

웹과 DB를 연동한 종합 정보 관리 시스템 구현

김 범 은* 박 진 연* 원 유 집**

한양대학교 전자통신전파공학과

{candy, jypark119}@hymail.hanyang.ac.kr yjwon@email.hanyang.ac.kr

Total Information Management System on Web

Beom Eun Kim, Jin Youn Park, You Jip Won
Dept. of Electronic Engineering, Hanyang University

요약

동화댐의 MMI로부터 발생하는 고속 대용량의 데이터를 DBMS에 생성하고 이 정보들을 실시간으로 웹상에서 관리함으로서 현장에서 발생하는 모든 상황에 즉각적이고 능동적으로 대처할 수 있는 시스템을 개발하였다. 본 논문에서는 동화댐의 MMI로부터 발생하는 고속 대용량의 데이터를 실시간으로 DBMS에 생성하는 방법과, DBMS에 생성된 정보를 실시간으로 웹상에서 관리(즉 질의, 생성, 삭제 및 문서화) 할 수 있는 구현 방법에 대해서 중점적으로 서술할 것이다.

1. 서론

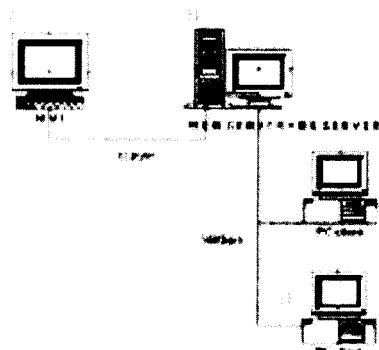
기존의 시스템은 동화댐의 MMI로부터 발생하는 댐의 대용량의 데이터를 단지 Local Machine에서만 볼 수 있었다. 그러나 인터넷의 발달과 대중화로 다양한 데이터를 실시간으로 웹상에서 관리하는 것이 가능해졌으며 또한 관리자들은 현장에 있지 않더라도 현장의 공정과정과 품질 예측 및 전체 시스템을 관찰하고 점검할 필요성이 대두되었다.

웹상에서 현장의 종합 정보 관리 시스템을 구현하는 것은 현장에서 일일이 기계의 상황을 점검하지 않더라도 현장에서와 똑같은 환경을 브라우저를 통하여 제공해 줌으로서 관리자의 시간을 절약해 줄 뿐만 아니라 현장과의 밀착감을 높이게 된다.

본 시스템의 구현 목적은 웹과 데이터베이스를 연동함으로써 웹상에서 실시간으로 데이터베이스의 정보 관리 및 측정을 자동화하고, 웹을 사용함으로써 단말의 형식과 운영체제에 관계없는 종합 관리 시스템을 구현하는 것이다.

본 시스템은 대략적인 구성은 댐의 MMI로부터 발생하는 정보들을 DB에 저장하고 Server-Client Model로 Java Applet을 구현하여 Client의 요청에 따라 DB의 정보를 Server로부터 화면에 나타내도록 설계하였다.

2. 전체 시스템 구성도



[시스템 구성도]

시스템 구축환경은 다음과 같다.

Web/DB Server는 Compaq PROLIANT 3000 Machine에 OS는 WINDOWS NT Server 4.0 + Service Pack 4.0을 설치하였고, Web Server는 IIS 4.0, DB Server는 Personal Oracle 8.0을 사용하였다. 그 외의 Server 시스템 사양은 CPU Pentium III 450, Memory 128MB, Hard Disk SCSI 2GB * 4이다.

시스템의 구성을 살펴보면 Server-Client Model을 취하여 사용자는 필요한 정보를 서버로부터 제공받는다. MMI로부터 발생한 정보들은 TCP/IP를 통해 공유 메모리에 저장되고 이 메모리의 정보 중에서 새로 업데이트 된 데이터만 다시 DB에 저장된다. DB에 저장된 정보를 실시간으로 웹상에서 관리하기 위해 Java Applet을 구현하였다. 즉 Java Client는 TCP/IP를 통해 Java Server와 DB로부터 사용자가 원하는 정보를 제공받는다. Java Client의 요청에 따라 Java Server는 DBMS와 연동함으로서 TCP/IP와 특정 Port의 소켓을 열어 원하는 정보를 주고 받는다.

