

분산계측 구현을 위한 시스템 아키텍처 구현 방안

송민규*

요약

분산계측은 시스템에 대한 제어 및 모니터링을 네트워크상에서 수행하는 기술이다. 분산계측시스템은 클라이언트, 서버, 인스트루먼트로 구성되며, 이들 객체는 네트워크상에 분산된 형태로 존재한다. 사용자는 웹 브라우저를 사용하여 서버 상의 임의의 프로그램을 실행할 수 있으며, 이를 통하여 인스트루먼트를 제어할 수 있다. 뿐만 아니라 시간과 공간의 제약을 넘어 작업 수행에 대한 결과를 일목요연하게 획득하는 것도 가능하다. 본 논문에서는 이러한 기능을 수행할 수 있는 분산계측시스템의 구현 모델을 설계하고자 한다. 분산계측시스템 설계에 있어 핵심 사항으로 클라이언트 서버 간 통신, 인스트루먼트 제어를 들 수 있는데, 본 논문에서는 파이썬 모듈에 기반하여 웹 서버와 GPIB 인터페이스로 이루어지는 분산계측 모델을 제안하고자 한다.

The Implementation Methodology of System Architecture for Distributed Measurement System

Min-Gyu Song*

Abstract

In this paper we present a system architecture for the distributed measurement system of instrumentation over the network. The proposed architecture allows remote control & monitoring of the instrument by using web browser.

Keywords : 분산계측 시스템, 원격 제어 및 모니터링, 가상 인스트루먼트, 파이썬 모듈, COMEDI, GPIB 통신

1. 서론

분산계측시스템은 기존의 계측 시스템에 네트워크 기술을 접목시킨 획기적인 개념으로 가상 실험실과 유사한 개념을 갖고 있다[1]. 이를 통해 네트워크상의 원격 사용자가 인스트루먼트를 다룰 수 있고 해당 인스트루먼트에 대한 제어 및 모니터링은 물론 그 결과에 대한 분석까지 수행하는 것이 가능하다. 기존의 방식에 비하여 활용 및 효율성 측면에 있어서 월등하기 때문에 현재 여러 대학 및 연구기관에서 이 분산계측시스템을 구축하여 운행중에 있다.

분산계측시스템은 원격에서 웹을 통하여 접속 가능한 실제적인 실험실의 형태로서의 기능을

하기 때문에 이는 원격 교육에 적합하다 할 수 있다. 이로 인하여 전 세계의 많은 대학 및 연구소에서는 관련 시스템 모델을 개발하여 운용 중에 있으며 이는 앞으로 더욱 활성화 될 것으로 전망되고 있다. e-Learning의 한 형태에 해당하는 분산계측시스템의 활용이 이처럼 증가하게 된 것은 네트워크 기술을 접목하여 오프라인에서만 작업수행이 가능하던 기존의 방법보다 그 성능 및 효율성 측면에 있어서 월등하기 때문이라 할 수 있다. 실제로 사용자는 웹브라우저만 있으면 언제 어디서든지 자신이 원하는 작업을 수행하는 것이 가능하다. 분산계측시스템을 통하여 사용자는 원격에서 인스트루먼트 파라미터 설정은 물론 데이터 분석도 수행할 수 있다[2]. 무엇보다 시간과 공간의 제약을 받지 않기 때문에 인스트루먼트 이용의 효율성을 배가시킬 수 있다는 것은 가장 큰 장점중의 하나이다.

분산계측시스템은 네트워크상의 다수의 클라이언트와 서버 그리고 GPIB, RS232 등의 시스

※ 제일저자(First Author) : 송민규
접수일자:2007년07월09일, 심사완료:2007년07월20일
* 한국천문연구원
mksong@kasi.re.kr

템 인터페이스로 연결되는 인스트루먼트로 구성되는데[3], 원격에서 컴퓨터 네트워크를 통하여 실제 디바이스를 조작하고 그 결과를 모니터링하기 위해서는 이와 같은 구성요소가 적절히 통합되어야 할 것이다.

