

가상 실험실 구현을 위한 시스템 설계

송민규^{1*}

System Design for the Implementation of Virtual Laboratory

Min-Gyu Song^{1*}

요 약 Virtual Laboratory는 네트워크 기술의 발전에 따라 제어 및 계측 분야에서 등장하게 된 새로운 개념으로서 현재 여러 대학 및 연구기관에서 폭넓게 사용되고 있다. Virtual Laboratory를 통하여 사용자는 언제 어디서든지 인스트루먼트에 대한 제어 및 모니터링을 할 수 있는데 이러한 특징은 기존의 실험실에서는 얻을 수 없었던 사용자 효율성 및 향상된 성능을 제공한다. 본 논문에서는 먼저 이러한 Virtual Laboratory의 개념 및 필요성에 대해 간략히 살펴본 후 이를 구현하기 위한 방안에 대해 논의하고자 한다. 사용자가 네트워크상에서 원격의 인스트루먼트를 조작 및 운용할 수 있는 클라이언트-서버 기반의 Virtual Laboratory 구현을 위한 프레임워크 설계를 하고자 하며 단일 프로세서는 물론이고 네트워크상에서 대칭적 다중 프로세서 기반의 제어 및 계측 기능을 수행할 수 있는 시스템 모델을 제안하고자 한다.

Abstract Virtual Laboratory is the new concept in the field of Control & Measurement as network technology development, it is used in many university and institute nowadays. It is possible to control and measure the instrument in anywhere, anytime by the use of Virtual Laboratory and we can raise the efficiency of experiment.

In this paper, we review the concept and necessity of Virtual Laboratory at first and discuss about how we can implement Virtual Laboratory. Then we design framework of Client-Server based Virtual Laboratory for operating of remote instrument through network, and finally we suggest system model of Virtual Laboratory capable of controlling & monitoring of instrument.

Key Words : Virtual Laboratory, Client-Server Model, Virtual Instrument, Database, Instrument

1. 서 론

현재 기술 및 산업계에 있어서 최고의 화두는 IT(Information Technology)이며 최근 10년간 그 양이나 질에서 급속도로 발전하고 현대인의 생활 패턴을 근본적으로 바꾼 원동력 역시 정보기술에서 비롯된다는 점에서 그것이 우리 사회에 미친 파장 및 효과는 이루 헤아릴 수 없을 것이다. 네트워크 기술의 급격한 발전은 인터넷, 이동통신, 와이브로 등의 직접 연관된 분야 뿐만 아니라 그 외의 산업 분야에까지 파급되어 기술 발전을 주도하고 있는데 이공계 분야에 있어서 IT 기술을 접목시키지 않은 분야가 없을 정도이니 이로서 그 영향력 내지는 이를

응용해 얻을 수 있는 효율성을 미루어 짐작할 수 있다. 본 논문에서 다루고자 하는 Virtual Laboratory도 정보기술의 영역에서 크게 벗어나 있지 않으며 향후 네트워크 기술의 발전과 더불어 성장할 것이라는 점에서 그 활용 가능성은 더욱 밝다하겠다. Virtual Laboratory를 통하여 사용자는 언제 어느 곳에 있든지 실험실의 인스트루먼트에 접근하여 원하는 작업을 수행하는 것이 가능한데 본 논문에서는 이처럼 원격의 인스트루먼트에 대한 제어 및 모니터링을 가능하게 하는 Virtual Laboratory 구현을 위한 기본 프레임워크를 설계하고자 하며 단일 프로세서는 물론이고 네트워크 상의 대칭적 다중 프로세서에서도 활용가능한 Virtual Laboratory 구현 방안을 제시하고자 한다.

본 논문의 구성에 대해 간략히 기술해보면 다음과 같다. 본 서론에 이은 2장에서는 Virtual Laboratory의 개요에 대해 간략히 소개하고자 하며 그 기본원리라 할 수 있

¹한국천문연구원 전파천문연구원

*교신저자: 송민규(mksong@trao.re.kr)

는 가상 인스트루먼트의 원리에 대해 알아보기로 한다. 3장에서는 네트워크를 기반으로 Virtual Laboratory의 구성 및 그 동작 모델에 대해 알아볼 것이며 4장에는 이를 하드웨어 및 소프트웨어적으로 구현하기 위한 방안에 대해 세부적으로 알아보려고 한다. 5장에서는 이러한 논의를 통하여 구현된 Virtual Laboratory 모델을 제시하고자 하며 마지막으로 6장에서 그에 대한 결론을 맺도록 한다.

2. Virtual Laboratory의 개요 및 원리

Virtual Laboratory는 기존의 실험실 형태에 네트워크 기술을 적용하여 원격에서도 컴퓨터를 통하여 물리적 인스트루먼트에 대한 제어 및 모니터링을 수행할 수 있도록 구현한 것으로서 이를 통해 원격의 사용자는 시간과 장소에 관계없이 원하는 작업을 수행할 수 있다. Virtual Laboratory의 활용에 있어서 가장 큰 특징으로 작업 수행에 있어서의 신속성, 효율성을 생각할 수 있는데 이러한 장점으로 인하여 현재 여러 대학 및 연구기관에서 Virtual Laboratory를 구축하여 활용중에 있다. 본 절에서는 이러한 Virtual Laboratory의 개념을 이해하기에 앞서 먼저 그 기본 단위로 할 수 있는 가상 인스트루먼트에 대해 살펴보기로 하며 나아가 가상 인스트루먼트 상에서의 제어 및 모니터링을 기반으로 Virtual Laboratory의 동작 매커니즘을 서술하고자 한다.