3. 시스템 구현

가. MMI와 DB연동 부분

MMI(Man Machine Interface)는 산업용 Workstation급 컴퓨터 및 CRT 모니터(X-Terminal), 키보드, 마우스 등의 장치로 모든 데이터를 신속하게 감시하고 조작할 수 있는 다양한 기능을 갖추고 있다. 또한, TCP/IP로 데이터를 웹 서버에 실시간으로 전송해 준다.

1) 주기적으로 발생하는 데이터의 생성

MMI에서 발생하는 고속 대용량의 모든 데이터들은 TCP/IP를 통해 공유 메모리에 저장되고, 별도의 Process가 이 정보를 데이터베이스에 저장하게 된다. MMI에서 제공해 주는 약 2000개의 스테이션 부품들의 정보는 약 5초내로 모두 데이터베이스에 업데이트되어야 한다. 이러한 조건은 최대한 빠른 데이터베이스 업데이트를 요구하는데 처음에 ODBC를 사용해서 데이터를 DB에 업데이트 했으나 ODBC는 2000개의 정보를 5초 안에 업데이트하기에는 역부족이었다. 그래서 Oracle사에서 제공해 주는 0040(Oracle Object for OLE)를 사용하여 고속 대용량의 데이터를 짧은 시간내에 주기적으로 업데이트하도록 구현하였다. 이것은 ODBC 보다 빠른 속도를 제공해 주며, Oracle Database에 대한 세세한 제어가 가능하다.

2) 데이터의 동향을 보기 위한 데이터의 주기적 축적
스테이션 부품 데이터 중에는 데이터의 추이를 보기 위한 정보가 있다. 이러한 데이터들은 1분마다 발생하는 데이터를 약 3년간 DB에 저장할 수 있어야 한다. 운영자가 추이를 보기 원하는 데이터를 Editor를 통해서 생성하면 이 데이터의 값을 1분마다 DB에 업데이트 해준다.

3) 경보 데이터 생성

각 스테이션에서는 Alarm이 발생한다. 이 Alarm은 불규칙적으로 발생하기 때문에 원도우에서 제공하는 Event를 이용해서 Event가 발생할 때만 Alarm의 정보들을 읽어 와서 DB에 저장하게 된다.

나. DB와 WEB 연동 부분

Server-Client Model로 Java Applet을 개발하였다. Java Client를 통해서 직접 데이터베이스와 연동하는 구조를 피함으로써 클라이언트 부분에서는 데이터베이스와 연동하는 부분을 고려하지 않고 서버로부터 필요한 정보를 전송받을 수 있다. (3-Tier 구조) 따라서 다수의 클라이언트에서 데이터를 요청할 경우에 클라이언트에 걸리는 부하를 줄일 수 있으며, 클라이언트에서 데이터 베이스와 연동하는 부분을 알 수 없으므로 보안상 중요한 데이터베이스 자료구조를 숨길 수 있다.

1) DB Server 부분

데이터베이스는 1분마다 발생하는 40byte 데이터를 약 3년간 저장할 수 있기에 충분한 분량이며 또한 5초내에 최대 3000개의 데이터 업데이트가 가능해야 한다. MMI에서 많은 양의 데이터가 발생하므로 이러한 데이터베이스의 특징을 고려하여 주기적으로 발생하는 축적 데이터의 경우, 데이터간의 자유로운 연동이 가능하도록 설계되었다.

즉 데이터베이스 안에는 테이블로 데이터가 분리되어 있으나 웹상에서는 그러한 사항이 고려될 필요가 없고 또한 테이블내의 대량의 데이터가 중복되면 저장공간의 낭비이므로 이런 것을 막기위해 데이터베이스 안에는 데이터의 실제 위치에 대한 포인터를 가지고 있어서 데이터간의 그룹화, 그룹 수정, 그룹 삭제가 가능하다. 또한 웹상에서도 동일하게 보인다.