이를 위하여 우리는 분산계측시스템의 구성요소인 클라이언트, 서버, 인스트루먼트의 기능 및 역할에 대해 살펴본 후, 이를 효율적으로 통합시킬 수 있는 방안에 대해 논의하기로 한다. 본 논문에서 제안하고자 하는 시스템 아키텍처는 클라이언트와 서버 간 통신, 서버 상의 인스트루먼트 제어 두 가지로 분류할 수 있으며 핵심 기능은 서버를 통하여 구현된다. 객체지향언어인 파이썬을 기반으로 이를 실제적으로 구현할 수 있는 방안에 대해 살펴볼 것이고 CGI, 웹서버의 GPIB등의 인터페이스의 설계를 통하여 분산계측시스템 모델의 구현방안을 기술하도록 하겠다. 전체적인 논문 진행은 다음과 같다. 먼저 2장에서는 분산계측시스템의 개요 및 원리에 대해 대해서 간략히 살펴볼 것이고 3장에서는 원격 제어 및 모니터링 시스템의 구성에 대해 논의하고자 한다. 4장에서는 분산계측시스템 구성을 위한 인터페이스에 대해 기술하고자 하며 5장에서 분산계측시스템 모델을 설계하고자 한다. 그리고 6장에서 본 논문의 결론을 맺고자 한다.

2. 분산계측시스템의 개요 및 원리

분산계측시스템을 사용하면 기존에는 작업수행에 있어서 불가능하던 여러 문제점을 극복할 수 있다. 이로 인하여 현재 여러 대학 및 연구기관에서 분산계측시스템을 사용 중에 있는데 본 절에서는 분산계측시스템의 필요성 및 구성요소에 대해 간략히 살펴보고자 한다.

2.1 분산계측시스템의 필요성

분산계측시스템의 필요성이 대두되고 이의 활용이 늘어난 것은 네트워크 및 관련 하드웨어/소프트웨어 기술의 발전과도 연관이 있지만 무엇보다 오프라인상의 실험실에서는 얻을 수 없었던 효과를 얻을 수 있다는 것이 가장 중요한 원인으로 작용하였다[4].

분산계측시스템이 등장하기 전에 사용자는 작업 수행을 위한 시간, 공간에 있어서 제약을 받을 수밖에 없었다. 이로 인하여 자유로운 작업수행은 근본적으로 불가능하였고 인스트루먼트 이용에 있어서도 제한된 서비스를 받을 수밖에 없었다. 이러한 문제점을 극복하고자 분산계측시스템의 필요성이 제기되었는데, 이를 통하여 네트워크와 컴퓨터 시스템으로 연결된 인스트루먼트를 기반으로 사용자들이 언제 어디서든지 작업을 수행하는 것이 가능하다. 이러한 이점으로 인하여 분산계측시스템의 활용은 앞으로 더욱 증가될 것이고 e-Learning의 새로운 패러다임을 제시할 수 있을 것으로 전망되고 있다.

2.2 컴퓨터 기반의 제어 및 계측

기존에 사용자는 원하는 실험 수행을 위하여 물리적으로 존재하는 장비를 이용하였다. 제어 및 계측 작업은 물론 데이터 획득을 수동으로 해야 하는 이러한 형태의 인스트루먼트 환경에서 시스템 통합은 극히 제한되었고 작업 효율은 저하될 수밖에 없었다. 이러한 단점 및 한계를 극복하기 위하여 고안된 것이 컴퓨터 기반의 제어 및 계측 시스템으로서 이를 통하여 원격에서 인스트루먼트에 대한 제어 및 모니터링을 효율적으로 수행할 수 있다[5].

컴퓨터 기반의 제어 및 계측 시스템은 개방형 시스템으로서 역할을 하므로 네트워크를 통하여 다른 애플리케이션이나 디바이스로 연결될 수 있는 특징을 지니고 있다. 이러한 형태의 시스템에 있어서 소프트웨어는 핵심기능을 수행하는 반면 하드웨어는 소프트웨어에 대한 입출력 채널을 제공한다. 때문에 시스템은 가상 인스트루먼트 애플리케이션에 의해 유연하게 구현될 수 있다. 즉, 구현된 애플리케이션 자체가 하나의 인스트루먼트에 해당하며 이를 통하여 제어 및 계측을 수행할 수 있다.

