

AJAX기술을 이용한 사용자 맞춤 인터페이스 웹 어플리케이션 개발*

이돈수[○] Nguyen Thi Nhung 신정훈 이상준
송실대학교 컴퓨터학부

donsu.lee@gmail.com, ntnhung24@gmail.com, coolhoony81@ssu.ac.kr, sangjun@ssu.ac.kr

Development of Web Application for Customized User Interface using AJAX Techniques

Donsu Lee[○], Nguyen Thi Nhung, Junghun Shin, Sangjun Lee
School of Computing, Soongsil University

요 약

웹 2.0의 도래와 더불어 웹 문화의 변화가 급속도로 변하고 있다. 기존의 정보 이용자는 정보 제공자가 제공하는 정보를 단순히 받아보는 수동적인 입장에서, 사용자가 정보를 창출하고 변화를 이끄는 1인 미디어의 시대가 왔다. 이러한 정보의 홍수 시대 속에서 우리가 원하는 정보의 요구량은 하나의 웹 페이지의 정보만으로는 충족될 수 없다. 이에 일률적이고 비슷한 이미지와 정보 배치를 보여주는 웹 페이지들을 좀 더 자신의 개성을 살려 표현하고, 사용자에게 맞춰진 정보들로 구성할 수 있는 맞춤형 개인화 웹 어플리케이션을 제안한다.

1. 개 요

웹에는 수많은 정보들이 다양한 형태로 산재되어 있다. 각종 포털사이트의 메인 페이지에는 수많은 정보가 있지만 사용자 입장에서는 불필요한 정보가 대부분이다. 그로 인한 정보의 가독성마저 저하된다. 한편 인터넷과 정보통신 기술의 발달은 콘텐츠의 제작 주체를 개인의 영역까지 확장시킴으로써 1인 미디어의 시대와 더불어 콘텐츠의 홍수 시대에 우리는 살고 있으며, 원하는 정보의 요구량을 하나의 웹페이지에서 모두 충족시킬 수 없다.

본 논문에서는 다양한 웹페이지에 산재해 있는 정보를 모아 하나의 페이지에 표현함으로써 사용자가 원하는 정보를 쉽고 빠르게 접할 수 있도록 한다. 또한 블로그, 미니홈피, twitter 등에서 볼 수 있는 자신만의 개성을 살린 웹페이지를 꾸미고 제작하는 서비스의 요구가 점점 증대되고 있는 현실을 반영하여, 일률적인 웹 페이지를 지양하고 편리한 인터페이스와 사용자에게 맞춤형 개인화 페이지[1]를 제공하는 웹 어플리케이션을 제안한다.

2. 관련 연구

2.1 Open API 현황

국내 각 업체별 Open API의 특징적인 항목을 살펴보면 다음과 같다. 네이버는 방대한 네이버의 지식IN 검색, 현재 시간의 네티즌 의견을 보여주는 실시간 검색어, 이미지 검색, 동영상 검색, 지도 API가 있다. 다음의 경우 카페 검색, 뉴스 검색, 설명문까지 출력되는 일본어 사전이 있고, 알라딘의 경우 자사 서비스에 축적된 모든 상품을 노출하고 있다.

해외를 살펴보면 사용자들이 붙이는 태그인 Flickr[2]나 del.icio.us[3], 사용자의 인터페이스인 검색창의 추천검색어, 검색로봇이 수많은 웹페이지를 돌아다니며 링크를 읽어들이 이를 바탕으로 데이터의 우선순위를 나타내주는 Google의 페이지 랭크나 아마존의 도서 리뷰 시스템, e-bay의 reputation 시스템 등도 웹 2.0의 특징을 잘 나타내 주는 대표적인 예이다.

Open API는 외형적으로 자신의 서비스 안에서 Open API를 제공하는 곳의 콘텐츠나 기능을 구현할 수 있는 장점을 가진다. Open API가 없다면 해당 서비스 사업

* 본 연구는 서울시 산학연 협력사업(10581C093115)의 지원에 의해 수행되었음

자와 직접 계약을 맺어 서비스를 제공하거나, iFrame과 같은 방법으로 구현하여야 한다. 기존의 기능을 확장한 Mash-Up 서비스 또한 가능하다.

