

원격 감시제어를 위한 웹 서비스 S/W 플랫폼 설계 및 구현

Design and Implementation of Web Service S/W Platform for Remote Monitoring and Control

이태희, 김주만

부산대학교 바이오정보전자공학과

Tae-Hee Lee(thlee@pusan.ac.kr), Joo-Man Kim(joomkim@pusan.ac.kr)

요약

본 논문은 원격 감시 제어를 위한 효율적인 웹 서비스 S/W 플랫폼을 제안한다. 웹 서비스의 성능 향상을 위해 서블릿 컨테이너를 제거함으로써 웹 서비스 구현의 필수 구성요소인 gSOAP 처리를 향상시켰으며, 로봇 또는 유비쿼터스 센서 응용에 적용이 가능한 경량화 된 웹 서비스 서버/클라이언트 소프트웨어 플랫폼을 설계하였다. 본 연구 결과를 검증하기 위하여 PXA270 프로세서를 탑재한 LDS4000 엔진 보드에 탱크와 센서 등을 결합한 감시 제어용 로봇 하드웨어를 제작하였으며, 웹 서비스 클라이언트 응용과 기존의 웹 서버를 통한 원격 감시 제어 기법 간의 gSOAP 메시지 교환 부하를 비교함으로써 본 연구 결과에 대한 실용적 우수성과 그 효율성을 검증하였다.

■ **중심어** : | 웹 서비스 | gSOAP | 원격모니터링 | 임베디드시스템 | 실시간제어 |

Abstract

In this paper, we propose an effective web service software platform for remote monitoring and control. We removed the servlet container for better web service performance so as to improve the gSOAP processing which is an essential element of web service implementation. Furthermore, we designed the web service server/client software platform which can be applied to robot or ubiquitous sensor applications. For validation of this study we tested it by manufacturing robot hardware for monitoring control which combined tanks and sensors on a LDS4000 engine board mounted with a PXA270 processor. The practical excellence and the efficiency of the result of this study was validated by the comparison of gSOAP message exchange load between the web service client application and the conventional remote monitoring control technique through a web server.

■ **keyword** : | Web Service | gSOAP | Remote Monitoring | Embedded System | RealTime Control |

I. 서론

웹서비스는 웹상에서 모듈화된 소프트웨어 컴포넌트

로서, 개방형 표준 데이터 표현 기법인 XML과 인터넷 프로토콜을 결합시킨 새로운 패러다임에 의해서 탄생된 차세대 분산 컴퓨팅 기술이다. 특히, 웹서비스의 상

* 본 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었습니다.

접수번호 : #070910-002

접수일자 : 2007년 09월 10일

심사완료일 : 2007년 11월 02일

교신저자 : 김주만, e-mail : joomkim@pnu.edu

호 운용성은 다양한 종류의 웹 서비스를 동적으로 발견하고 결합함으로써 부가 가치를 가진 새로운 형태의 복합 웹서비스의 창출을 가능하게 하였으며[1][7][8], 이러한 특징이 부각되면서 초기의 웹 서비스는 기업 내, 기업 간은 물론 공공 기관 간의 프로세스 통합 및 협력 자동화를 위한 핵심 기술로 적용되기 시작하였다. 그러나 다양한 분야로 웹서비스의 적용이 점진적으로 이루어지면서, 웹 서비스를 유비쿼터스 환경에 적용 시키려는 노력의 일환으로 주요 IT 기업인 마이크로소프트, IBM 등은 웹서비스를 이용하여 다양한 기술에 대한 연구를 진행하고 있는 상황이며 현재 웹 기술은 XML을 기반으로 프로그램 가능한 형태로 발전해 가고 있다[6].

웹 서비스 기술은 문서 교환을 위한 비즈니스 모델을 기반으로 기술 전개가 이루어졌으나, 최근 이기종간의 원격 제어의 방법으로 웹 서비스 기반의 웹 서버 연동 원격 감시 제어 기술에 적용하는 연구로서 인터넷 접속이 가능한 곳이면 시간과 장소에 제한을 받지 않고, 웹 브라우저를 통하여 원격 장치의 상태 정보 수집 및 제어를 가능하게 하여, 저 비용, 고 효율의 유비쿼터스 환경을 구축하는데 일조하였다[2-4].