3. 원격 제어 및 모니터링 시스템의 기본 구성

인스트루먼트를 원격에서 제어하기 위해서는 먼저 두 가지 전제조건이 충족되어야 한다. 그 첫 번째는, 인스트루먼트를 컴퓨터상에서 제어할

수 있도록 소프트웨어적으로 구현해야한다는 것이다. 두 번째 요건은, 컴퓨터와 해당 인스트루먼트 간 명령어를 송수신 할 수 있는 인터페이스를 구성하는 것이다. 본 절에서는 이 두 가지 절차에 대해 알아보기로 한다.

3.1 Virtual Instrumentation 개요

Virtual Instrument는 의미 그대로 컴퓨터상에 존재하는 가상 장비라고 정의내릴 수 있는데, 적은 비용으로 더 많은 성능을 구현하도록 소프트웨어가 중심이 되어 컴퓨터와 데이터 입출력 하드웨어를 밀접하게 통합시키는 기술을 의미한다 [5]. 이를 통하여 기존에 널리 사용되고 있는 계측 장비들을 컴퓨터상에서 효율적으로 구현할 수 있으며 사용자에게 최적화된 시스템 통합을 이끌어내는 것이 가능하다. 본 논문에서 활용할 원격 제어 및 모니터링 시스템(Remote Control & Monitoring) 시스템은 원격에서 사용자가 인터넷을 통하여 인스트루먼트를 제어하는 것이기 때문에 해당 인스트루먼트를 컴퓨터상에 소프트웨어적으로 구현하는 과정이 반드시 필요하다.

3.2 가상 인스트루먼트의 기능 및 역할

기존의 제어 및 계측 장비들은 하나의 장비에 모든 기능들이 들어있으며 사용자가 전원을 넣고 버튼을 누르면 실행을 할 수 있었다. 그러나 이러한 장비는 하나의 독립된 장비에 모든 기능이 다 들어있기 때문에 편리하지는 하지만 다른 기능이 필요하다면 다시 다른 고가의 장비를 새로 구입해야 되었다. 반대로 그 장비의 많은 기능 중에서 대부분은 사용자가 한 번도 사용하지 않는 필요 없는 경우가 많았다. 또한 여러 독립된 장비들을 하나의 시스템에서 통합하기가 어려웠다. 물론 GPIB나 시리얼 케이블을 이용하여 어느 정도까지는 통합이 가능하지만 그것에는 한계가 있다. 또한 장비 구입 이후 업그레이드가 아주 힘들거나 불가능하였다[5].

이러한 단점 및 한계성을 극복하기 위하여 나온 개념이 컴퓨터를 기반으로 한 자동제어 및 계측 시스템이다. Virtual Instrumentation은 컴퓨터를 메인 컨트롤러로 두고, 사용자가 필요로 하는 요소들을 연결하여 컴퓨터 속의 가상공간에서 사용자가 정의한 인스트루먼트를 만드는 것으로 정의내릴 수 있다. 가상 인스트루먼트의

이점을 간략히 정리하면 다음과 같이 나타낼 수 있다.

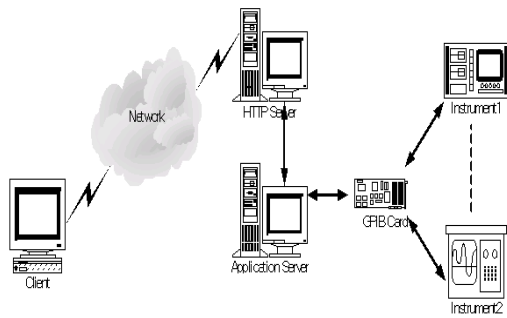
- 사용자가 원하는 기능의 시스템 추가 용이
- 컴퓨터의 성능향상에 비례하여 인스트루먼트의 성능 향상
- 사용자에게 친숙한 작업환경으로 인한 작업 효율 극대화
- 인터넷 기반의 인스트루먼트 구현

