

# Ajax 를 이용한 감시제어시스템 설계구축

이종구

고려대학교 디지털정보공학과

e-mail : [ljk4810@korea.ac.kr](mailto:ljk4810@korea.ac.kr)

## Design and Construction of SCADA System using Ajax

Jong-Gu Lee

Dept. of Digital Information Engineering, Korea University

### 요 약

인터넷 및 네트워크 환경의 비약적인 발전으로 기업의 모든 기반 Application 은 사용자의 편리함과 시스템 및 데이터관리의 용이성 때문에 웹 기반모델로 변경되어 사용되고 있다. 그러나, 산업현장설비를 실시간 원격제어 감시하는 시스템인 SCADA(Supervisory Control and Data Acquisition)는 웹의 여러 한계성 때문에 아직까지 기존 Client-Server 모델로 대부분 활용되고 있다. 본 논문에서는 이러한 웹의 한계성 중 Client 시스템의 실시간 통신 방식 및 속도, 부하 등을 기존 Client-Server 방식과 동일한 성능을 갖도록 설계 구현하여 적용하고 성능을 평가 하였다.

### 1. 서론

Ajax(Asynchronous JavaScript and Xml)는 기존 웹의 한계성을 극복하는데 많은 역할을 하고 있으며, 이를 활용해서 서비스를 제공하는 웹사이트를 손쉽게 찾아볼 수 있다. 개발자가 새로운 언어를 습득 할 필요가 없다는 점과 Ajax 의 빠른 처리속도, 확장성, 효율성 등의 장점 때문이다. 최근에는 신뢰성과 보안성이 요구되는 금융권의 웹 기반 응용 Application 에도 사용되는 등 활용범위가 점차 확대되고 있다.

산업현장설비를 감시제어 운영하는데 필수 시스템인 SCADA (Supervisory Control And Data Acquisition) 시스템 등은 거의 대부분 Client-Server 구조로 구축운영되고 있다. SCADA 시스템의 특성상 송유관, 가스관, 전기, 철도 등 넓은 지역의 감시제어설비와 많은 Client 시스템이 필요하기 때문에 현장설비의 변동 시 업데이트와 백업문제, 장비노후에 따른 특정장비의 교체사용문제 등 여러 가지 문제점을 가지고 있다. 이러한 문제를 해결하고자 Ajax 를 이용한 웹 기반 구조의 SCADA 시스템을 제안한다.

본 연구에서 제안하는 Ajax 를 이용한 SCADA 시스템은 Client-Server 구조를 웹 기반으로 변경함으로써, Client 시스템의 효율적인 업데이트화 백업문제, 특정장비의 사용문제 등은 구조적으로 해결하였고 그에

따른 Client 시스템과의 통신속도 및 부하문제 등에 대한 문제점을 해결하였다.

본 논문의 구성은 다음과 같다. 2 장에서는 Ajax 의 기술과 적용사례 등을 살펴보고, 3 장에서는 Ajax 를 이용한 신뢰성 있는 SCADA 시스템 설계 및 구축방법을 제안하고 4 장에서는 기존의 Client-Server 구조의 SCADA 시스템과 Ajax 를 이용하여 구축한 시스템과의 속도, 부하 및 효율성 등에 대한 연구결과를 비교한다. 5 장에서는 본 논문에서 제안한 모델에 대해 기대효과와 향후 연구과제를 제시한다.

### 2. Ajax 의 기술과 적용사례

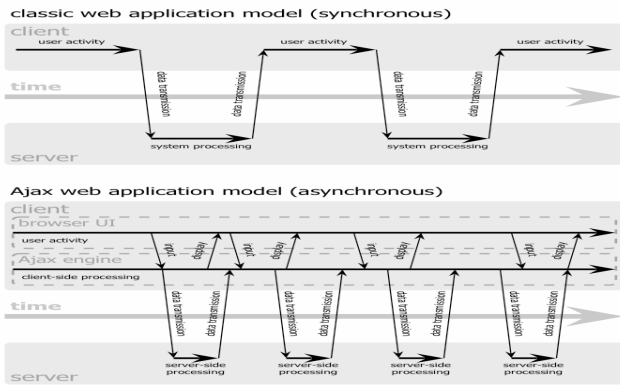
Ajax(Asynchronous JavaScript and Xml)는 HTTP Client 를 이용하는 것으로 2005 년부터 사용되었으며, 짧은 시간에 Web2.0 과 RIA(Rick Internet Application)같이 동적이면서 풍부한 데이터를 필요로 하는 웹 application 을 제작하는데 급속히 확산되었다.

Ajax 를 구성하는 4 가지 요소는[1] XMLHttpRequest, DOM, CSS, JavaScript 등이며, 기존의 일반 웹이 절차적이라면 Ajax 를 이용한 웹 Application 은 객체지향적이라 할 수 있다.