웹 브라우저를 통한 웹 서버 접근 시 사용자 뷰의 리프레시 현상이 발생하는데, 이를 해결하는 방법으로 기존의 브라우저 플러그인이나 액티브-X 대신 비동기 자바 실행 환경을 제공하는 Ajax 기술을 채용하여 브라우저와 웹 서버간의 응답성을 높여주고, 웹 페이지의 변경 가능한 항목만 반영할 수 있어, 원격 모니터링 기술 구현 시에 끊임 없는 사용자 뷰 화면을 가능하게 하였다[4]. 또한 연구 [5]는 웹 기반의 HTTP 프로토콜을 이용하여 제어기에 CGI 병렬포트 제어 프로그램을 후면처리기로 두어 원격 감시 제어를 가능하게 함으로써 언제 어디서나 웹 브라우저를 통하여 원격 노드에 접근을 용이하게 하였다. 그러나 웹 서비스 클라이언트 환경에서 자바 가상 엔진과 Ajax 엔진 및 웹 브라우저를 탑재한다는 것은 고성능 데스크탑이 아닌 자원이 빈약한 모바일 단말기 실행 환경에서 어려운 현실이다.

본 논문은 원격 매체간의 상호 작용을 지원하는 정합 도구로써 웹 서비스 기반의 실시간 원격 모니터링 및 제어를 위한 소프트웨어 플랫폼을 설계하고 구현하는

데 있다. 본 연구의 구현 및 검증을 위하여 극한 또는 위해환경에 적합한 로봇을 가정하여 PXA270 기반의 LDS4000 보드를 이용한 하드웨어를 설계 제작하였다. 웹 서비스 서버/클라이언트 메시지 정합은 임베디드 시스템임을 고려하여 경량화 된 SOAP 라이브러리인 gSOAP을 적용하였다. gSOAP는 C/C++ 코딩으로 SOAP/XML을 통한 웹 서비스를 구현하는 툴로서 간편하게 웹 서비스 클라이언트 파일을 생성하며, WSDL 파서를 내장하여 자동 코드 생성이 가능하다는 장점이 있다[9]. 사용자 정합 도구인 모바일 단말기에는 웹 서비스 클라이언트 모듈인 gSOAP 라이브러리 및 GUI 정합 stub 으로 구성되어, 원격 웹 서비스 서버와 gSOAP 통신으로 원격 제어 및 모니터링이 가능한 소프트웨어 플랫폼을 구축하였다.

본 논문의 구성은 2장에서 개요 및 시스템 구성을 보이고, 3장에서 실제 설계된 부분을 소개하고 4장에서 구현방법에 대해 설명하며, 5장에서 시험 및 결과에 대하여 기술하여 6장에서 결론을 맺는다.

II. 개요

1. 시스템 구성

타겟 보드는 최소한의 센싱 기능과 프로세싱 기능을 가지며 최소한의 메모리, 저성능의 프로세서 등 제한된 자원을 갖는 임베디드 시스템 구조로써, GPIO 포트를 통한 다양한 로봇의 자원 구동 명령 세트를 제공하는 리눅스 기반의 웹 서비스 서버를 탑재하였다.

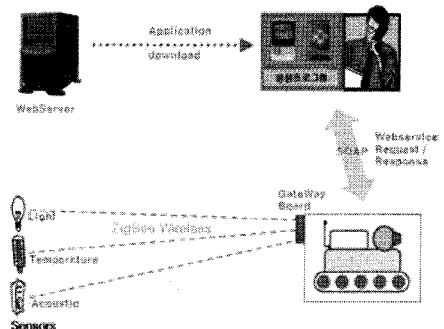


그림 1. 전체 시스템 구조

사용자는 웹 서비스 클라이언트단의 GUI(AP)를 통하여 로봇서버와 정합하기 위한 방법으로 gSOAP에 의한 XML 메시지 교환 방식을 적용한다.

[그림 1]은 본 논문에서 구현한 시스템 운용환경을 도시하고 있다. 사용자의 웹 서비스 클라이언트 응용 프로그램(AP)은 최적의 단말기 환경에 적합한 AP를 웹 서버로부터 다운로드 받아 설치하게 되며, 설치된 웹 서비스 클라이언트 응용 프로그램을 통하여 원격의 로봇에 직접 접속 가능함을 보여주고 있다. 사용자는 응용 프로그램을 통하여 원격의 웹 서비스 서버에 gSOAP 메시지 프로토콜에 의해 명령을 주고받을 수 있다. 타겟은 로봇의 기능을 가지며, ZigBee 무선을 통한 센서 노드들이 결합된 구조를 보여주고 있다.

2. 하드웨어 구성

2.1 로봇 엔진

로봇의 엔진부는 Corebell사의 LDS4000모델로써 PXA270-520MHz 프로세서와 64MBytes Flash ROM, 128MBytes SDRAM, RS-232C, 10/100 Base-T Ethernet 및 무선 LAN 카드 등으로 구성되었다. 장치 제어를 위한 4개의 GPIO 포트가 있는데, 로봇의 방향 및 센서 제어, 카메라 제어를 위해 최소 8개의 제어 시그널이 필요하여, 3V8 MUX를 별도 제작 구성하였다.