4. 분산계측시스템 구성을 위한 인터페이스

분산계측시스템은 그 기능면에 있어서 클라이언트, 웹서버, 제어 애플리케이션, 인스트루먼트로 분류할 수 있다. 이를 구현하기 위한 기술로서 네트워크와 시스템 인터페이스 기술을 들 수 있는데 HTTP, 웹 서버 기술이 전자에, 리눅스 디바이스 드라이버가 후자에 해당한다. 분산계측시스템이 기존의 Laboratory를 대체할 수 있다면 이를 지원할 수 있는 유연한 시스템 구성이 선행되어야 하며 위의 구현 기술들이 유기적으로 조화되어야 한다. 본 절에서는 이러한 시스템 구성에 하드웨어와 소프트웨어로 분류하여 접근하고자 하며 이를 기반으로 전체 시스템의 동작 메커니즘을 살펴보고자 한다.

4.1 하드웨어 인터페이스 구성

분산계측시스템을 통하여 외부의 사용자가 로컬상의 인스트루먼트를 제어하기 위해서는 클라이언트, 서버, 인스트루먼트의 3가지 요소가 적절히 조화되어야 한다[2]. 각 구성요소는 네트워크로 결합되어 전체적인 분산계측시스템으로서의 기능을 수행하게 되는데 이에 대한 개략적인 구성도를 도시하면 다음과 같이 나타낼 수 있다.



(그림 1) 분산계측시스템 구현을 위한 시스템 구성

위 구성도로부터 서버는 네트워크를 통하여 클라이언트와 연결되며 로컬상에서는 GPIB 인터페이스를 통하여 인스트루먼트와 연결되는 것을 알 수 있다. 클라이언트, 인스트루먼트 두 요소와 인터페이스를 갖는 것으로부터 분산계측시스템에서 서버의 역할을 규정할 수 있는데 이는 기술하면 다음과 같다.

먼저 서버는 네트워크상의 다수의 클라이언트로부터 수신되는 요청을 처리할 수 있어야 하기 때문에 웹 서버 기능이 구현되어야 할 것이다 [3]. 클라이언트는 인스트루먼트 제어 및 모니터링을 수행하기 위하여 이에 관련된 정보를 서버로 전송하는데 이는 이후 가상 인스트루먼트의 입력값으로 사용된다. 서버에 연결되는 두 번째 인터페이스로 인스트루먼트를 들 수 있다. 물리적 인스트루먼트를 컴퓨터상에서 제어하기 위하여 가상 인스트루먼트가 애플리케이션 서버에서 실행되어야 하며, 본 논문에서는 파이썬을 활용하여 이를 구현할 것이다. 이를 통하여 원격의 사용자는 웹 브라우저를 통하여 로컬 상에 위치한 인스트루먼트를 제어하는 것이 가능하다.

4.2 소프트웨어의 동작 메커니즘

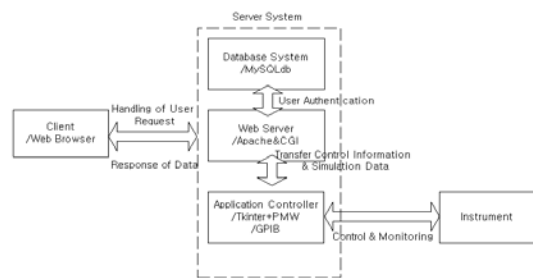
본 논문에서는 분산계측시스템 구현을 위한 프로그래밍 언어로서 파이썬을 사용하고자 하며 분산계측의 구현을 위하여 다음의 모듈을 활용하고자 한다.

