

Ajax 기반의 위치추적 서비스 설계 및 구현

이주래*

*고려대학교 컴퓨터 정보통신대학원 소프트웨어 공학과
e-mail : kain5512@korea.ac.kr

Design and Implementation of an Ajax based Location Tracking Service

Ju-Rae Lee*

*Dept. of Software Engineering, Graduate School of Computer and Information Technology,
Korea University

요 약

개인용 컴퓨터와 인터넷, 그 밖의 다양한 기술의 발전으로 사람들은 일상마저 데이터화 되어 관리되고 보여지게 되었다. 그 중에서 위치 기반 서비스는 점점 일상화되어 웹 사이트를 통해 개인 별 위치추적까지 서비스되고 있다. 더 많은 정보를 원하는 사람들의 욕구와 소프트웨어의 발전은 RIA(Rich Internet Application) 라는 새로운 트렌드를 만들었고, 기존의 HTML 에서는 불가능 했던 동적이면서도 강력한 기능을 제공하는 기술들이 늘어가고 있다. 본 논문은 Ajax 를 이용하여 별도의 프로그램 설치 없이 동적이면서 풍부한 이동체의 위치추적 서비스를 설계 구현한다.

1. 서론

위성 기반의 주파수 항법 시스템(GPS) 과 RFID, UWB 와 같은 무선 주파수 기술의 위치 감지기능 기술의 발전에 따라 위치기반 서비스(LBS: Location Based Service)는 점점 일상화 되고 있다. 군사, 물류, 항법 등에서 사용되던 위치추적은 미아·도난방지, 네비게이션 등으로 우리 생활에 보다 가까운 목적으로 사용되고 있다. 또한 위와 같은 심각한 목적이 아닌, 위도와 경도가 정수로 만나는 모든 지점의 사진과 정보를 수집하는 디그리 컨플루언스 프로젝트(Degree Confluence Project) 나 GPS 좌표를 보고 보물을 찾는 지오키링(Geocaching) 대회 등의 재미를 목적으로 하는 사이트들도 큰 인기를 누리고 있다[1].

또한 소프트웨어 기술의 발전에 따라 사람들은 더욱 많은 정보를 원하고 더욱 쉽게 보기를 원하게 되었다. 이러한 사람들의 생각에서 만들어진 최신 트렌드로 RIA (Rich Internet Application) 와 SaaS (Software As A Service) 등이 있다. RIA 기술은 데스크탑 소프트웨어 어플리케이션의 강력한 기능성과 웹 어플리케이션의 접근성 등의 장점들을 포함하는 기술을 말한다. 가트너는 2010 년이면, 적어도 60%의 신규 어플리케이션 개발 프로젝트가 RIA 기술을 포함하고, 그 중의 최소 25%는 RIA 기술을 주로 사용할 것이라고 예상했다[2].

이와 같이 발전된 기술로 인해 더 많은 정보를 쉽게 구할 수 있게 되었다. 예를 들어 축구나 럭비 같이 넓은 범위에서 하는 경기에서 선수들의 위치를 추

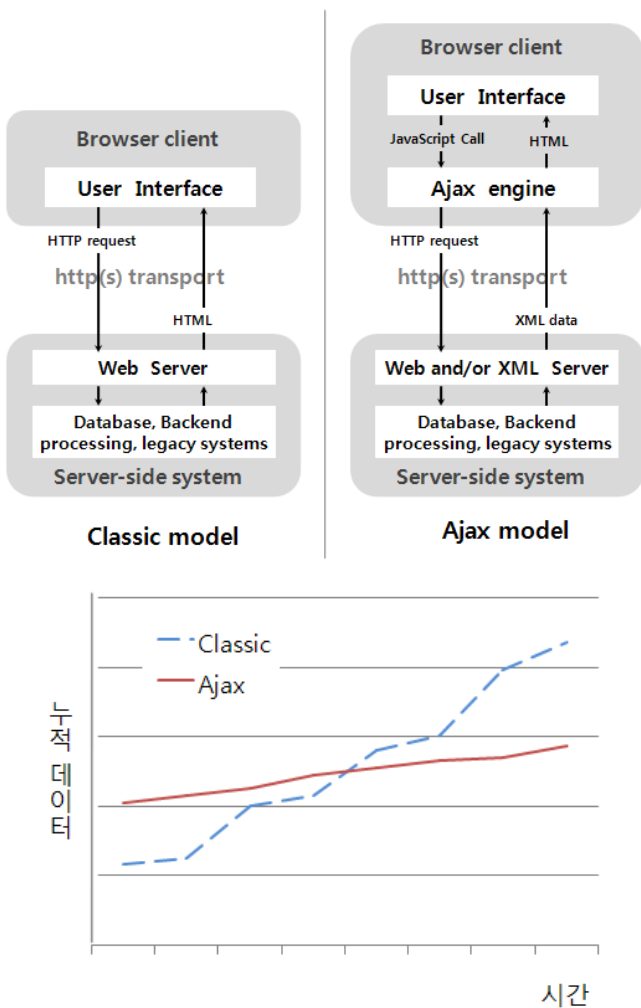
적하면, 실제 경기장에 가지 않아도 카메라에 잡히지 않는 선수들의 움직임을 알 수 있게 된다. 이와 같이 이동체의 위치를 웹 상에서 동적이고 풍부한 서비스 할 수 있는 시스템을 Ajax 로 구현해보고자 한다.