2.1 Virtual Instrument 소개

Virtual Laboratory의 등장 이전에 사용자는 물리적으로 존재하는 장비를 수동으로 조작하여 실험을 수행할 수 밖에 없었는데 이로 인하여 인스트루먼트 활용에 있어서 많은 불편 및 단점이 존재하였다. 획일화된 인스트루먼트 인터페이스, 시스템 통합의 어려움은 물론이고 무엇보다 사용에 있어서의 시간과 공간의 제약은 기존 실험실 형태에서 가장 큰 단점이고 작업에 있어서 상당한 비효율을 초래하였다. 이러한 문제점을 극복하기 위하여 고안된 것이 Virtual Instrument 즉 가상 인스트루먼트로서 이는 물리적인 인스트루먼트를 컴퓨터상의 실행 가능한 프로그램으로 구현한 것으로 정의될 수 있으며 가상 인스트루먼트로 구성된 Virtual Laboratory를 통하여 사용자는 네트워크를 경유하여 컴퓨터상에서 인스트루먼트에 대한 제어 및 모니터링을 수행할 수 있다[5]. 뿐만 아니라 필요시 프로그램 수정을 통하여 실험 환경에 대한 설정 및 시스템 통합을 구현하는 것이 가능하다. 그림 1은 이러한 가상 인스트루먼트의 구성 및 그 동작 매커니

즘을 간략히 도식화한 것으로서 각 인스트루먼트는 소프트웨어로 구현되어 네트워크를 통하여 다른 애플리케이션이나 디바이스로 연결되는 것이 가능하다[1].

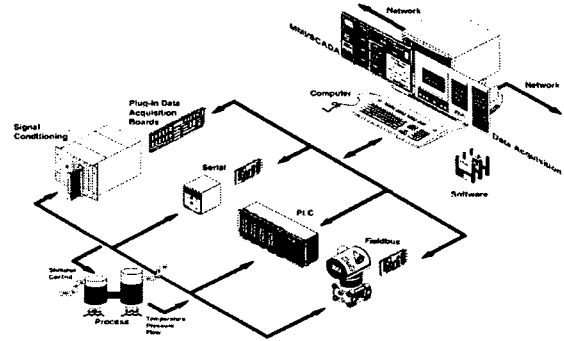


그림 1. 가상 인스트루먼트를 활용한 컴퓨터 기반의 제어 및 계측

컴퓨터 상의 가상 인스트루먼트는 물리적 인스트루먼트의 특성을 그대로 반영하며 이를 통하여 사용자는 해당 인스트루먼트에 대한 작업 수행은 물론 시스템 통합도 할 수 있다. 가상 인스트루먼트로 구동되는 Virtual Laboratory에서 소프트웨어는 이처럼 인스트루먼트에 대한 제어 및 모니터링 기능을 수행하며 반면 하드웨어는 소프트웨어에 대한 입출력 채널을 제공한다. Virtual Laboratory의 이러한 특성으로 인하여 시스템은 가상 인스트루먼트의 형태로 유연하게 구현될 수 있으며 사용자는 자신의 필요에 따라 프로그램의 변경 및 업그레이드를 통하여 최적화된 시스템을 구현하는 것이 가능하다.

2.2 새로운 Virtual Laboratory 모델의 필요성

컴퓨터 상의 프로그램을 통하여 인스트루먼트에 대한 제어 및 모니터링을 수행하는 Virtual Instrument 개념은 시간과 공간에 관계없이 원하는 작업을 수행할 수 있고 소프트웨어를 통한 시스템 통합이 가능하다는 특징으로 인하여 Virtual Laboratory의 형태로 구현되어 널리 활용되고 있다. 이러한 장점으로 인하여 NI(National Instruments), Agilent(구 HP(Hewlett-Packard))와 같은 인스트루먼트 생산 업체에서는 이미 Virtual Instrument를 구현할 수 있는 소프트웨어를 출시하여 그 활용을 적극적으로 지원하고 있는데 그 대표적인 예로 LabVIEW, VEE를 들 수 있다. 본 절에서는 이 중 관련 분야에서 가장 널리 활용되고 있는 NI사의 LabVIEW를 기반으로 기존 Virtual Laboratory의 단점 및 한계성을 지적하고 이를 극복할 수 있는 새로운 Virtual Laboratory의 필요성에 대

해 언급하고자 한다.

2.2.1 기존의 Virtual Laboratory

LabVIEW는 컴퓨터 기반의 제어 및 계측을 위하여 NI사에서 대략 20년전 출시된 소프트웨어로서 이를 통하여 사용자는 물리적 인스트루먼트를 컴퓨터상의 Virtual Instrument로 구성하여 원하는 작업을 수행하는 것이 가능하다[4]. 다른 텍스트 기반의 언어들(예>Visual C++, Visual Basic, Boland C, Delphi)과 비교하여 가장 큰 차이점으로 사용자 인터페이스 뿐만 아니라 소스 코드 또한 그래픽한 요소로 이루어져 있다는 것을 들 수 있는데 물리적 인스트루먼트는 LabVIEW상에서 아이콘의 형태로 존재하게 된다[7]. 그래픽 기반의 언어가 아니라는 점을 제외하면 Agilent의 VEE 또한 Virtual Instrument 구현에 있어서 LabVIEW와 비슷한 원리로 동작하는데 이러한 소프트웨어를 통하여 구현된 Virtual Laboratory의 구성 요소는 다음과 같이 세 가지 형태로 분류될 수 있다.