2.2 Open API 사용

Open API[4]를 사용하기 위해서는 그림 1과 같이 GET/POST 방식으로 해당 서비스 URL에 요청해야 한다. 이때 일반적으로 각 서비스에서 할당 받은 Key 값을 전달해야 한다. 다음과 네이버의 경우 해당 서비스의 Query를 하루에 5,000회로 제한하고 있는 부분도 실제 서비스 구축에 주의해야 하는 부분이다.

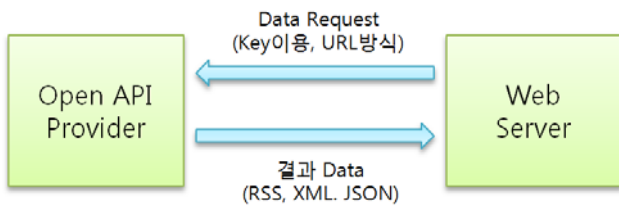


그림 1. Open API 데이터 요청 흐름

2.3 Open API 데이터 형식

Open API의 실행결과는 보통 RSS, XML, JSON 방식 중 하나로 반환해 준다. 네이버의 경우 RSS 형식으로만 결과 값을 반환하고, 다음의 경우 RSS, XML, JSON을 모두 지원하며, 알라딘의 경우 RSS와 JSON을 모두 지원한다. 이들 OpenAPI가 공통적으로 지원하는 반환 형식은 그림 2와 같은 RSS 형식이다[5].

```
<?xml version = "1.0">
  <rss version = "2.0">
    <channel>
      <title> 타이틀 제목 </title>
      <link> 타이틀 링크 </link>
      <description> 설명 </description>

      <item>
        <title> 아이템 제목 </title>
        <description> 아이템 설명 </description>
        <link> 아이템 링크 </link>
      </item>
    </channel>
  </rss>
```

그림 2. 기본적인 RSS의 형식

2.4 Web 기반의 위젯

기존의 위젯은 대부분 Application 기반으로 이루어져 있어 파일을 직접 다운받아 설치하여야만 위젯 서비스를 이용할 수 있다[6]. 이는 매우 번거롭고, 프로그램의

설치, 삭제가 빈번하게 일어난다. Web 기반으로 위젯 서비스를 제공한다면 그림 3과 같이 소프트웨어 설치와 필요 없을 뿐만 아니라, 개인화된 Web 페이지에 손쉽게 추가 삭제를 할 수 있다.

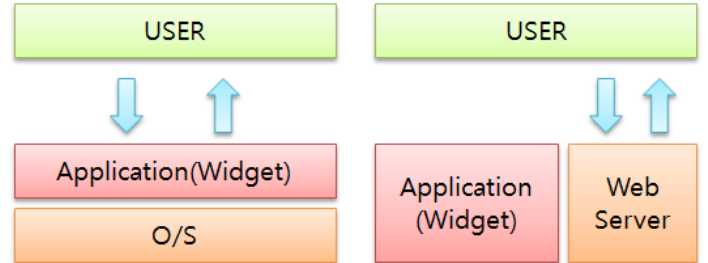


그림 3. 기존의 OS 기반의 Application Widget(좌)/ Web Server 기반의 Widget(우)

3. 제안 시스템의 설계 및 구현

3.1 시스템 구성도

웹페이지는 HTML, JavaScript를 기반으로 AJAX기술을 사용하여 구성하고, 웹서버의 OS는 Windows XP이며, Apache, PHP를 이용해 구축하였다. 데이터베이스는 My-SQL을 이용하였다. 전체 시스템의 구성은 그림 4와 같다.