2. Ajax 기술과 활용

이전에 사용된 RIA 기술들로는 ActiveX, Flash, Java Applet 등이 있다. ActiveX 는 별도의 프로그램을 설치할 필요가 있고, Flash 는 무겁고 텍스트 기반이 아니기 때문에 수정이 어려우며 Java Applet 은 자바 가상 머신을 설치해야 한다. Ajax 는 별도의 프로그램을 설치할 필요가 없고, 빠르고 강력하며 표준과 개방성, 확장성 또한 좋다. Ajax (Asynchronous Javascript and Xml) 라는 용어가 2005 년에 만들어져 알려지기 시작했고[3], 웹 2.0 과 RIA 트렌드로 인해 각광받았기 때문에 신기술인 듯 보이지만 그 각각의 핵심 기술은 이전부터 연구되고 사용되고 있던 것들이다[3]. Ajax 가 기본으로 하는 DHTML 은 1997 년, XMLHttpRequest 오브젝트는 2000 년에 이미 지원을 시작하여 사용되었다. Ajax 는 하나의 웹 페이지를 여러 개의 컴포넌트로 구성하고 비 동기 통신으로 필요한 데이터만을 가져올 수 있어 서버와 네트워크의 처리량이 줄어든다. (그림 1) 은 기존의 웹 어플리케이션 모델과 Ajax 어플리케이션 모델을 비교한 것이다. 기존의 웹 어플리케이션 모델은 사용자가 요청을 할 때마다 서버가 HTML 을 통째로 보내주었지만, Ajax 어플리케이션 모델은 처음 요청할 때에 어플리케이션의 클라이언트에

해당하는 부분을 웹 페이지와 함께 보내고, 그 이후 서버는 요청 받은 데이터만을 XML, JSON 등의 TEXT 형태로 보내면 이 어플리케이션이 HTML 형태로 구성해 사용자에게 보여진다. 두 번째 요청부터는 데이터만이 전송되므로 클라이언트와 서버간의 데이터 전송량이 줄어들고, 클라이언트 어플리케이션의 처리에 의해 사용자는 요청 후 응답이 올 때까지 다른 업무를 수행할 수 있다. 또한 서버와의 통신 없이 클라이언트에서 한번에 받은 데이터를 동적으로 표현할 수도 있고, 이미 출력된 HTML 문서를 자유롭게 재배치할 수도 있다.

(그림 1)은 기존의 어플리케이션 모델과 Ajax 어플리케이션 모델의 구동 방식과 시간에 따른 누적 데이터양의 비교이다[7].



(그림 1) 기존의 웹 모델과 Ajax 모델 비교

Ajax 는 현재 다양한 서비스를 하고 있는데, 대표적인 서비스로는 구글 맵(Google Map)과 제시어 기능 등이 있다. 계속해서 변하는 주식사이트나 EMS (Enterprise Management System)도 Ajax 기술을 사용하여 더욱 동적이고 풍부한 기능을 제공한다. MS Office 나 msn 메신저와 같은 프로그램을 설치하지 않고도 문서 작업이나 메신저를 사용할 수 있다[4]. Ajax 는 기본적으로 JavaScript 를 이용하여 유연성이

매우 높아, 공개 API 를 이용하는 것도 매우 쉽고, 여러 API 들을 혼합하여 새로운 서비스로 활용할 수도 있는데, ‘구글 야후 교통 날씨 지도’에서는 구글 맵, 야후 교통정보, 월드웨더(worldweather) 날씨정보를 결합하여 지도를 통해 교통상황과 날씨를 함께 확인할 수 있다[5].