- 클라이언트 - 인스트루먼트에 대한 제어 및 모니터링을 수행하고자 하는 사용자 시스템
- 인스트루먼트 서버 - 인스트루먼트 제어 및 모니터링을 위한 가상 인스트루먼트가 실행되는 시스템으로서 인스트루먼트와 물리적으로 연결됨
- 인스트루먼트 - 사용자가 제어 및 모니터링 하고자 하는 대상인 물리적 인스트루먼트

사용자는 원격에서 인스트루먼트 서버상의 가상 인스트루먼트를 실행함으로써 물리적 인스트루먼트에 대한 제어 및 모니터링을 수행하는 것이 가능하다. 하지만 이러한 형태의 Virtual Laboratory의 모델은 그 확장성 및 효율성에 있어서 한계를 가질 수 밖에 없는데 그 이유를 두가지로 요약해보면 다음과 같다.

그 첫 번째로 제어 및 모니터링 구현에 있어 적용할 수 있는 인스트루먼트 갯수의 제한성을 들 수 있다. 각각의 인스트루먼트 서버는 GPIB, RS-232, USB 등 제한된 갯수의 인터페이스 버스뿐만 아니라 그 시스템 용량의 한계로 인하여 처리할 수 있는 용량에 한계가 있다. 이는 시스템 확장에 있어서 치명적 단점으로 작용하는데 여러대의 인스트루먼트를 연결하여 하나의 Virtual Laboratory에 대한 관리 기능을 수행하는 실험실 서버를 구축하면 이러한 단점을 극복하는 것이 가능하다. 두 번째로는 인스트루먼트 서버를 관리하는 또 다른 상위 계층 서버의 부재로 인하여 각 가상 인스트루먼트가 개별적으로 존재할 수 밖에 없기 때문에 Virtual Laboratory의 확장에 있

어서 한계가 있다는 것과 실험 장비, 데이터 등의 체계적 관리가 불가능하다는 점을 들 수 있다. 가상 인스트루먼트는 물리적 인스트루먼트를 컴퓨터 상의 소프트웨어로 구현한 것에 해당하기 때문에 Virtual Laboratory를 구성하는 여러 하드웨어 및 소프트웨어 자원에 대한 체계적 관리가 불가능하며 인스트루먼트 서버에 연결된 소수의 인스트루먼트에 대한 제어 및 모니터링만이 가능하다. 이와 같은 취약점으로 인하여 기존 Virtual Laboratory는 그 사용 및 확장에 있어 한계성을 갖을 수 밖에 없는데 이러한 단점을 극복하기 위해서는 Virtual Laboratory 설계에 있어서 이를 극복할 수 있는 기능이 구현되어야 할 것이다. 본 논문에서는 이를 위하여 인스트루먼트 서버의 상위 계층에 실험실 서버를 추가로 위치시키고자 하며 이를 통하여 보다 사용자는 다양한 인스트루먼트에 대한 효과적 제어 및 모니터링은 물론 체계적인 시스템 관리를 통하여 보다 확장된 Virtual Laboratory를 구축하는 것이 가능할 것이다.

2.2.2 Virtual Laboratory의 기본 구성

인스트루먼트를 통한 작업 수행에 있어 사용자가 고려해야 할 사항으로 신속·정확한 데이터 처리, 보다 넓은 영역에서의 활용성을 언급하지 않을 수 없는데 근래 시스템의 거대화 및 다양화에 따라 그 구현에 있어서 각 인스트루먼트간의 통합의 필요성은 더욱 커지고 있다. 이를 해결하기 위한 방법으로 본 논문에서는 클라이언트, 실험실 서버, 인스트루먼트 서버 그리고 인스트루먼트 4가지로 구성되는 Virtual Laboratory의 설계를 제안하고자 하며 각 구성 요소의 역할에 대해 기술해보면 다음과 같다.

• 클라이언트

Virtual Laboratory에 접속하기 위한 컴퓨터 시스템에 해당하며 원격의 인스트루먼트에 대한 제어 및 모니터링을 수행하는 가상적 작업 테이블이라 할 수 있다. TCP, UDP, DataSocket 등의 프로토콜을 통하여 원격의 서버와 통신하며 사용자 요청 및 명령을 서버로 전송하고 서버로부터 수신된 데이터를 사용자에게 출력한다. 일반적으로 사용자는 클라이언트의 웹 브라우저를 통하여 서버에 접속한다.

• 실험실 서버

하나의 Virtual Laboratory에 대한 전반적 운용 및 관리를 수행하는 시스템으로서 이를 수행하기 위한 웹서버, 데이터베이스, 관리용 애플리케이션으로 구성된다. 실험에 관련된 제반 사항을 담당하며 이를 위한 별도의 애플리케이션-자원 관리(Resource Manager) 애플리케이션-을

실행한다. 자원 관리 애플리케이션은 인스트루먼트 서버와의 통신을 통하여 실험실 내의 사용자 및 인스트루먼트에 관한 정보를 획득한다.