2.2 로봇 몸체

로봇의 몸체 및 구동부는 무선 RF 제어가 가능한 채인 방식의 탱크 모듈을 사용하였다. 회전 포탑에 웹 카메라를 장착하여 좌우 방향 제어가 가능하도록 하며, 로봇의 4방향 제어 및 멈춤 등의 리모컨 제어부를 분해하여 제작된 MUX와 결합하였다.

2.3 센서 노드

ZigBee 센서 네트워크는 기존의 블루투스, 무선랜등과 달리 독자적으로 네트워크를 구성할 수 있으며, 센서노드에서 감지된 데이터는 싱크 노드에 의하여 수집되어 요청 프로세스에 전달된다. 본 논문에서는 크로스보우사의 MICA2프로세서를 탑재하여 데이터 검출하는 MIB510 게이트보드, 온도·조도·음향 센서를 장착

한 MTS300 센서보드, Atmega128프로세서를 탑재해서 센서 값을 읽어 들이는 MPR 400CB 라디오 보드를 사용하였다.

3. 소프트웨어 구성

gSOAP은 객체에 접근하기 위한 프로토콜로서 원격에 있는 객체를 참조하기 위한 다양한 정의를 가능하게 한다. gSOAP은 단순하고 가벼우며, 산업 표준인 XML에 기반하고 있기 때문에 다양한 응용 도메인에 제한을 받지 않고, 비즈니스 프로세스뿐만 아니라, 산업 응용에 적용 가능한 표준 프로토콜로 그 활용 분야가 확대되고 있다.

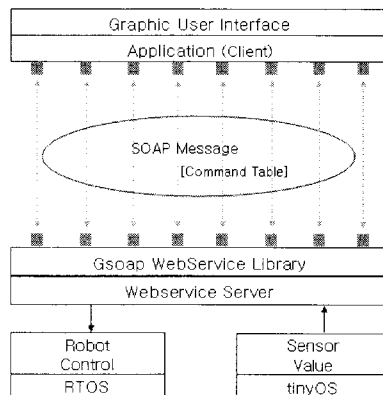


그림 2. 소프트웨어 구성도

[그림 2]는 본 연구의 서버/클라이언트 소프트웨어연동 구성도를 보여주고 있다. 사용자 클라이언트의 AP에 의해 gSOAP메시지가 웹 서비스 서버에 보내지면, 로봇의 웹 서비스 서버는 명령 수신 대기 모드에서 메시지를 받아, 메시지 내부 명령을 해독하여 후면 프로세스에게 사용자 요청을 전달하고, 로봇이나 센서 처리부의 수행된 결과를 gSOAP 메시지로 사용자 클라이언트 AP에게 돌려주는 과정이다.

III. 설 계

1. 하드웨어 설계

1.1 하드웨어 설계

로봇의 하드웨어는 모니터링 및 제어를 안정적으로 수행할 수 있도록 불필요한 기능은 배제하였고, 각각의 모듈을 최적화하여 필요로 하는 기능만을 수행 가능하도록 설계하였다.

[그림 3]은 임베디드 보드로 구성된 로봇의 블록 다이어그램을 도시하고 있다. 메인 보드인 PXA270 보드는 무선 랜을 내장하였으며, 기본적으로 USB, 시리얼 포트 및 4개의 GPIO 포트를 지원한다. 본 논문에서 가정하는 로봇은 운행 4방향 구동과, 카메라 좌·우 방향 제어 및 정지에 필요한 8개의 제어 접점이 요구되어 3 X 8 MUX를 설계 제작하여 포트를 확장하였다.

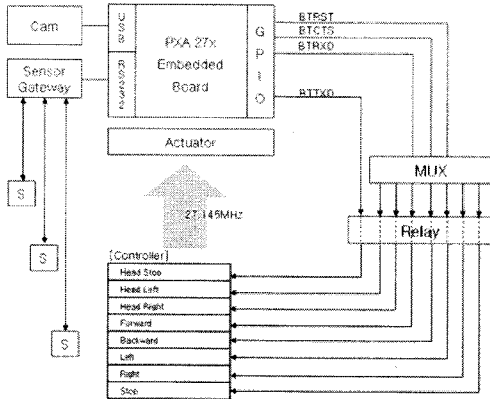


그림 3. 하드웨어 블록 다이어그램

USB 포트는 웹 카메라를 연결하고, 센서 게이트웨이는 시리얼 포트를 사용하였으며, 개발환경에서 요구되는 시리얼은 USB To RS232로 대체하여 구성하였다.