3. 위치추적 서비스 설계 및 구현

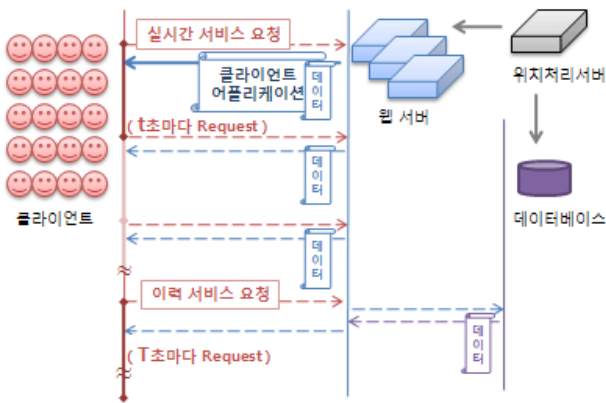
Ajax 가 기존의 웹 서비스 방식보다 적은 부하를 가지기는 하지만 그 한계를 벗어날 수는 없기에 추적 대상과 클라이언트의 수가 프로세스의 중요한 요소로 작용한다. 본 논문에서 구현한 위치추적 서비스는 서론에서 예를 든 축구선수 위치추적 서비스로 매우 많은 클라이언트들이 모두 같은 데이터를 보는 방송과 같은 서비스이다. 실시간 위치추적과 이동경로추적을 모두 서비스하며 클라이언트에서 초당 N 회의 새로운 데이터를 보여준다. 실시간 위치추적 서비스일 경우, 클라이언트가 초당 N 회씩의 데이터를 요청한다면 비동기 통신을 한다 해도 서버의 부하가 클 것이므로 t 초 마다 요청을 보내고, 위치이력 서비스일 경우, 그보다 긴 시간 T 초마다 요청을 보낸다. 시스템은 다음과 같이 동작한다.

1) 각 선수들의 식별코드와 위치 데이터는 수신기를 통해 초당 N 번씩 위치처리 서버로 들어오고, 위치처리 서버는 위치 데이터를 서비스 제공에 필요한 형태로 가공한 후, 각 선수들의 데이터들의 시간 동기를 맞추어 데이터베이스와 웹 서버들로 전송한다. 서버와 클라이언트의 통신은 비동기지만, 데이터의 시간 동기는 매우 중요하다. 이동체끼리의 시간 동기는 위치처리 서버에서 데이터를 추출할 때 맞추어 주고 연속된 응답으로 온 데이터를 자연스럽게 연결해 주기 위한 데이터들의 시퀀스에도 위치처리에서 설정한 절대시간을 이용한다.

2) 웹 서버는 파일에 데이터를 실시간으로 떨어뜨리고, 그 데이터는 일정 시간 동안만 유지한다. 기본적으로 실시간 서비스는 웹 서버 파일의 데이터를, 이력 서비스는 데이터베이스의 데이터를 사용한다.

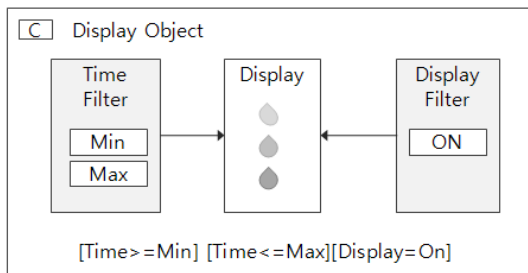
3) 실시간 위치추적 서비스를 요청할 경우, 클라이언트는 요청 후 t 초마다 웹 서버에 요청을 보낸다. 웹 서버로부터 받은 데이터를 1/N 초마다 뿌려주고, t 초가 지나기 전에 다시 요청을 보내 다음에 연결된 데이터를 받는다.

4) 위치이력 서비스를 요청할 경우, 웹 서버는 클라이언트로부터 요청 받은 시점부터 T 초만큼의 데이터를 데이터베이스에 요청한 후, 데이터베이스로부터 받은 데이터를 클라이언트에 보내준다. 클라이언트는 1/N 초마다 화면에 뿌려주고 이후 T 초마다 웹 서버에 동일한 요청을 보낸다.