• 인스트루먼트 서버

인스트루먼트와 직접 연결되어 이에 대한 실제적 제어 및 모니터링을 수행하는 시스템으로서 가상 인스트루먼트가 실행되는 곳이기도 하다. Virtual Laboratory의 동작에 관련된 각 인스트루먼트의 정보를 포함하며 실험실 서버와의 통신을 통하여 인스트루먼트로부터 발생된 결과를 전송하고 클라이언트의 요청을 처리하는 역할을 한다.

• 인스트루먼트

인스트루먼트 서버와 연결되는 물리적 디바이스. 인스트루먼트 서버에서 실행되는 가상 인스트루먼트를 통하여 제어 및 모니터링되며 서버와 인스트루먼트 간의 인터페이스로 GPIB, RS-232, PXI 등이 주로 사용된다.

이러한 각 요소들이 네트워크 상에서 유기적으로 결합하여 원격의 물리적 인스트루먼트에 대한 제어 및 모니터링이 가능한 것이며 이를 기반으로 Virtual Laboratory의 간략한 구성을 도시해보면 다음과 같다.

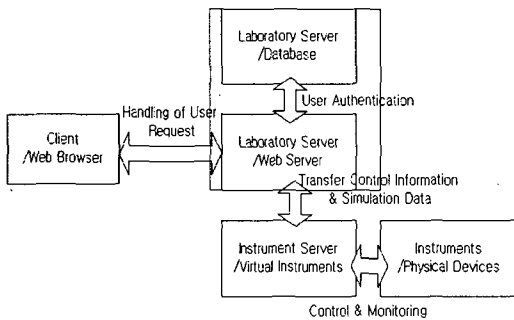


그림 2. Virtual Laboratory의 개략적 구성

그림에서 블록은 각각의 구성요소에 해당하며 왼쪽부터 차례로 클라이언트, 실험실 서버, 인스트루먼트 서버, 인스트루먼트를 나타내고 있다. 각 요소는 네트워크를 통하여 서로 연결되며 Virtual Laboratory 모델 구현에 있어 인스트루먼트를 제외한 각 요소는 TCP/IP를 통하여 연결된다.

3. Virtual Laboratory의 동작 메커니즘 및 모듈 구성

Virtual Laboratory를 구성하는 각 구성 요소들이 네트

워크 상에서 서로 통신하기 위해서는 각 요소들간 적절한 인터페이스가 구현되어야 하며 이를 소프트웨어적으로 구현하기 위한 모듈이 구현되어야 할 것이다. 원격의 클라이언트로부터 수신된 요청을 서버에서 처리하고 이를 기반으로 인스트루먼트에 대한 제어 및 모니터링을 수행함은 물론 인스트루먼트에서 획득된 데이터를 서버를 통하여 클라이언트로 전송하는 Virtual Laboratory의 기본 메커니즘에 대해 본 절에서는 보다 구체적으로 살펴보고자 하며 이를 구현하기 위한 모듈에 대해 알아보고자 한다.

3.1 Virtual Laboratory의 동작 메커니즘

Virtual Laboratory와 기존 형태의 실험실은 그 구성 및 활용에 있어서 상당한 차이가 있으며 이는 네트워크 기술의 구현 유무로 인한 것이라 할 수 있다. 기존 형태의 실험실에서 사용자가 직접 인스트루먼트 조작을 통하여 제어 및 계측을 수행하여야 했음에 반해 Virtual Laboratory에서는 네트워크를 경유하여 원하는 작업을 수행할 수 있다[6]. 이러한 사실에 근거하여 Virtual Laboratory의 동작 메커니즘을 네트워크 기반으로 간략히 도시해보면 다음과 같다.

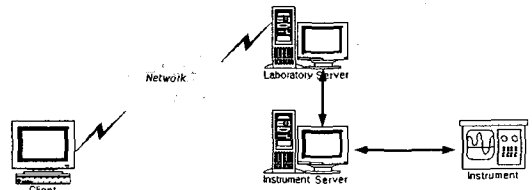


그림 3. Virtual Laboratory의 기본 메커니즘

서로 멀리 떨어진 클라이언트와 실험실의 서버, 인스트루먼트는 네트워크를 통하여 하나로 연결되며 요청, 처리, 응답 등의 메커니즘을 통하여 각 요소는 하나의 Virtual Laboratory로서 구동된다. 제어 및 모니터링 관련 모든 시스템 설정은 원격의 실험실에 위치하며 사용자는 네트워크 상의 서버에 접속하여 작업을 수행한다. 이로써 Virtual Laboratory를 구성하는 각 요소들 간의 통신을 위한 기본 모델로서 클라이언트-서버 모델이 적합하다 할 수 있으며 클라이언트와 서버는 네트워크를 통하여 서로 통신하며 각 인스트루먼트와 서버간의 통신은 RS-232, GPIB, PXI 등의 인터페이스를 통하여 구현됨을 알 수 있다[3].