- CGIHTTPServer모듈: 웹 브라우저와 웹 서버 간 데이터의 요청 및 응답을 처리하는 미들웨어
- BaseHTTPServer모듈: Socket서버위에

만들어진 프레임워크로서 HTTP서버의 기본 프레임워크를 제공

- GPIB 모듈: 애플리케이션 서버에서 인스트루먼트와의 통신 수행을 위한 파이썬 모듈
- Tkinter 모듈: GUI 구현을 위한 Tcl/Tk 인터페이스 모듈
- PMW(Python MegaWidget) 모듈: Tkinter와 관련된 대표적인 써드파티 모듈로서 고급 복합 위젯을 제공
- MySQLdb 모듈: CGI 프로그래밍에서 데이터베이스 작업을 위한 DB 모듈

원격의 클라이언트로부터 수신되는 제어 요청을 수신하고 그에 대해 HTML 페이지로 응답하기 위하여 로컬 상에 웹 서버가 필요하다. 본 논문에서는 이를 위한 웹 서버의 형태로 리눅스 상에서 아파치 웹 서버를 사용하였으며, 클라이언트에서 파이썬으로 구현된 가상 인스트루먼트 프로그램을 실행하는 원격 프로시저로서 CGI 모듈을 활용하였다. 또한 GPIB 방식으로 인스트루먼트를 제어하기 위하여 GPIB 모듈이 필요하며 이러한 제어를 GUI 형태로 구현할 수 있도록 Tkinter과 PMW 모듈을 활용하였다. 이를 통하여 인스트루먼트 제어 및 모니터링 역할을 하는 가상 인스트루먼트 프로그램은 애플리케이션 서버에서 실행되며 웹 서버와 연동함으로써, 원격의 클라이언트가 웹 브라우저를 통하여 인스트루먼트에 대한 제어 및 모니터링을 수행하는 것이 가능한 것이다. 이러한 소프트웨어의 구성 및 동작 메커니즘을 그림으로 나타내보면 아래와 같다.



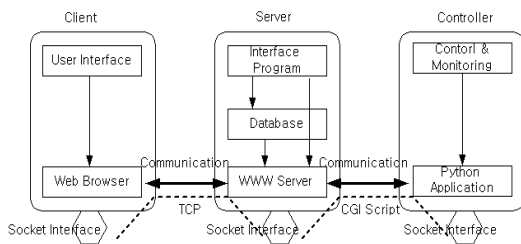
(그림 2) 분산계측시스템의 소프트웨어 구성 및 동작 메커니즘

5. 분산계측시스템 설계

지금까지 분산계측시스템의 하드웨어 및 소프트웨어 구성, 그리고 동작 메커니즘에 대해 살펴 보았다. 이를 바탕으로 본 절에서는 분산계측시스템 실제적 구현 모델을 제안하고자 한다.

5.1 원격 제어 및 모니터링을 위한 DCSS 모델

분산계측시스템의 유연한 동작을 지원하고 시스템의 확장성을 고려하여 본 논문에서는 클라이언트-웹서버-컨트롤러로 구성되는 클라이언트-서버 모델을 제안하고자 한다. 이 모델은 그 기능면에서 두 개의 클라이언트-서버가 결합된 것과 동일한 기능을 수행하는데 클라이언트-웹서버, 웹서버-컨트롤러가 결합하여 분산계측시스템으로서 완벽한 기능을 수행한다. 이 두 쌍을 완벽하게 결합시켜 분산계측시스템으로서 구동시키기 위하여 웹서버에서는 TCP 소켓 인터페이스를 구현하여야 한다. 이러한 시스템 구성을 통하여 분산계측시스템은 보다 유연한 기능 및 확장 가능한 성능을 얻을 수 있을 것으로 예상된다. 이렇게 제안된 분산계측시스템의 통신 모델을 (그림 3)에서 보여주고 있는데 소켓 인터페이스에 기반한 클라이언트와 서버 사이의 메커니즘을 확인할 수 있다. 네트워크를 경유하여 클라이언트, 웹서버, 컨트롤러는 전체 분산계측시스템을 구성하게 되며 이를 통하여 클라이언트로부터는 컨트롤러에게 인스트루먼트 정보를 전송하며 컨트롤러는 클라이언트에게 데이터를 전송하는 것이 가능하게 된다.