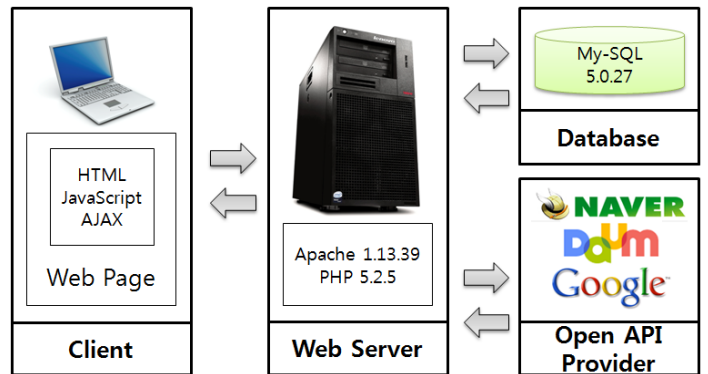


그림 4. 전체 시스템의 구성

3.2 AJAX(Asynchronous Javascript And XML)

AJAX 기술[7][8]을 이용하여 사용자가 버튼을 클릭하면 JavaScript 및 DHTML을 사용하여 즉시 UI를 업데이트할 수 있고, 비동기 요청을 서버에 전달하여 업데이트를 수행하거나 데이터베이스에 Query할 수 있다. 요청이 반환되는 경우에는 그림 5와 같이 JavaScript 및 CSS를 사용하여 전체 페이지를 새로 갱신하지 않고 UI를 업데이트 할 수 있다[7].

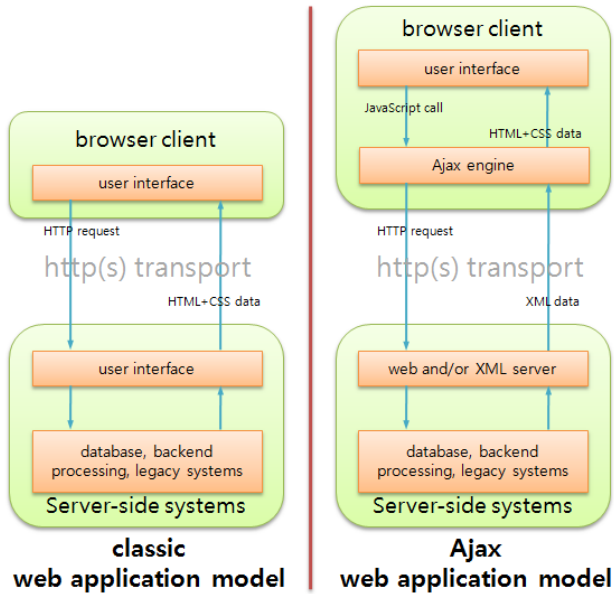


그림 5. 전통적인 웹 어플리케이션 모델(좌)/ AJAX 기술을 적용한 웹 어플리케이션 모델(우)[7]

제한한 시스템에서 AJAX의 구현은 XMLHttpRequest 객체를 통해 이루어지며 이러한 과정은 그림 6에 나타나 있다. 웹 양식에 입력할 때 데이터는 직접 서버 스크립트에 전송되지 않고 JavaScript 코드에 전달된다. 대신 JavaScript 코드는 데이터를 포착해 Request를 서버에 전송한다.

```
function createXMLHttpRequest(){
    if(window.ActiveXObject){
        try{
            //win ie.5, ie.6, ie.7
            return new ActiveXObject("Msxml2.XMLHTTP");
        }catch(e){
            try{
                return new ActiveXObject("Msxml2.XMLHTTP");
            }catch(e){
                return null;
            }
        }
    }else if(window.XMLHttpRequest){
        //ie, safari, kconqueror, firefox, nescape, opera
    }else{
        return null;
    }
}
```

그림 6. XMLHttpRequest객체 생성 함수

3.3 Web Slicing

Web Slicing은 웹페이지에서 원하는 부분을 지정하여 저장하는 기능이다. 저장된 부분은 자신의 웹페이지로

불러올 수 있다. 마우스 드래그를 통해 좌표 값을 저장하고 <div>를 통해 원하는 크기로 화면을 출력하며 그림 7은 <div>의 예제를 보여주고 있다.

```
<div id="layer1" style="width:72px; height:21px; position:absolute; left:125px; top:145px">
    <p></p>
</div>
```

그림 7. Web Slicing 예제

그림 7에서 볼 수 있듯이 <div>를 통해 웹문서의 구역이 설정된다. 이때 절대위치로 구역을 설정하게 되는데, 위치의 기준은 웹문서 자체로 한다. 여기서 상대위치로 <div> 태그가 속한 테이블이나 다른 <div> 태그를 기준으로 한다. 그림 8 및 그림 9는 Web Slicing의 과정 및 실제 운영 화면을 보여주고 있다.