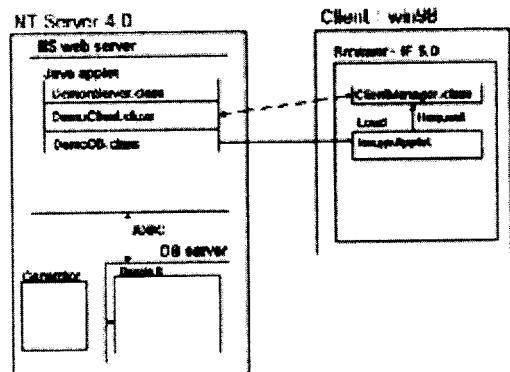
본 시스템은 특정 사용자만을 대상으로 하므로 DB Server를 동시에 사용자가 1명인 Oracle Personal Edition을 설치하였다. 따라서 범용 DB와 웹 연동 서비스처럼 사용자가 폭증할 경우를 대비하지 않는다. 본 시스템은 JDBC를 사용하여 Oracle DBMS와 연결을 설정하고 DBMS로부터 데이터를 주고 받을 수 있게 하였다.

2) Java Server 부분

Server의 역할은 특정 포트로부터 들어오는 클라이언트의 요청을 기다리는 것이다. 즉 컴퓨터상에서 동작하는 여러가지 다른 프로그램(Application)중에서 특정 Port로 연결된 자신의 데이터를 식별할 수 있다. 본 시스템에서는 Port Number로 2020을 사용한다.

클라이언트로부터 Connection이 요청되면, Server는 클라이언트가 서버에 접속한 IP와 PORT 번호를 확인하여 새로운 Socket을 클라이언트에게 할당하고 실제로 클라이언트와 통신을 전달할 새로운 Thread를 생성한다. 이 Thread는 클라이언트에게 요청을 받아서 필요한 메시지를 DB와 연동하여 주고 받는 역할을 담당하는, 다시 말해서 클라이언트와의 통신을 담당하는 클래스로서 새로운 클라이언트가 생성될 때마다 하나씩 생성된다. 즉 Server는 새로운 클라이언트가 들어올 때마다 새로운 클라이언트와 독립적으로 Transaction을 처리할 수 있도록 Thread를 새로 생성시키는 것이다. 클라이언트로부터 Connection이 종료되면 Server는 클라이언트에게 할당된 Socket을 닫는다.

3) DB와 Server 연동 부분



서버와 클라이언트간의Socket을 통해 데이터를 읽어오고, 데이터를 Socket에 써서 보내는 Server-Client 간의 프로토콜은 다음과 같다.

브라우저를 통해 Java Client 가 Web Server 로부터 Client에 로딩된다. 서버에서는 JDBC 를 이용하여 Java Server 와 DBMS 간의 Connection 을 설정한다. Client 로부터 요청정보에 대한 Command 와 부가적인 정보를 함께 보내면, Java Server 에서는 DB 와 연동하여 수행된 결과를 소켓을 통해 Java Client 로 보낸다. Java Client 에서는 데이터를 받아서 웹 상에서 실시간으로 보여주게 된다. 웹 상에서 한글 지원 문제는 다음과 같이 해결하였다. 자바의 디풀트 스트링 인코딩 방식인 KSC5601 을 오라클 데이터베이스의 디풀트 스트링 인코딩 방식인 8859_1 로 변형하여 DB에 저장하며 Java Applet 에서 한글 값을 보여주기 위해서는 이와 반대의 과정으로 스트링을 다시 인코딩해준다. 이를 보장하려면 Bypete 단위로 네트워크 상에서 데이터를 주고 받는 InputStream / OutputStream 방법으로는 문제가 발생하므로 Char 단위로 전송하는 방식인 Reader / Writer 를 이용하여 데이터를 전송한다.

다. Excel 부분

또한 웹상에서 보이는 자료를 일시적 자료로 두지 않고 Excel로 문서화 하여 저장할 수 있는 기능을 추가하였다. 이 방법은 사용자가 웹에 보이는 자료를 문서화하고자 하면 이 요청을 받은 클라이언트가 TCP/IP connection을 이용하여 사용자의 요청을 서버로 보낸다. 그러면 서버에서는 시스템 함수를 사용하여 요청된 페이지의 내용을 텍스트 파일로 생성하여 이를 엑셀로 만들어 주는 매크로를 실행시킨다. 텍스트 파일에서 읽은 값들은 Visual Basic Script를 이용하여 Excel 파일로 저장된다. 또한 사용자에게 엑셀 종료시간을 알려주기 위해서 시스템에서 파일이 생성되었을 때 리턴 값을 보냄으로써 엑셀 종료후에만 사용자가 파일을 다운로드 할 수 있게 하였다.