Ajax 의 주요한 특징을 살펴보면 다음과 같다. 첫째 작은 규모의 서버 이벤트가 발생한다. Client 에 적용되는 기술로 서버 처리량 및 부하가 적고, DOM 트리

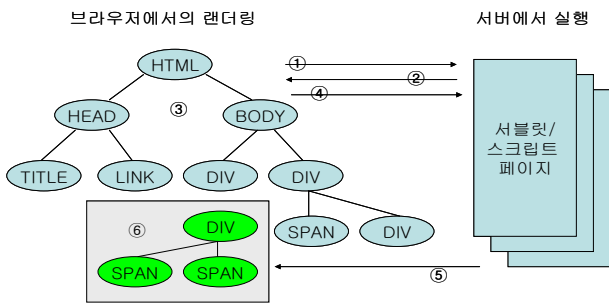
조작으로 페이지 모양변경 가능하며 전체 페이지의 Refresh 가 필요 없다. 둘째, 비동기식 통신을 사용한다. 서버로 요청 시 Client 의 브라우저가 다른 일을 수행 할 수 있으며, 실제 데이터 처리시에도 다른 부분을 계속 사용 할 수 있다. 셋째, 별도의 프로그램이 필요치 않다. 운영체제가 제공하는 사용자 이벤트와 동등한 수준으로 수행하며 별도의 프로그램 설치나 배포가 필요 없다.

Ajax 의 주요한 특징인 비동기 통신방식[2]은 전통적인 웹 Application 모델과는 확연히 다르다. 그림 1 과 같이 기존 웹 통신방식은 user 가 이벤트를 요청하면 서버시스템이 반응하는 절차이고, 비동기 통신인 Ajax 의 통신방식은 Client 시스템의 브라우저의 백프로세스로 항상 가동 중이므로 정해진 시간에 서버와 통신을 하도록 하여 많은 데이터를 효율적으로 처리할 수 있게 한다.



(그림 1) 동기 비동기 통신 모델

Ajax 를 사용한 페이지 구동절차를 살펴[3]보면 요청한 데이터에 대해 Client 브라우저 페이지에 렌더링한 후 DOM 트리 조작으로 전체 페이지의 변경 없이 필요로 하는 일부 화면만 비동기적으로 처리가 가능하다.



(그림 2) Ajax 페이지 구동 절차

- ① 사용자가 지정한 URL 에 초기 요청을 요구
- ② 서버는 원본 HTML 페이지 반환
- ③ Client 브라우저는 페이지를 메모리상에 DOM 트리로 생성
- ④ 사용자의 비동기 명령을 해도 기존 DOM 트리에 영향 없음
- ⑤ 브라우저는 페이지 내 콜백 함수를 통해 데이터를 수신
- ⑥ 브라우저는 결과를 해석하고 새로운 데이터를 이용해 메모리 내 DOM 트리를 갱신하면서 페이지를

다시 그림

### 3. Ajax 를 이용한 신뢰성 있는 SCADA 시스템 설계 및 구축 방법

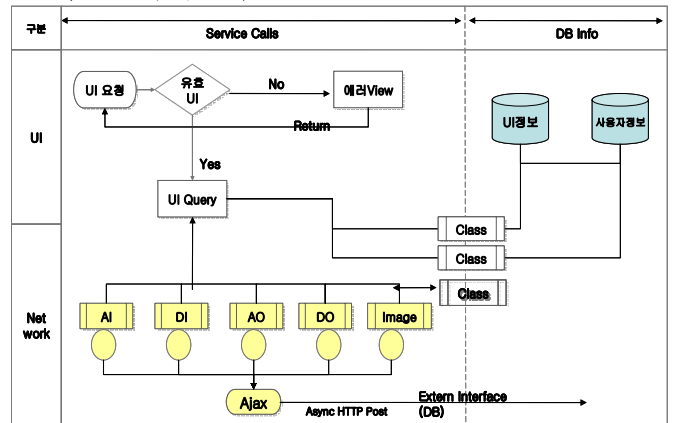
기존 Client-Server 구조의 SCADA 시스템은 다음과 같은 문제점을 가지고 있다.

첫째, 많은 수의 Client 장비에 대한 업데이트와 백업문제가 있다. 이는 Client-Server 방식에서 구조적인 문제이지만, 산업현장설비를 제어 모니터링 하는 중요한 Client 화면을 만약 업데이트가 안 되었을 경우는 심각한 문제가 발생할 수 있다.

둘째, 장비 노후에 따른 교체 시 특정장비의 사용 및 속도, 호환성에 문제가 있다. 대부분의 산업현장설비의 경우 설치가 되면 시스템의 오류가 발생할 때까지는 교체하지 않는다. 그러나, 이 기간이 너무 길어져서 해당장비가 공급이 불가능한 경우나 장비의 속도 및 호환성문제가 교체 시에는 불가능한 경우가 많다.