1.2 컨트롤러 제어를 위한 GPIO포트 확장 MUX

[그림 4]에서 GPIO 4개 포트 중 3개의 신호인 BTRST, BTCTS, BTRXD를 입력 받아 HD74LS11P, HD74LS04P 두 개의 IC로 결합하여 MUX를 구성하였고, 나머지 포트인 BTRXD 핀과 함께 타임 릴레이에 인가하여 8개의 제어 접점을 만들어 실행 가능하게 하였다.

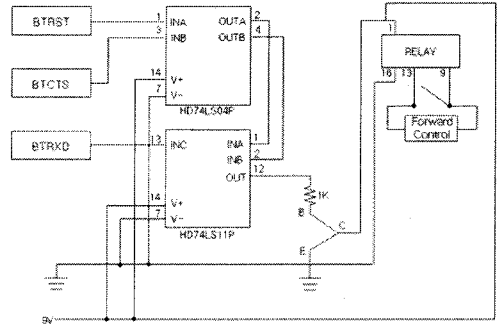


그림 4. 3X8 MUX Forward 회로도

1.3 센서노드 설계

본 논문에서 로봇에 적용한 센서는 [그림 5]와 같이 빛, 온도 및 음향 등 3종류의 센서를 지원하였다. 각 센서 노드에서 측정된 정보는 센서 게이트웨이를 통하여 메인 보드의 시리얼 포트에 접속된 베이스 스테이션으로 전달한다.

베이스 스테이션은 로봇의 웹 서비스 서버 후면 프로세스에게 패킷형태로 센싱 데이터들을 보내주면, 이 패킷을 분석하여 gSOAP 메시지로 구성하여 응용 프로그램에 전달한다.

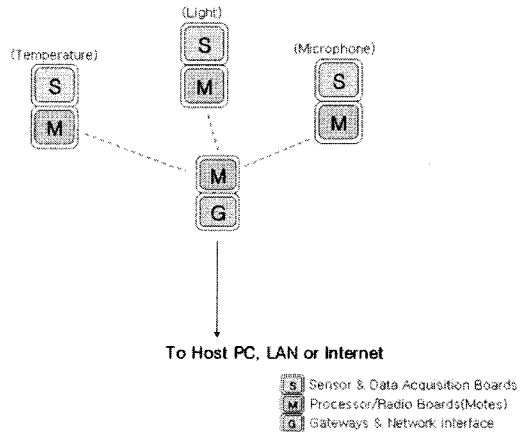


그림 5. 센서 시스템 구성

2. 소프트웨어 설계

2.1 웹서비스를 위한 WSDL설계

웹서비스는 WSDL(Web Services Description

Language)을 정의하여 기술한다. WSDL의 정의는 메시지, 오퍼레이션, 포트타입, 바인딩 및 서비스로 구성된다. 본 논문에서 WSDL 설계를 위해 우선, 포트타입에 대한 8개의 오퍼레이션을 정의하였고, 각각의 오퍼레이션에 대해서 파라미터 타입을 정의하여 gSOAP 입출력 메시지를 설계하였다. 바인딩은 포트타입에서 정의된 오퍼레이션의 인코딩 방식을 정의하였으며, 서비스는 웹 서비스의 물리적 접점 정보인 URL을 정의하여 웹 서비스 서버의 연결 주소를 명시하였다.

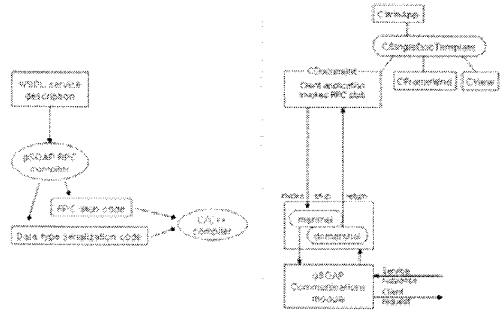


그림 6. gSOAP을 이용한 웹서비스 호출

2.2 웹서비스 클라이언트 코드 생성

WSDL 정의에서 바인딩 스타일을 RPC 지정하여 gSOAP의 전송 프로토콜로 정의하였다. 이 의미는 클라이언트가 웹 서비스 서버에 RPC 스타일의 gSOAP 메시지로 정합한다는 것이다. 따라서 클라이언트 코드에 RPC 함수 호출 방식으로 서버에 접속할 수 있다. RPC 함수는 gSOAP의 WSDL Importer를 이용하면 자동적으로 코드가 생성되며, 이 코드는 함수의 프로토타입과 데이터 타입이 선언되어 있다. gSOAP 컴파일러는 서비스 RPC Skeleton 과 데이터 타입 순차 코드를 자동적으로 생성하여 서비스 응용 프로그램이 웹 서비스로서 실행될 수 있도록 해준다.

