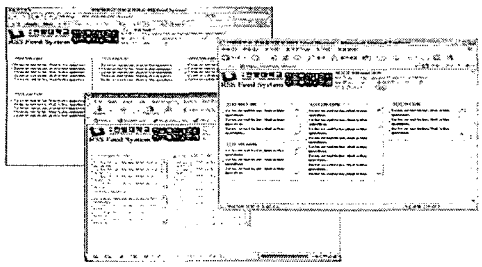

그림 17. 각 브라우저별 실행화면

Ⅳ. 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 애플리케이션이 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 제 3세계 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] Ben Hammersley, RSS and Atom, O'Reilly, 2005.

[4] adaptivepath, "Ajax: A New Approach to Web Applications", http://www.adaptivepath.com/publications/essays/archives/000385.php,2005.

[5] W3C, "DOM Level 1 Specification", http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001, 1998.

[6] The Unicode Condortium, The Unicode Standard, Version 2.0, 1996.

[7] Michał Małaj, "RIA - Rich Internet Application", http://flex2.blogspot.com/2006/09/ria-rich-internet-application.html, 2006