3.2 Virtual Laboratory 구현을 위한 모듈

Virtual Laboratory를 통하여 제어 및 모니터링을 수행

함에 있어 각 구성 요소간 통신 구현을 위한 다수의 모듈이 필요하다. 사용자 요청의 처리, 서버와의 연결 설정, 인스트루먼트 관리 등의 절차는 Virtual Laboratory 실행에 있어서 반드시 구현되어야 하는 사항으로서 본 논문에서는 각각의 기능에 해당하는 개별적 모듈을 설계함으로써 Virtual Laboratory의 구체적 모델을 제시하고자 한다. 각 모듈은 클라이언트, 서버, 인스트루먼트 등 Virtual Laboratory를 구성하는 여러 요소를 기반으로 제어 및 모니터링을 구현하는 프로그램에 해당하며 Virtual Laboratory의 기능에 기반하여 고려해야 할 모듈을 나열해보면 다음과 같다.

- 클라이언트
 - 원격 연결 관리자
 - 시뮬레이션 엔진
- 서버
 - 요청 관리자
 - 인스트루먼트 관리자
 - 인증 관리자

이러한 모듈은 인스트루먼트, 사용자 관련 정보를 저장하는 데이터베이스와의 통신을 통하여 클라이언트와 서버의 연결 및 데이터 처리를 수행하기 때문에 Virtual Laboratory 구축에 있어서 반드시 해결되어야 할 부분이다[2]. 이에 대한 구현 방안은 다음 장에서 기술하기로 한다.

4. Virtual Laboratory 구현을 위한 시스템 설계

Virtual Laboratory는 클라이언트, 서버, 인스트루먼트 등 각 요소의 유기적 조합으로 구성되며 그 기능 수행에 있어 원격 연결 관리자, 요청 관리자 등 다양한 모듈의 지원을 받는다는 것을 알 수 있었다. 본 장에서는 이러한 구성 및 메커니즘을 기반으로 Virtual Laboratory의 계층적 모델을 제시하고 이를 실제적으로 구현하기 위한 시스템 설계 방안에 대해 기술하기로 한다.

4.1 사용자 인증을 위한 소프트웨어 모듈 설계

Virtual Laboratory 설계에 있어 유의할 사항으로 인증 절차를 들 수 있다. 즉 오직 하나의 클라이언트만이 해당 인스트루먼트에 대한 제어 및 계측을 수행할 수 있어야 하며 이 동안 다른 클라이언트에서의 시스템 접근은 블로킹 되어야 한다. 따라서 Virtual Laboratory 소프트웨어

의 설계에 있어 이러한 동작 메커니즘을 구현하기 위한 웹 서버 프로그램 및 관련 정보를 저장하는 데이터베이스가 존재되어야 할 것이다. 이러한 Virtual Laboratory의 특징을 기반으로 모듈 설계를 위한 다이어그램을 도시해보면 다음과 같다[4].

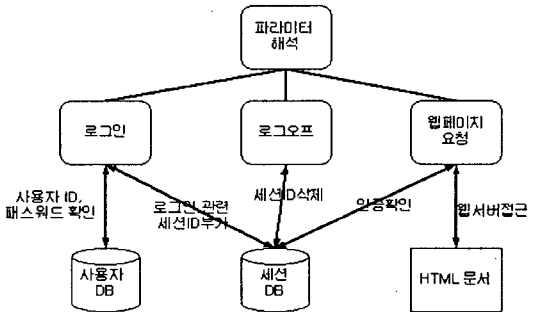


그림 4. 사용자 인증을 처리하는 Virtual Laboratory의 모듈

사용자 인증 처리와 관련된 시스템은 위 그림에서 보는 것처럼 세 개의 계층으로 구성되며 파라미터 해석 모듈(parameter interpretation module)을 정점으로 최하위 계층에 해당하는 사용자 DB, 세션 DB DB 등에 이르기 까지 여러 모듈의 유기적 조합으로 구성되는 것을 알 수 있다. 파라미터 해석 모듈은 하위 계층에서 발생한 정보의 수신 및 분석을 통하여 임의의 요청이 적절한 모듈에서 처리되도록 하는 역할을 한다. 예를 들어 클라이언트의 로그인 요청은 파라미터 해석 모듈에 의하여 로그인 모듈로 전송되어 처리되며 이 과정에서 수신된 ID와 패스워드는 사용자 DB를 통하여 검색될 것이다. ID와 패스워드가 적합한 것으로 판명되면 세션 DB에 의하여 해당 클라이언트에 유일한 세션 ID가 생성되어 사용자 로그인이 가능하게 되며 이러한 정보는 임의의 파일에 기록된다. 다른 모듈들도 마찬가지로 동작하며 각 모듈들의 이러한 구성 및 메커니즘을 기반으로 Virtual Laboratory에 수신된 클라이언트의 요청은 적절히 처리되어 그에 대한 적절한 응답이 반환된다.

4.2 클라이언트 및 서버 설계

이미 2.2.2절에서 언급한 것처럼 본 논문에서 제안하고자 하는 Virtual Laboratory는 클라이언트, 실험실 서버, 인스트루먼트 서버 그리고 인스트루먼트로 구성되며 각 구성 요소의 유기적 결합을 통하여 네트워크 기반의 인스트루먼트 제어 및 모니터링이 구현된다. 본 절에서는 이러한 기능을 구현함에 있어서 Virtual Laboratory 내에서 각 시스템이 담당하는 역할에 대해 알아보고 이를 구

현하기 위한 시스템 설계 방안에 대해 클라이언트, 실험실 서버, 인스트루먼트 서버, 클라이언트의 순으로 기술하고자 한다.