2.3 어플리케이션(AP)

AP는 웹 서비스 클라이언트 모듈과 GUI 모듈로 구성되었다. 클라이언트 모듈은 gSOAP 라이브러리와 프로토타입 및 데이터 타입을 포함하여 stub 형식으로 코딩을 구현하였고, GUI 모듈에서 관련 stub을 호출하도록 설계하였다. GUI 모듈은 비주얼 스튜디오 도구를 사용하여, MFC 방식으로 프로그래밍 하였다. MFC는 윈도우즈 프로그램을 크게 4개의 클래스인 CWinApp, CDocument, CFrameWnd, 그리고 CView로 나누어 구성하는데, CFrameWnd와 CView클래스로 GUI를 구성했다. [그림 6]은 어플리케이션 프로그램에서 gSOAP을 이용한 웹서비스 호출방법을 도식화한 것이다. GUI 모듈의 CDocument 클래스는 RPC stub 부분에 코딩되어, GUI와 웹 서비스 서버와의 데이터 교환을 가능하게 하였다

gSOAP은 C/C++ 환경에서 웹 서비스를 활용하기 위한 라이브러리이다. 클라이언트 응용프로그램은 RPC stub과 자동 생성된 gSOAP 통신 모듈을 통해 원격의 웹 서비스 서버와 메시지를 교환한다.

IV. 구현

1. 하드웨어

[그림 7]은 타겟 하드웨어를 보여주고 있다. 무선 RC 탱크의 구동 부분과 리모컨 송신부를 이용하여 로봇 몸체를 구성하고, PXA270의 GPIO 포트의 확장 3X8 MUX의 출력을 RC 송신부에 결합하여 로봇 엔진을 구성하였다. 또한, USB 웹 카메라, 시리얼을 통한 센서노드 및 무선랜을 장착하여 전체 하드웨어를 구성하였다.

보드의 전원 공급은 1.2V 2700mAh규격의 배터리 4개를 직렬로 연결하여 해결하였고, 탱크의 구동 전원은 9V 내장 전원을 사용하였다.

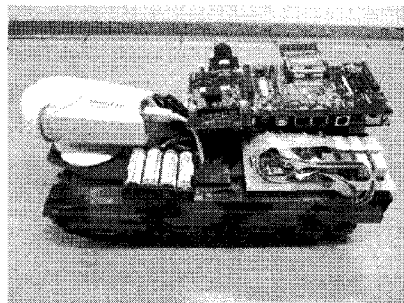


그림 7. 하드웨어 구현

1.1 웹캠의 영상처리

본 논문에서 사용한 카메라는 Omni사의 CCD방식으로 동작하는 OMUC-401로 USB방식의 웹캠이다. PXA270는 기본적으로 한 개의 USB 포트를 지원하여, USB 허브로 확장하기 위하여 커널을 재구성하여 캠을 설치했다. motion 영상 서버는 카메라로부터 획득한 영상을 JPEG 포맷으로 디스크 파일시스템에 저장하던 것을 램 디스크로 설정하여 영상 전송을 빠르게 하였다.

1.2 멀티플렉서

27.145MHz주파수를 가지는 무선 RC탱크의 리모트 컨트롤러의 8개의 접점을 제어하기 위하여 3X8 MUX를 제작하였다. PXA270에서 제공하는 4개의 GPIO 포트 중 BTRST, BTCTS 및 BTRXD등 3개를 MUX에 인가하면 8개의 출력이 나오지만, 그중 비활성 값을 가지는 0의 값을 가지는 접점을 제외한 7개를 사용하며, GPIO 포트중 한 포트(BTTXD)를 명령 포트로 릴레이에 인가하여 전체 8개의 출력을 갖도록 설계하였으며, 관련 명령어는 다음 [표 1]에 나타나있다.

표 1. 명령 테이블

BTRST	BTCTS	BTRXD	Command
0	0	1	Forward
0	1	0	Backward
0	1	1	Left
1	0	0	Right
1	0	1	Stop
1	1	0	Head Left
1	1	1	Head Right
BTTXD			Head Stop

1.3 센서노드

센서 노드는 크로스보우사의 MTS310A를 사용하였다. 조도, 온도, 음향 센서 노드로부터 추출된 raw 데이터는 시리얼에 연결된 데이터 검출보드와 게이트웨이 보드를 통하여 전송된 패킷에서 데이터 값만 추출하여 DB에 저장하였다. 웹 서비스 요청에 따라 저장된 센싱 값을 gSOAP로 전달할 수 있도록 하였다.