(그림 2) 구현한 위치추적 시스템

RIA 설계의 특징은 서버 측과 함께 클라이언트 측을 고려하여 설계해야 한다는 점이다[6]. 데이터의 조작을 클라이언트가 XML 이나 JSON 형태로 저장해놓고, 서버와의 접속 없이 클라이언트에서만 할 수 있다. 클라이언트는 가장 널리 사용되고 있는 RDBMS에 비해 조작하기에 편리한 객체로 저장이 가능하기에 서버와는 다른 데이터 모델이 필요하다. 또한 데이터를 조작하는데 있어 얼마만큼 서버에서 조작하고, 얼마만큼 클라이언트에서 조작하는가 하는 작은 단위의 하이퍼텍스트 모델링 또한 필요하다. 본 시스템은 다수의 클라이언트에게 실시간 서비스를 목적으로 하기에 데이터는 XML 보다 전송량이 적은 JSON 을 사용하였다. 또한 클라이언트 별 추적 이동체의 선택이 가능하지만, 실시간 서비스 시 DB 접속 등의 클라이언트 별 데이터 전송을 하지 않고 모든 데이터를 전송한 뒤 클라이언트 측에서 필터링 시킨다. 아래 (그림 3) 은 추적 시간과 화면에 표시할 이동체의 클라이언트 필터링을 표현했다.



(그림 3) 클라이언트 필터링

Ajax 를 사용함으로써 사용자가 가장 크게 느끼는 부분은 동적인 화면일 것이다. 웹 브라우저에서 TV 화면에서 잡히지 않는 선수의 움직임을 1/N 초 단위로 보여주어 전체의 움직임을 파악하기 쉽다. 누구인지 궁금한 대상에 마우스를 올리기만 하면, 위치데이터가 표현되는 것에 영향을 주지 않고 해당 선수의 데이터를 가지고 와 뿌려줄 수 있다. (그림 4)은 구현한 시스템의 화면이다. 경기 모니터링 화면이나 선수의 상세 정보가 나오는 곳의 위치도 사용자가 브라우

저 상에서 자유자제로 정할 수 있도록 여러 개의 영역으로 나누어 각각 드래그 앤 드랍 으로 움직일 수 있도록 만들었으며, 주 화면에 이동체의 모습이 끊김 없이 보여지며, 각 이동체의 디스플레이 여부를 나타내는 영역을 하나씩 두었다. 각 이동체를 마우스 오버 시키면 간략한 데이터가 나오고 이동체나 디스플레이 리스트를 클릭하면 상세정보가 보여진다. 또 위치추적 중 특정 데이터를 볼 수 있는 영역도 있다.



(그림 4) 모니터링중인 화면

본 논문에서는 Ajax 위치 추적을 보여주는 기본적인 서비스 구현하였는데, 다양한 Ajax 기술을 더욱 활용하면 사용자 편의성을 매우 높일 수 있다. 좋아하는 선수를 잘 구별하기 위하여 특별히 다른 색으로 표현하거나 자기 설정을 저장할 수도 있고, 위치 이력 서비스 시 원하는 속도로 경기를 재생하는 것도 어렵지 않다.

4. 결론 및 향후 과제

위와 같이 Ajax 기술을 이용해 현재 많은 관심을 가지고 있는 위치기반 서비스를 구현하였다. 브라우저가 동작하는 동안 최소한의 데이터만을 비동기 통신으로 가져와 서버의 부하를 줄이고, 기존의 웹 서비스에서는 제한되었던 동적이면서도 풍부한 서비스를 제공함을 볼 수 있었다. Ajax 자체는 부하를 줄이는 기법이지만, 이 시스템 자체는 부하간 큰 서비스 이기에 실제적인 서비스를 위해서는 대용량 데이터 처리에 있어 서비스 할 초당 횟수와 클라이언트의 수, 위치추적 대상의 수 등을 고려해 버퍼링 시간을 조정하거나 서비스 할 초당 횟수를 조정하는 기능의 자동화와 로드밸런싱 대한 연구가 필요하다.

참고문헌

- [1] Peter Morville, “Ambient Findability”, Book, 2005
- [2] Gartner, “Rich Internet Applications Are the Next Evolution of the Web”, Research, 2005
- [3] Jesse James Garrett “Ajax: a New Approach to Web Applications”, Essay, 2005
- [4] <http://www.meebo.com>
<http://www.zoho.com>
- [5] <http://www.traffic.poly9.com>
- [6] A Bozzon, S Comai, P Fraternali, GT Carughi, “Conceptual modeling and code generation for rich internet applications”, Essay, 2006
- [7] dave crane “Ajax in Action”, Book, 2005