(그림 3). 각 구성 요소의 통신 모델

클라이언트, 서버, 컨트롤러는 분산계측시스템에서 각각 사용자 인터페이스, 웹서버, 가상 인스트루먼트에 해당한다. 컨트롤러에서 인스트루

먼트에 대한 제어 및 모니터링 수행을 위하여 파이썬 GUI 프로그램을 작성하였으며, 이 과정에서 Tkinter 모듈을 활용하였다. 이렇게 구현된 가상 인스트루먼트를 CGI 스크립트 형태로 웹서버에 포팅하였고, 원격에 위치한 클라이언트에는 웹 브라우저를 통하여 로컬 상의 인스트루먼트를 제어하는 것이 가능한 것이다.

이러한 차별화된 기능을 수행하는 각 구성요소가 유기적으로 결합되어 분산계측시스템이 구현되며 이를 통하여 원격의 사용자는 언제 어디서든 웹 브라우저를 통하여 자신이 원하는 작업을 수행하는 것이 가능하게 된다.

5.2 제어 시스템 구현

분산계측시스템은 클라이언트와 서버 그리고 인스트루먼트로 구성되기에 이들 각 요소가 유기적으로 어떻게 조합되는지가 시스템의 성능을 결정짓는 바로미터에 해당한다. 분산계측시스템의 구현은 그 기능에 따라 크게 세 가지로 블록화할 수 있는데 이를 요약해보면 같다.

- 클라이언트와 웹서버 간의 소켓 통신
- 웹서버와 가상 인스트루먼트 간 데이터 처리
- 가상 인스트루먼트와 물리적 인스트루먼트 간 제어 소프트웨어 구축

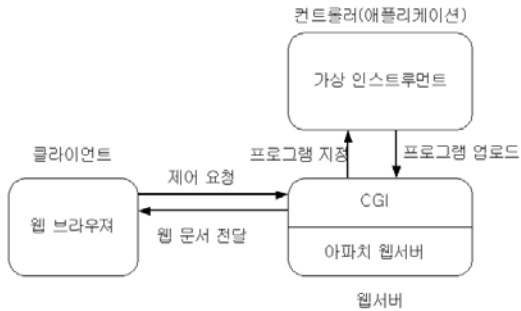
이러한 각 요소의 메커니즘 및 구현 방법에 대해서는 하단의 각 절에서 구체적으로 설명하고자 한다.

5.2.1 클라이언트와 서버 시스템 간의 웹 서비스

원격에 위치한 클라이언트는 웹 브라우저를 통하여 로컬 상의 가상 인스트루먼트에 접속할 수 있어야 한다. 따라서 로컬 상의 리눅스 시스템에 웹 서버는 반드시 구축되어야 하며 클라이언트는 웹 서버를 통하여 가상 인스트루먼트를 실행하는 것이 가능하다.

설치할 웹서버로 본 논문에서는 아파치 웹서버를 선택하였으며, 가상 인스트루먼트의 원격 실행을 지원하기 위하여 CGI를 설치하였다. 이를 통하여 클라이언트는 서버의 주소와 실행하고자 파이썬 프로그램을 지정하여 프로그램을 실행할 수 있으며 이의 대략적인 메커니즘을 그

림으로 나타내면 아래와 같다.



(그림 4) 클라이언트와 서버 간의 통신 모델

이러한 동작 메커니즘을 기반으로 클라이언트는 웹 브라우저를 통하여 로컬 상의 가상 인스트루먼트에 접속할 수 있다. 웹 브라우저에서는 접속하고자 하는 서버 어드레스와 실행할 프로그램 이름을 지정하면 되며, 이를 통하여 원격에서 로컬 상의 인스트루먼트를 실제적으로 제어하는 것이 가능하다.