2. 소프트웨어

2.1 웹서비스

WSDL을 정의하여 메시지의 포트 타입에 대한 프로토콜과 데이터 형태의 바인딩을 구성하였으며 WSDL2CPP 도구를 통하여 AP에서 동작할 코드를 생성하여 웹 서비스의 호출과 응답을 가능하게 하였다.

표 2. 웹서비스 구현 소스코드

```

HeadLeft Request Message
<SOAP-ENV:Body SOAP-ENV:
encodingStyle=
http://schemas.xmlsoap.org/soap/encoding/
<ns2:HeadLeftRequest>
  <degree>0</degree>
</ns2:HeadLeftRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

HeadLeft Response Message
<SOAP-ENV:Body SOAP-ENV:
encodingStyle=
http://schemas.xmlsoap.org/soap/encoding/
<ns2:HeadLeftResponse>
  <code>0</code>
</ns2:HeadLeftResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
    
```

클라이언트 응용프로그램과 웹서비스 서버, 즉 로봇과의 통신은 SOAP 프로토콜을 사용하여 로봇과 응용 프로그램 간에 WSDL에서 정의한 입출력 형식에 따라 XML형식으로 표현되어 송수신 된다. [표 2]는 로봇과 AP사이에 웹 서비스의 실행 요청과 응답으로 송수신되는 XML메시지인데, HeadLeft명령에 대한 예를 보여주는 코드이다.

2.2 어플리케이션(AP)

어플리케이션은 윈도우즈용 응용 프로그램의 통합 개발 환경인 마이크로소프트 비주얼 C++을 사용했으며 MFC로 구현했다. [그림 8]은 MFC로 구현한 사용자 응용프로그램이다.

로봇의 영상은 영상서버와 연결되어 영상 이미지를 JPG형식으로 받아오게 하여 각각의 그림들을 슬라이딩 방식으로 처리하며, 이를 웹 컴포넌트 상단의 applet으로 구동시켜 영상출력부분을 구현하였다.

웹캠의 방향 제어 및 로봇의 구동에 대한 제어 버튼을 두었으며, 3종류의 센서는 로봇의 현재 위치에서의

센싱 값들을 프로그레스바로 나타내도록 화면을 구성하였다.

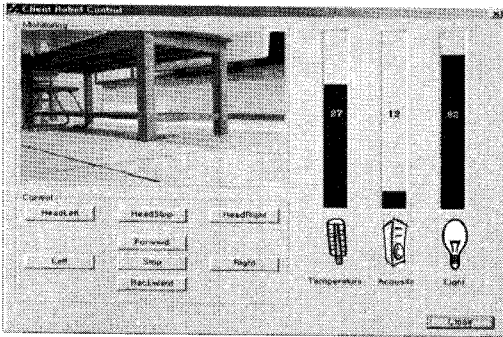


그림 8. 사용자 응용프로그램

V. 시험 및 결과분석

본 연구의 시험 목적은 기존 관련 연구와의 비교를 통하여 성능, 자원 이용률, 효율성 등을 제고하기 위하여 시도되었다. 시험을 위해 원격 장치에 대한 제어 및 모니터링의 응용 도메인을 가정하였고, 원격 장치로봇에 웹 서비스 서버가 구축되었다. 시험 대상 및 방법은 웹 서비스 서버와의 gSOAP 교환을 웹 서버의 하부 엔진을 통하느냐, 아니면 직접 웹 서비스 클라이언트 모듈과 메시지 교환하는가 하는 관점에서 비교 분석을 시도하였다. 즉, 본 논문에서 제안하는 모바일 단말기를 위한 가벼운 코드의 웹 서비스 클라이언트 모듈과 웹 서버와의 메시지 교환과 사용자 뷰로써 웹 브라우저를 채용하는 종전의 연구 결과를 비교하여, 향후 원격 모니터링을 위한 소프트웨어 플랫폼의 적절한 선택을 제시하는데 있다.