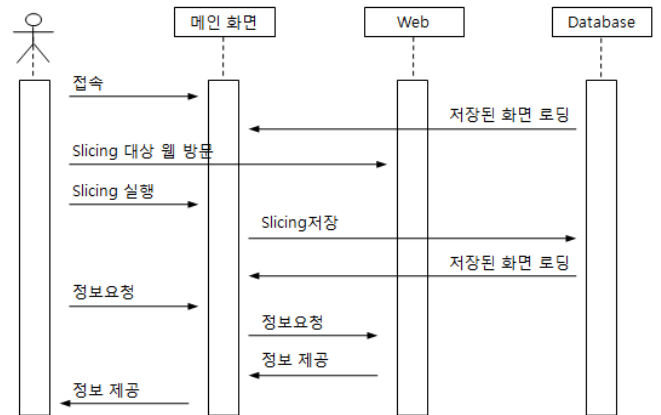


그림 8. Web Slicing Sequence Diagram



그림 9. Web Slicing 운영 화면

3.4 Web 위젯

본 시스템은 시계, 바이오리듬, 메모장, 날씨, 검색, 뉴스 총 6가지 위젯을 제공한다. 사이드 바에서 로드하여 원하는 위치에 놓으면 바로 설정이 가능하다. 부가적인 기능으로는 웹의 바탕화면을 자신이 원하는 그림을 업로드하여 적용시킬 수 있다. 또한 자신이 보유한 mp3 파일을 서버에 업로드하여 음악 재생기를 통해 음악을 들을 수 있다. 한편 히스토리 기능을 통하여 위에서 열거한 모든 기능을 저장하여 개인화 페이지를 추후 로드할 수 있다. 그림 10은 본 시스템에서 제공하는 여러 기능을 적용시킨 화면이다.



그림 10 Web Slicing 및 Widget 실행 화면

4. 결론

웹2.0의 시대의 도래와 더불어 우리는 정보의 홍수시대 속에서 살고 있다. 또한 기존의 수동적인 서비스를 지양하고 서비스의 내용에 의미를 부여하여 이를 사용자의 요구와 사용자의 설정에 맞춰 필요한 정보를 제공하는 행태의 서비스에 대한 관심이 증대되고 있다.

본 논문에서는 사용자에게 편리한 인터페이스와 함께 사용자 맞춤형 개인화 페이지를 제안하였다. 제안된 웹 어플리케이션의 장점은 별다른 설치 없이 인터넷이 연결된 PC에서 동일한 환경을 구축할 수 있다는 점이다. 이점에 착안하여 자신만의 개인화 페이지를 통해 다양한 정보에 능동적으로 대처할 수 있고, 편리한 인터페이스를 제공할 수 있는 웹 어플리케이션을 구현하였다. 사용자에게 익숙하고 간편한 인터페이스는 획일화된 정보제공자의 형식에서 벗어나 사용자 중심의 서비스를 제공한다. 이는AJAX 기술을 적용하여 더 빠른 속

도로 사용자에게 다양한 서비스를 제공할 수 있다. 향후 사용자에게 더욱 편리한 인터페이스에 대한 지속적인 관심과 더불어 자유도를 높인 시스템 개발 및 구현이 추가적으로 필요할 것이다.

참고문헌

- [1] Y. Jeon, E. Hwang, "Personalizing Web Service Pages for Mobile Users", Journal of KIISE: Computing Practices, Vol.11, No.1, pp.69-80, 2005 (in Korean)
- [2] <http://www.delicious.com/>
- [3] <http://www.flickr.com/>
- [4] http://en.wikipedia.org/wiki/Open_API
- [5] 권영제, 김차중, "웹2.0을 위한 AJAX 기반 RSS 리더 모듈 개발에 관한 연구", 한국해양정보통신학회 논문지, Vol.12, No.7, pp.1263-1270, 2008
- [6] 위젯(Widget) 서비스와 발전방향, KT경제경영연구소, 2008
- [7] Ajax: A New Approach to Web Applications. Adaptive Path. 2005-02-18. Retrieved 2010-04-24.
- [8] K. Smith, "Simplifying AJAX-Style Web Development", Computer, Vol.39. No.5, pp. 98-101 2006