Virtual Laboratory에서 사용자에게 가상의 작업 환경을 제공하는 클라이언트는 제어 및 모니터링 구현을 위한 시뮬레이션 엔진과 사용자 요청을 처리하는 원격 접속 관리자로 구성된다. 시뮬레이션 엔진은 네트워크를 통하여 전송된 가상 인스트루먼트가 원격의 클라이언트에서 실행될 수 있도록 하는 가상 머신(Virtual Machine)에 해당하며 이를 통하여 가상 인스트루먼트 애플리케이션은 OS 독립적으로 동작하는 것이 가능하다. 이와 더불어 클라이언트를 구성하는 또 다른 요소인 원격 접속 관리자는 가상 인스트루먼트로부터 발생한 요청을 웹 브라우저와의 연계를 통하여 네트워크로 전송하는 역할을 하며 서버로부터 전송된 결과를 사용자에게 나타내주는 네트워크 애플리케이션으로서 웹 브라우저가 이에 해당한다. 클라이언트에서 가상 인스트루먼트는 ActiveX 기술을 통하여 웹 브라우저와 통합되어 실행되며 가상 인스트루먼트로부터 발생한 요청이 웹 브라우저를 통하여 네트워크로 전송될 수 있도록 두 애플리케이션 간의 적절한 통신이 이루어져야 할 것이며 원격 접속 관리자를 통하여 이러한 기능이 구현될 것이다.

실험실 서버는 클라이언트로부터의 네트워크 연결 요청을 처리하는 웹 서버와 Virtual Laboratory 관리 및 운용에 필요한 각종 정보를 저장하는 데이터베이스로 구성된다. 실험실 서버의 설계를 위하여 그 기능에 대해 좀더 세부적으로 기술해보면 다음과 같다. 실험실 서버는 클라이언트와 인스트루먼트 서버 사이에 위치하여 두 시스템 간의 통신을 위한 중계 역할을 한다. 클라이언트로부터 전달된 요청은 실험실 서버를 통하여 인스트루먼트 서버로 전달되고 인스트루먼트 서버에서 처리된 데이터는 실험실 서버를 통하여 클라이언트로 반환된다. 이러한 동작 메커니즘 구현을 위하여 Laboratory Server 내에 웹 서버가 구축되어야 하는데 아파치나 IIS(Internet Information Server) 등이 적절한 것으로 생각된다. 클라이언트에서 사용하는 웹 브라우저는 아파치나 IIS 등의 범용 웹 서버와 100% 호환되기 때문에 사용자는 이를 통하여 안정적으로 실험실 서버에 접근하는 것이 가능하다.

실험실 서버의 두 번째 특징으로 Virtual Laboratory를 구성하는 다양한 인스트루먼트 및 이에 대한 제어 및 모니터링을 수행하는 사용자들에 대한 체계적 관리를 들 수 있으며 각각 컴포넌트 DB, 사용자 DB를 통하여 이를 구현할 수 있다[4]. 이와 더불어 클라이언트와 인스트루먼트 서버 간의 통신 설정 관리를 위한 세션 DB가 추가적으로 구현되어야 하는데 이를 위하여 실험실 서버

내에 MySQL이나 Oracle 등의 데이터베이스 엔진이 구축되어야 한다.

인스트루먼트 서버는 물리적 인스트루먼트를 컴퓨터 상의 소프트웨어로 구현한 가상 인스트루먼트가 실행되는 시스템으로서 실험실 서버로부터 수신된 사용자 요청을 인스트루먼트로 전달하여 구동시키는 역할을 하며 인스트루먼트에 대한 실제적 제어 및 모니터링이 구현되는 부분이다. 가상 인스트루먼트를 구현하기 위한 대표적인 틀로 앞 절에서 언급한 LabVIEW나 VEE등의 소프트웨어가 많이 활용되고 있으며 이를 통하여 구현된 가상 인스트루먼트는 실험실 서버를 거쳐 수신된 사용자의 요청을 처리하고 물리적 인스트루먼트에 대한 제어 및 모니터링을 수행하는 역할을 한다.

마지막으로 인스트루먼트는 GPIB, RS-232, PXI 등의 인터페이스 버스를 통하여 인스트루먼트 서버와 물리적으로 연결되는 디바이스이다. 이를 소프트웨어적으로 제어 및 모니터링 하는데 필요한 관련 함수 및 디바이스 드라이버는 인스트루먼트 서버에서 가상 인스트루먼트의 형태로 구현되며 이 과정에서 VISA(Virtual Instrument Software Architecture)가 사용된다[4]. VISA는 인스트루먼트의 기능을 소프트웨어로 쉽게 구현하도록 해주는 드라이버로서 GPIB, VXI, RS-232 등 다양한 인터페이스로 구성되는 인스트루먼트 시스템을 지원하기 때문에 가상 인스트루먼트 구현에 있어서 상당한 효율성을 제공한다.

서버 설계에 있어서는 실험실 서버의 기능을 보다 충실히 반영하기 위하여 단일 서버의 형태보다는 실험실 서버와 인스트루먼트 서버가 조합되는 이중 서버를 구축하는 것이 바람직하다 할 수 있으며 그 역할을 간략히 요약해보면 다음과 같다.