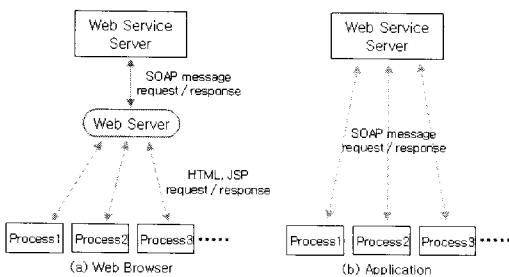


그림 9. 성능비교를 위한 클래스 분배

연구 방법은 [그림 9]에서와 같이 시험 대상 (a)는 웹 서버를 통하여 시험 환경 (b)는 웹 서비스 클라이언트 프로그램에 의해 직접 메시지 교환함을 도시키고 있다. gSOAP 메시지의 측정은 각각 웹 서버 ASIX 엔진과 응용 프로그램의 stub 에 두어 메시지 요청에 대한 결과의 시간을 측정하였다. (a)에서는 웹 브라우저에 ajax 를 수행시켜 서버와의 비동기 통신을 수행하지만 부하는 측정하지 않았다. 또한 영상은 두 경우에 CWebBrowser클래스를 이용하여 띄워진 웹브라우저 컴포넌트를 통해서 이미지를 자바에플릿의 슬라이딩 방식으로 처리하여 320X240 해상도로 초당 15프레임으로 CIF급 기준을 만족시켰다.

본 성능 비교를 위하여 웹 브라우저나 클라이언트 응용을 수행하기 위해 2.4GHz 펜티엄 4에 원격 로봇 제어를 위한 메시지 생성과 소비하는 임의의 프로세스를 작성하여, 동시 수행 가능한 프로세스 수를 10개까지 증가하며 시험하였다.

표 3. 성능 비교 테이블

proc	50		100		150		200		250	
	Ws	App	Ws	App	Ws	App	Ws	App	Ws	App
1	3.0	1.2	4.6	2.3	6.2	3.4	7.3	4.5	9.2	5.6
2	5.2	2.1	10.7	4.3	13.8	6.5	16.4	8.7	18.8	11.0
3	4.6	3.3	9.8	6.7	13.8	10.0	18.2	13.4	20.8	16.8
4	4.7	4.4	9.5	8.9	14.2	13.4	19.0	17.9	23.7	22.3
5	5.1	4.5	9.9	9.8	14.1	15.7	18.8	21.4	23.5	27.0
6	4.9	6.7	9.9	13.4	14.1	20.1	19.1	26.8	23.6	33.5
7	4.7	7.9	9.5	15.7	14.2	23.5	19.0	31.4	23.7	39.2
8	4.7	8.9	9.5	17.9	14.2	26.8	19.0	35.7	23.7	44.7
9	4.7	10.1	9.5	20.1	14.3	30.1	19.1	40.2	23.8	50.3
10	4.7	11.4	9.5	22.5	14.2	33.7	19.0	44.8	23.7	56.0

[표 3]에서는 각 10개까지의 동시 프로세스가 메시지 생성을 50개부터 250개까지에 대하여 웹 서버 기반과 응용 프로그램에서의 측정된 결과를 초 단위로 보여주고 있다. 측정된 결과 중 메시지 생성 횟수가 250인 경우에 대하여 [그림 10]에서 도시키고 있다.

[그림 10]에서 응용 프로그램인 경우, 동시 프로세스 수가 5개 이상부터 웹 서버 기반보다 상대적으로 성능이 저하됨을 보여주고 있다. 즉, 원격의 웹 서비스 임베디드 시스템의 열악한 컴퓨팅 환경으로 인해 동시 프로세스가 늘어나면 등차적으로 응답 속도가 늦음을 알 수

있고, 아울러 웹 기반에서는 웹 서버의 자체 부하로 인하여 프로세스의 수가 늘어남에 대한 부하의 변동이 없음을 알 수 있다. 이는 웹 서버와 웹 서비스 서버 간의 부하는 일정한 수준에서 포화되고, 메시지 교환은 웹 서버가 완충하고 있지만, 응용 클라이언트의 접근은 메시지 교환 및 처리의 주체가 웹 서비스 서버이기 때문에, 메시지 처리에 대한 선형적 증가에 대한 부하가 비례증가하고 있다.

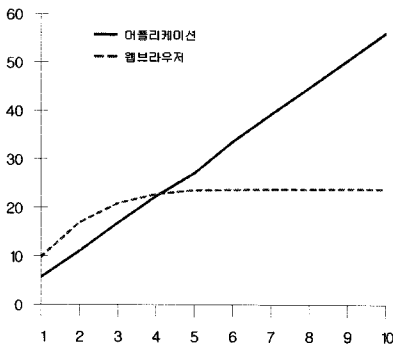


그림 10. 메시지 생성 250에 대한 성능 비교

본 연구에서 가정하는 원격 장치는 센서나 로봇과 같은 장치 접근 시, 동시 사용자가 많음을 가정하지 않으며, 아울러 접근 경로를 단순화하고 자원의 사용을 최소화할 필요성이 있다. 따라서 동시 접속 사용자가 작은 응용 도메인에서 경량화 된 웹 서비스의 규격을 만족하기 위하여 웹 서비스 클라이언트 응용 프로그램에 의한 소프트웨어 플랫폼이 적합함을 본 실험을 통하여 알 수 있었다.