5.2.2 인스트루먼트 제어를 위한 소프트웨어 구현

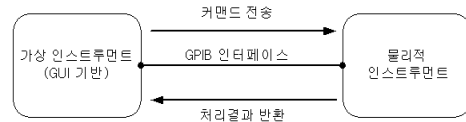
오실로스코프, 스펙트럼 분석기, 주파수 합성기 등의 인스트루먼트는 컴퓨터와 통신하기 위한 다양한 인터페이스를 지원한다. 이러한 인터페이스의 예로 시리얼 통신, GPIB를 비롯하여 이더넷 통신을 들 수 있다. 본 논문에서는 GPIB를 기본 인터페이스로 하여 인스트루먼트에 대한 제어 및 모니터링을 구현하고자 하는데 이는 제어 및 계측 분야에서 GPIB가 가장 범용적인 인터페이스이고 다수의 인스트루먼트에 GPIB가 기본으로 활용되기 때문이다. 파이썬에서 지원하는 GPIB 모듈을 임포트하여 커맨드 전송 및 처리를 위한 메소드를 호출하는 것이 가능하다. 부가적으로 DAQ(Data Acquisition)카드의 경우에는 인스트루먼트 제어에 필요한 각종 드라이버가 COMEDI 모듈로 지원되기에 파이썬에서 이를 효율적으로 활용할 수 있다.

PC 상에서 인스트루먼트 제어의 친숙성 및 사용자 편의를 위하여 제어 인터페이스는 GUI 환경으로 구축되어야 한다. 이를 위하여 파이썬의 GUI 모듈인 Tkinter와 확장모듈PMW 모

듈을 활용하였다. Tkinter 모듈의 경우 bind함수를 통한 이벤트 처리를 적극 지원하고 있다. 따라서 GUI 환경에서 메뉴 선택시 그에 관련된 메소드를 아래의 방법으로 호출하는 것이 가능하다.

```
widget.bind(event, handler)
```

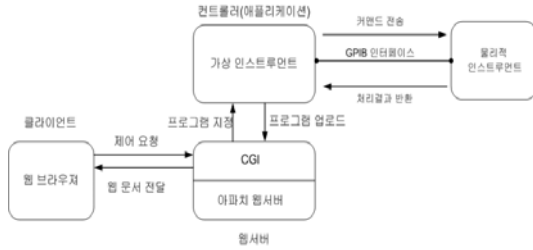
handler 부분에 인스트루먼트 제어 및 모니터링 관련 메소드를 지정하면 GUI에서 이벤트가 발생하였을 때 원하는 기능을 수행토록 할 수 있다. 물리적 인스트루먼트를 제어하기 위한 이러한 소프트웨어의 구성 메커니즘을 블록화하여 그림으로 나타내보면 아래와 같다.



(그림 5) GPIB 기반의 제어 메커니즘

5.2.3 제안된 시스템 모델

분산계측시스템은 클라이언트와 서버, 그리고 물리적 인스트루먼트로 구분되며, 각 요소를 조합시키기 위한 모델에 대해 알아보았다. 원격의 클라이언트는 웹서버를 통하여 가상 인스트루먼트에 접속하여 물리적 인스트루먼트에 대한 제어 및 모니터링을 수행한다. 이는 서버 상에서 CGIHTTPServer, BaseHTTPServer 모듈을 통하여 구현하였다. 물리적 인스트루먼트를 제어하기 위한 가상 인스트루먼트의 GUI 환경은 Tkinter, PMW 모듈에 기반하여 작성하였으며, GPIB 인터페이스를 통하여 인스트루먼트와의 통신을 수행하였다. 분산계측시스템은 이처럼 클라이언트, 서버 시스템 그리고 인스트루먼트 3가지 요소가 네트워크상에서 구성되며 제어 및 통신 관련 모듈을 활용하여 파이썬으로 구현된다. 상기 언급한 요소와 모듈로 이루어지는 분산계측시스템의 구현 모델을 도시해보면 아래와 같다.



(그림 6) 분산계측시스템의 구현 모델

6. 결 론

본 논문에서 우리는 분산계측시스템의 원리 및 구성 그리고 동작 메커니즘에 대해 살펴보았다. 또한 이를 구현하기 위한 하드웨어 및 소프트웨어 구성에 대해 알아보았으며 후반부에서는 이의 구현 모델을 제안하였다.