이를 해결 할 수 있는 방안이 Client-Server 구조의 SCADA 시스템을 Ajax 를 이용한 웹 기반 시스템으로 재설계하고 구현하는 것이다. Ajax 기반으로 재 설계함에 있어 고려되어야 할 몇 가지 부분이 있다.

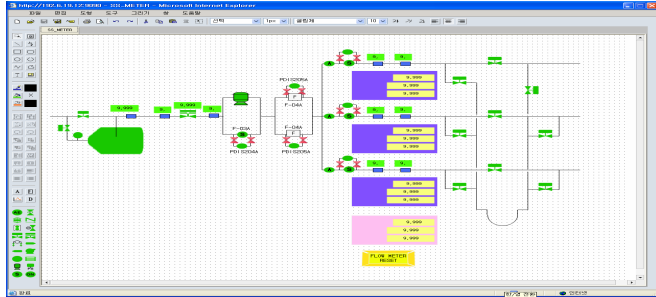
첫째 우선적으로 Client-Server 구조에서 Socket 을 이용한 안정적인 통신구조를 Ajax 를 사용하여 동일한 신뢰수준을 확보하는 것이다. 이에 기존 페이지 단위의 통신방식을 극복할 수 있는 Ajax 사용은 필수적이다. Ajax 엔진을 효율적으로 관리하기 위해서 그림 3 과 같이 Client 시스템의 구조를 크게 Network, DB, UI 3 개 영역으로 구분하였고 Network 영역에서는 다양한 현장설비 등의 특성에 맞게 5 개의 통신 프로세스로 설계하였다. 각각의 5 개 Ajax 프로세스는 HTTP 통신 프로토콜기반에서 post 방식으로 서버와 실시간 통신을 하는 역할을 수행한다.



(그림 3) Ajax 기반 Client 시스템의 3 개 영역구조

둘째, 웹 구조 시스템으로 변경함으로써 Client 시스템의 응답속도와 UI 를 통한 반응속도를 확보하는 것이 주요한 고려 대상이다. 주요한 Client-Server 구조 SCADA 시스템을 살펴보면, 통신 방식은 Thread 또는 주요한 통신프로세스를 미리 fork 처리하는 pre-fork 형태를 사용하고, UI는 각 Client 마다 응용 Application 을

별도로 설치 후 운영 사용한다. Ajax 기반의 SCADA 시스템은 기본적으로 브라우저상에서 동작하기 때문에 HTML 코드를 사용하기 때문에 속도 및 효율성 측면에서는 많이 뒤진다. 이러한 문제점을 해결하기 위해서 VML(Vector Markup Language)코드를 사용해서 Client UI 화면을 디자인 했다.



(그림 4) VML 로 만든 Client UI 편집기

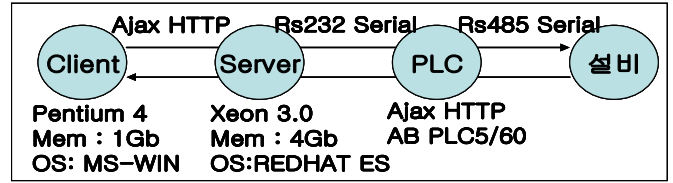
브라우저에 구현된 비트맵방식이 아닌 벡터 그래픽 구현기술로 XML 에 기반을 두고 있다.[4] 벡터의 미려한 그래픽을 그리기 위해 정의된 XML 태그를 이용해 간단하게 웹 페이지에 원하는 그래픽을 그리는 것이다. DHTML 코드를 이용해 VML 코드를 직접 다루므로 동적인 그래픽 효과가 뛰어나고 파일크기가 작으며 반응속도 또한 빠르기 때문에 마이크로소프트사의 MS Office 제품에 폭 넓게 사용되고 있다.

셋째, 브라우저 기반인 Ajax Application 에서 메모리를 가장 많이 차지하는 것은 DOM 이다. 정확히 말하면 DOM 엘리먼트이다. 효과적인 DOM 엘리먼트 관리로 최적의 메모리로 효율성을 높이기 위해서는 기본적으로 DOM 노드를 재사용하는 디자인 패턴을 적용해야 한다. 여기에 사라지면 즉시 참조 제거하는 옵션을 설정하면 최적의 메모리 상태를 얻을 수 있다.

4. Ajax 를 이용한 SCADA 시스템의 연구결과

Ajax 를 이용한 웹 기반 SCADA 시스템을 기존의 Client-Server 기반의 SCADA 시스템과 비교하여 발생하는 주요한 문제점인 Client 시스템과의 통신속도 문제, 부하 등에 대한 연구결과는 다음과 같다.