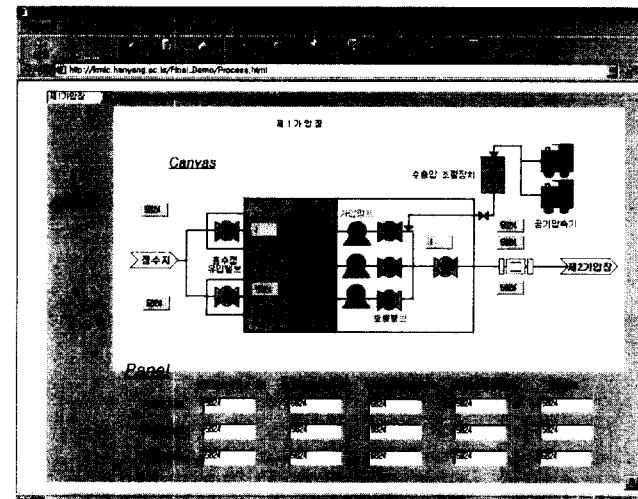
그리고 동시에 여러 사용자가 웹 화면을 파일로 생성할 경우 같은 파일 이름이 있으면 경쟁적으로 같은 자원을 서로 쓸려고 하는 문제가 생기게 된다. 한 클라이언트는 동시에 여러 번 요청을 보낼 수 없으므로 (동시가 아니면 여러 번 요청을 줄 수 있다.) 클라이언트의 IP를 파일의 이름으로 생성하였다. 따라서 경쟁적으로 같은 자원을 동시에 사용하려는 문제를 해결하였다.

이 경우 다운로드 받을 파일의 이름이 동적으로 생성되므로 Excel을 다운 받을 HTML을 동적으로 생성해야 한다. Java Script를 이용하여 동적으로 클라이언트의 IP로 새로운 HTML으로 생성하여 Excel을 다운받을 수 있게 만들어 준다.

4. 시스템 화면

Client 에서는 Java Applet 을 Server 로부터 loading 한다.

즉 Web browser 를 통해서 Web Server 에 접속한 후 Web Server 로부터 Applet을 Loading 하게 된다. Client 의 화면 구성은 Panel , Canvas 등과 같은 화면 구성요소들을 가지고 있으며 choice는 선택에 따른 적절한 동작을 하도록 제어하는 역할을 한다. 각 Applet 을 통해서 Web상에서 데이터를 생성, 추가, 삭제하고 질의할 수 있다.



[구현된 시스템 화면]

5. 향후 과제 및 결론

본 시스템의 보완해야 할 점은 Java Applet 구현에 따른 속도 문제이다. Client에서 Server로부터 Loading 하는데 걸리는 시간은 아직 사용자가 만족할 만한 수준에 이르지는 못하였다. 그러나 MMI로부터 발생하는 고속 다량의 데이터를 DB에 축적하고 이를 효과적으로 웹 상에서 보여줌으로서 또한 사용자와 데이터간의 interactive 환경을 조성함으로서 웹을 통하여 실시간으로 현장의 종합적 관리가 가능한 시스템을 개발했다고 할 수 있다.

Web과 DB를 연동하여 사용자에게 특별한 프로그램 없이 웹 브라우저의 공통된 인터페이스만을 가지고 정보를 관리하게 해 준다는 것은 상당한 매력이다. 이 분야의 개발은 앞으로 더욱 가속화 되리라 생각되며, 어디서나 필요한 정보를 볼 수 있는 것은 인터넷이 나야가야 할 바라고 여겨진다.

6. 참고 문헌

- [1] Oracle Web Page : <http://technet.oracle.com>
 - [2] MSDN (Microsoft Development Network) :
<http://msdn.microsoft.com>
 - [3] SUN Java Web Page : <http://java.sun.com>
 - [4] Campione, Walrath, Java Tutorial Second Edition,
Addison Wesley, 1998
 - [5] 김종근외, Oracle Bible ver.8.x, 영진출판사, 1999
 - [6] ELMASRI/NAVATHE, Fundamentals of Database Systems,
Addison Wesley