- 실험실 서버 : Virtual Laboratory 운용에 필요한 각종 데이터베이스, 런-타임 엔진등의 소프트웨어가 저장되는 시스템
역할 : 작업 수행을 위한 클라이언트의 작업 환경 설정
- 인스트루먼트 서버: 인스트루먼트와 물리적으로 연결되는 서버.
역할 : 런-타임 시뮬레이션을 통하여 실제적으로 제어 및 모니터링이 구현되는 시스템

원격 접속 관리자 모듈을 통하여 사용자 요청은 네트워크를 통하여 실험실 서버로 전송된다. 클라이언트를 구성하는 두 요소(시뮬레이션 엔진, 원격 연결 관리자 모듈)는 클라이언트의 설정 단계에서 ActiveX 형태로 다운로드 설치되는데 사용자는 웹 브라우저를 통하여 가상 인

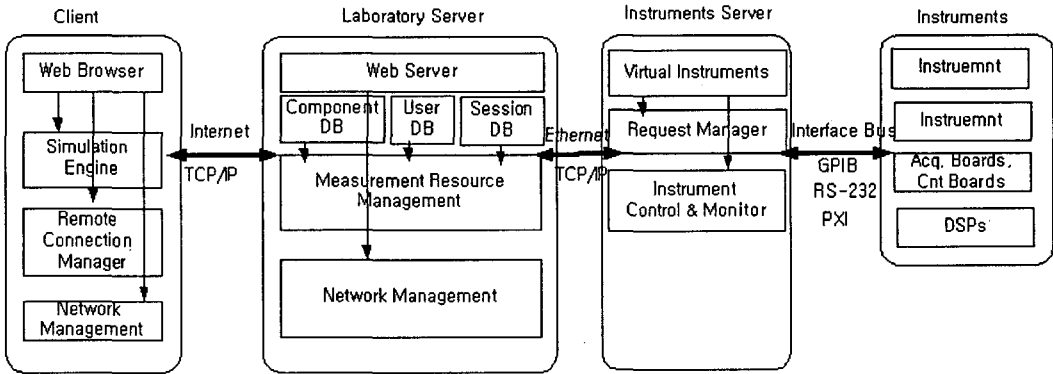


그림 5. 클라이언트 서버 기반의 Virtual Laboratory 모델

스트루먼트를 설정하며 그 내역은 실험실 서버상의 컴포넌트 데이터베이스에 파일 형태로 저장된다. 지금까지의 내용을 기반으로 보다 상세화된 클라이언트 서버 기반의 Virtual Laboratory 모델을 도시해보면 다음과 같다.

4.3 클라이언트-서버 기반의 Virtual Laboratory 구현

Virtual Laboratory를 구성하는 다양한 시스템은 클라이언트-서버 형태로 네트워크에 분산되어 동작하기 때문에 분산 계측 시스템(distributed measurement system)이라고 불리기도 하며 Virtual Laboratory의 각 요소들이 네트워크를 통하여 하나로 조합되어 제어 및 모니터링을 수행하는 것을 지금까지의 논의를 통해 알 수 있었다. 또한 Virtual Laboratory 구현을 위한 통신 방법으로 클라이언트-서버 모델이 가장 적합하다는 것을 알 수 있었으며 각 요소에서 이를 효율적으로 지원하기 위한 시스템 설계 방안에 대해 살펴보았는데 이를 좀더 세부적으로 기술하면 다음과 같다.

본 논문에서는 Virtual Laboratory 구현을 위한 기본 통신 방법으로 클라이언트-서버 모델을 사용하고자 하며 인스트루먼트, 데이터베이스에 대한 효율적 운용을 위한 이중 서버 모델을 제시하였다. 하나 이상의 인스트루먼트들은 인스트루먼트 서버를 거쳐 실험실 서버로 연결되어 네트워크 상의 클라이언트와 통신을 수행하며 서버측 시스템 설계에 있어서 인스트루먼트는 최하위 계층에 인스트루먼트 서버와 실험실 서버는 차례로 그 상위에 위치하게 된다. Virtual Laboratory에서 데이터의 발생 경로는 두 가지로 생각해볼 수 있는데 물리적으로 제어 및 모니터링이 수행되는 인스트루먼트와 데이터 수집 및 분석이 이루어지는 이중 서버가 이에 해당되며 물리적 인스트루먼트는 RS-232, GPIB나 VXI 등의 표준 인터페이스를 통

하여 네트워크 노드로서 동작하는 서버 시스템에 연결된다. Virtual Laboratory에 필요한 네트워크는 그 구성에 있어 내부와 외부 네트워크로 세분화될 수 있는데 인스트루먼트와 서버 시스템간은 이더넷, 클라이언트와 서버 시스템간은 인터넷을 기반으로 구축하는 것이 일반적인 형태라 할 수 있으며 서버에서 처리되는 데이터의 네트워크 전송에 있어 QoS 보장을 위한 프로토콜 등이 지원되어야 할 것이다. 이를 기반으로 Virtual Laboratory를 설계해보면 다음과 같다.