VI. 결론

본 논문은 원격 제어 및 감시를 위한 웹 서비스 기반 소프트웨어 플랫폼을 설계하고 구현하였다. 기존의 웹 서비스 기술의 웹 서버 연동 방식과 성능 비교를 통하여 원격 감시의 효율적인 사용면에서 클라이언트 웹 서비스 응용 프로그램 방식이 우수함을 증명하였다. 본 연구의 결과는 극한 환경의 탐사 로봇을 가정한 하드웨

어를 제작하여 탑재 시험하였으며, 이러한 타겟 보드를 대상으로 성능 측정이 이루어 졌다. 비록 사용자가 5명 이상일 때, 웹 서비스 서버의 메시지 부하가 비례 증가 되지만, 실제 원격 탐사 로봇을 운용하는 주체는 소수의 특정인으로 한정되며, 또한 생성 메시지가 많지 않음을 예측할 수 있으므로 비교 우위로 결론을 내릴 수 있었다.

본 연구의 결과는 센서 노드나 자원이 열악한 소형 단말기의 원격 감시 및 제어용을 위한 표준 통신 소프트웨어 플랫폼에 적합하며, 아울러 본 연구가 향후 센서 네트워크에서의 이기종 노드간의 통신 소프트웨어 표준 플랫폼으로 확장할 수 있도록 추가적인 연구가 망된다. 또한 메시지 생성 주체가 클라이언트가 아닌 서버 단에서 이루어져 동시 양방향 요청/응답 기법이 가능하다면, 보다 폭넓은 응용 분야에 적용이 가능할 것이다.

참고 문헌

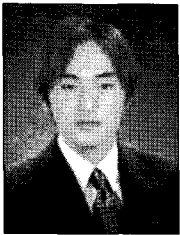
- [1] 이강찬, 이승윤, “웹서비스 표준기술 동향과 전망”, 정보과학회지, 제22권, 제10호, pp.11-18, 2004.
- [2] 이태희, 임동선, 김주만, “URC 로봇 원격 모니터링 기술 설계 및 구현”, 한국콘텐츠학회 춘계학술대회 논문집, 제4권, 제1호, pp.136-139, 2006.
- [3] 이태희, 서상희, 임성호, 임동선, 김주만, “웹 서비스 기반 원격 감시제어 S/W 플랫폼에 관한 연구”, 한국콘텐츠학회 추계학술대회 논문집, 제4권, 제2호, pp.70-74, 2006.
- [4] 임성호, 김주만, “웹 서비스 기반 URC 로봇 원격 모니터링 기술의 설계 및 구현”, 한국콘텐츠학회 논문지, 제6권, 제11호, pp.285-294, 2006.
- [5] 한경호, “웹 기반 하드웨어 원격감시 및 제어를 위한 초소형 내장형 웹서버 시스템의 구현”, Journal of KIEE, Vol.20, No.6, 2006.
- [6] <http://www.xml.com/pub/a/2002/02/06/webservices.html>

- [7] 이경하, 이규철, “웹 서비스의 표준화 동향과 발전 방향”, 데이터베이스연구, 제19권, 제1호, 2003.
- [8] E. Christensen and F. Curbera, *Web Services Description Language (WSDL) 1.1*, W3C Note, 2001.
- [9] <http://www.cs.fsu.edu/~engelen/soapdoc2.html>

저 자 소 개

이 태 희(Tae-Hee Lee)

준회원

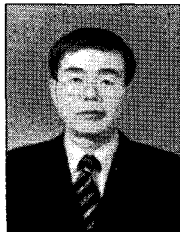


- 2005년 2월 : 밀양대학교 정보통신공학과(공학사)
- 2006년 3월 ~ 현재 : 부산대 바이오정보전자공학 석사과정

<관심분야> : 임베디드 시스템, 실시간 원격 제어, 웹 응용

김 주 만(Joo-Man Kim)

정회원



- 1984년 2월 : 숭실대 전자계산학(공학사)
- 1998년 8월 : 충남대 컴퓨터공학(공학석사)
- 2003년 8월 : 충남대 컴퓨터공학(공학박사)

- 1985년 1월 ~ 2000년 2월 : ETRI 책임연구원(OS팀장)
- 1995년 7월 ~ 1996년 6월 : Novell Inc. 방문연구원.
- 2001년 3월 ~ 2006년 2월 : 밀양대 정보통신공학부 조교수
- 2006년 3월 ~ 현재 : 부산대 바이오정보전자공학과 부교수

<관심분야> : 실시간 임베디드 시스템, 저전력 OS, 바이오 센서 제어, 유비쿼터스 컴퓨팅