로컬상의 인스트루먼트를 원격에서 제어하기 위하여 네트워크와 연관된 소프트웨어, 하드웨어 기술이 사용됨을 알 수 있었다. 본 논문에서는 분산계측시스템의 구현모델로서 클라이언트, 웹서버, 컨트롤러, 인스트루먼트로 구성되는 DCSS(Double Client Server Structure)를 제안하였으며 이를 구성하기 위하여 통신 및 제어 모듈을 활용하였다. 제안된 분산계측시스템 모델을 통하여 각 구성요소가 이에 부합되는 개별적인 기능을 갖는다는 것을 알게 되었고 이것이 유기적으로 결합하여 분산계측시스템을 구성한다는 것 또한 확인할 수 있었다. 분산계측시스템에서 네트워크 사용자는 컴퓨터에서 실행되는 프로그램 통하여 인스트루먼트를 제어하게 된다. 즉, 사용자는 물리적 인스트루먼트의 복사본에 해당하는 애플리케이션의 조작을 통하여 해당 인스트루먼트에 대한 제어 및 모니터링을 수행하며 이러한 애플리케이션에 해당하는 가상 인스트루먼트는 분산계측시스템을 가동시키는 엔진에 해당한다. 이처럼 제어 및 모니터링을 수행하는 각각의 유기체들이 네트워크상에 분산된 형태에서, 각 유기체간의 통신 및 제어는 더더욱 중요하다. 본 논문에서는 이를 구현하기 위한 구현 모델을 제안하였으며, 클라이언트와 서버간의 소켓 통신, 가상 인스트루먼트의 실행에 기반한 웹페이지 전송, GPIB 통신에 기반한 인스트루먼트 제어로 이를 세분화하였다.

분산계측시스템은 네트워크 및 하드웨어 소프트웨어 기술의 발달로 등장하였기 때문에 이러한 관련 기술이 발전함에 따라 그 비중 및 활용이 더욱 커질 것으로 전망된다. 분산계측시스템의 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 분산계측시스템이 실제 실험실에서 직접 장비를 다루고 실험결과를 얻는 것을 완벽히 대체하지는 못하는 실정이고 바로 이것이 분산계측시스템의 마지막 과제라고도 할 수 있다. 이것이 극복될 때 분산계측시스템은 모든 사람들이 공감할 수 있는 획기적인 수단으로 자리 잡을 수 있을 것이다.

참 고 문 헌

- [1] Cristaldi L. Ferrero A. and Muscas, C, "The impact of Internet transmission on the uncertainty in the electric power quality estimation by means of a distributed measurement system," Instrumentation and Measurement, IEEE Transactions, Vol. 52, No. 4, 2003.
- [2] Horak Goran. Vasic Darko. and Bilas Vedran, "A Framework for Low Data Rate, Highly Distributed Measurement Systems," Instrumentation and Measurement Technology Conference Proceedings IEEE, pp. 1-4, 2007.
- [3] Burch J. Eidson, J. and Hamilton B, "The design of distributed measurement systems based on IEEE1451 standards and distributed time services," Vol. 2, pp. 529-534, 2000.
- [4] Cicirelli F. Furfaro A. Grimaldi D. Nigro L. Pupo F, "Management architecture for distributed measurement services [power demand monitoring and control system example]," Instrumentation and Measurement Technology Conference, Vol. 2, pp. 974-979, 2004.
- [5] 광두영, LabVIEW Express : 컴퓨터 기반의 제어와 계측, Ohm사(오르사), 서울, 2003

송 민 규



2001년 : 강원대학교 전기학과 (공학학사)

2003년 : 강원대학교 전자공학과 (공학석사)

2002년~현재 : 한국천문연구원 전파천문연구부 연구원

관심분야 : 분산 컴퓨팅, 썬 클라이언트, 리눅스 프로그래밍, 초고속 네트워크, e-VLBI