Client-Server 간 통신 방법을 Ajax 로 변경함으로써 Client 시스템의 통신속도 및 실시간 UI Display 속도를 고려하여 설계 구현한 결과 만족할 만한 결과를 나타냈다. Client 시스템의 특정 현장설비화면을 대상으로 통신테스트를 실시 한 결과, 평균 3.45 초의 반응시간을 기록했다. 기존의 시스템과 비교해도 0.25 초 내의 오차를 보이기 때문에 Ajax 를 사용함으로써 발생하는 속도의 차이는 없는 걸로 나타났다. 테스트환경은 Client 화면에서 매 1 초마다 Ajax 가 서버를 거쳐 PLC(Programmable Logic Controller)가 실제 현장제어 설비까지 제어 감시하는 전 과정이며 최대 프로토콜 크기는 8kbyt, 평균 약 3kbyte 크기로 통신을 테스트 하였다.



(그림 5) 구축한 장비현황 및 통신방식

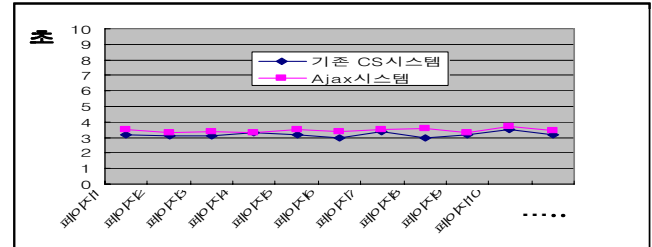


(그림 6) VML 로 제작한 Client 실행 화면

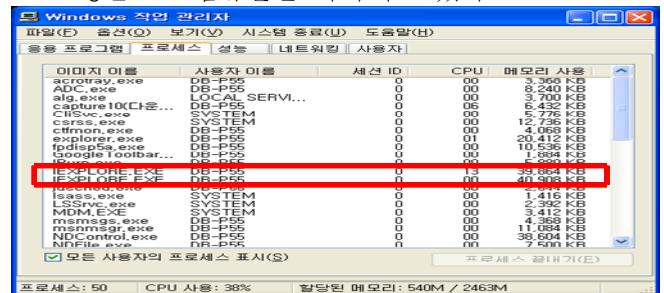
표 1 주요 기능별 평균 반응시간 (단위:초)

	기존 C/S 시스템	Ajax 시스템
로그인	0.5	0.6
Client 명령실행	3	3.1
설비모니터링	3.4	3.8
평균시간	3.2	3.45

표 2 페이지별 테스트 반응시간 (단위:초)



또한, Ajax 의 DOM 노드를 재사용하는 방식으로 관리함으로써 그림 6 과 같이 Ajax 를 이용한 Client 시스템을 실행하였을 때 최적의 부하관리가 되었다. 그림 [7]은 70 개의 현장설비를 실시간 모니터링 하는 화면으로 실제 통신하는 데이터 량은 3.5Kbyte 이다. 이 브라우저의 메모리 사용량은 그림 7 과 같이 일반적인 포털사이트의 사용량과 동일한 수준인 40Mbyte 이다. 그리고, CPU 의 사용량은 1 초 간격의 지속적인 통신으로 평균 17%점유율을 나타내고 있다.



(그림 7) Client 실행시 메모리 현황

## 5. 결론

본 논문은 Client-Server 구조의 SCADA 시스템을 Ajax 를 이용한 웹 기반으로 설계 구축을 하면서 웹의 문제점을 해결하는 방안을 제시하였고 성능테스트를 하였다. Client 시스템의 안정적이고 신뢰성 있는 통신을 위해 Ajax 기반 프로세스를 3 개영역으로 설계하고 부하를 최소화하기 위해서 DOM 노드를 재사용 하는 디자인 패턴을 적용하였다. 또한 UI 의 반응속도를 높이기 위해 HTML 기반에서 VML 을 적용 구축하여 테스트한 결과 만족한 결과를 얻었다. 이로써 현장설비를 감시제어 하는 대부분의 Client-Server 구조 시스템을 웹 기반시스템으로 변경 구축 운영할 수 있으며 이로 인해 Client 들의 업데이트, 백업 및 장비교체 등 많은 문제가 해결 될 수 있을 것으로 기대한다. 하지만, Ajax 를 이용한 대규모 데이터통신에서의 보안 및 처리방법 등에 대해서는 지속적인 연구가 필요하다

## 참고문헌

- [1] Dave Crane, Eric Pascarello, Darren James “Ajax I Action”, 2005
- [2] Jess James Garrett “Ajex”: A New Approach to Web Applications”, Essay, 2005
- [3] Justin Gethland, Ben Galbraith, Dion Almejar Programatic Ajax”, 2006
- [4] <http://www.w3.org/TR/NOTE-VML.html>