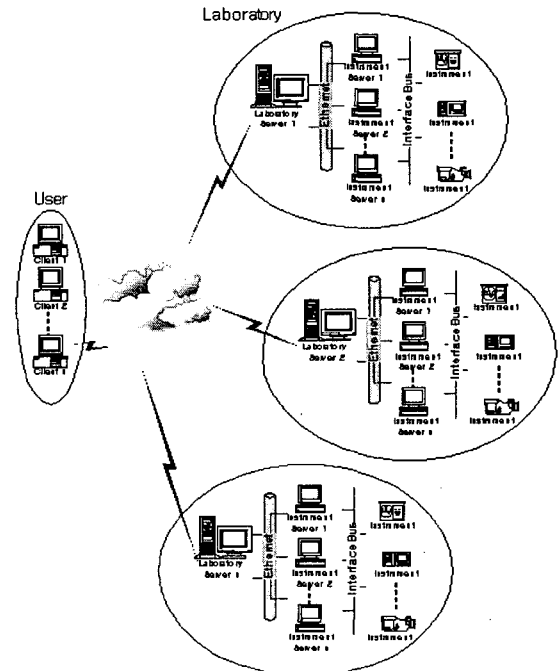


그림 6. 원격 제어 및 모니터링을 수행하는 Virtual Laboratory 설계

5. 결론

본 논문에서 우리는 Virtual Laboratory의 원리 및 구성 그리고 동작 메커니즘에 대해 살펴보았다. 또한 이를 구현하기 위한 하드웨어 및 소프트웨어 구성에 대해 알아보았으며 후반부에서는 이의 구현모형을 제안하였다.

로컬상의 인스트루먼트를 원격에서 제어하기 위하여 네트워크와 연관된 소프트웨어, 하드웨어 기술이 사용됨을 알 수 있는데 본 논문에서는 사용자가 원격에서 인스트루먼트에 대한 제어 및 모니터링을 수행하는 시스템 설계를 위하여 클라이언트, 실험실 서버, 인스트루먼트 서버, 인스트루먼트로 구성되는 클라이언트-서버 기반의 Virtual Laboratory 모델을 제시하였다. 뿐만 아니라 이중 서버 구조(실험실 서버 + 인스트루먼트 서버)를 선택하여 Virtual Laboratory를 구성하는 시스템과 리소스에 대한 확장성 및 체계적 관리를 보다 효과적으로 구현할 수 있도록 하였다.

Virtual Laboratory는 네트워크 및 하드웨어 소프트웨어 기술의 발달로 등장하였기 때문에 이러한 관련 기술이 발전함에 따라 그 비중 및 활용이 더욱 커질 것으로 전망된다. Virtual Laboratory의 활용으로 인하여 얻을 수 있는 이점은 상당하지만 사용자가 언제 어디서든지 웹 브라우저를 사용하여 원하는 작업을 수행할 수 있다는 것을 가장 큰 장점으로 꼽을 수 있다. 하지만 Virtual Laboratory가 실제 실험실에서 직접 장비를 다루고 실험 결과를 얻는 것을 완벽히 대체하지는 못하는 실정이고 바로 이것이 Virtual Laboratory의 마지막 과제라고도 할 수 있다. 이것이 극복될 때 Virtual Laboratory는 제어 및 계측 분야에서 향후 모든 사람들이 공감할 수 있는 획기적인 수단으로 자리 잡을 수 있을 것이다.

참고문헌

[1] B. Q. Cabell, et al., "Using Java to Develop Interactive Learning Material for the World Wide Web", International Journal of Engineering Education, Vol. 13, No. 6, pp. 397-406, July, 1999

[2] K. P. Shi, et al., "Virtual Instrument Technology Used in the BEPC Beam Diagnostic System", <http://www.aps.anl.gov/conferences/icalpeps/97/paper97/p038.pdf>

[3] S.J. Lu, et al., "The Application of Streaming Video in Web-Based 3D Virtual Laboratory", The 2002 International Conference on Imaging Science, Systems, and Technology, pp. 24-27, June, 2002.

[4] C. C. Ko, et al., "A Web-Based Virtual Laboratory on a Frequency Modulation Experiment", Application and Reviews, Vol.31, No.3, pp. 295-303, August, 2001

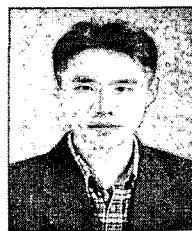
[5] J. Overstreet, et al., "Internet-Based Client/Server Virtual Instrument Designs for Real-Time Remote-Access Control Engineering Laboratory", ACC99-IEEE0280, pp. 1472-1476, June, 1999.

[6] C. C. Ko, et al., "Development of a Web-Based Laboratory for Control Experiments on a Coupled Tank Apparatus", IEEE Transactions on Education, Vol. 44, No. 1, pp. 76-86, February, 2001.

[7] John F. Richardson, "Real World Data Acquisition for Simulations Used in Web Based Virtual Reality" <http://www.scs.org/scsarchive/getDoc.cfm?id=1775>

송민규(Min-Gyu Song)

[정회원]



- 2001년 2월 : 강원대학교 전기공학과 (공학사)
- 2003년 2월 : 강원대학교 전자공학과 (공학석사)
- 2002년 12월 ~ 현재 : 한국천문연구원 전파천문연구부 연구원

<관심분야>

컴퓨터 네트워크, 리눅스 시스템 프로그래밍, BcN.