웹 서비스 기반 URC 로봇 원격 모니터링 기술의 설계 및 구현

Design and Implementation of Remote Monitoring Technology based-on Web-Service for URC Robot

임성호*, 김주민**
한국전자통신연구원 로봇텔레메틱스연구팀*, 부산대학교 바이오정보전자**

Sung-Ho Im(shim@etri.re.kr)*, Joo-Man Kim(joomkim@pusan.ac.kr)**

요약

본 논문에서는 웹 서비스 기술을 적용한 새로운 URC 로봇 원격 제어 및 모니터링 기법을 제안한다. URC 로봇은 유비쿼터스 환경에서 외부 세계와의 다양한 인터페이스를 지원하기 위하여 이 기종 하드웨어 및 소프트웨어 플랫폼에 두루 적용될 수 있는 구조가 요구된다. 본 논문에서 웹 서비스 기술을 임베디드 환경에 최적화하고, 외부 응용 에이전트들과 상호 작용을 지원 할 수 있는 웹 서비스 기반의 로봇 원격 제어 방법을 설계 및 구현한다. 본 연구의 결과는 시뮬레이션을 거쳐 URC 로봇인 NETTORO에 탑재 및 운용하여 실용적 가치를 검증하였다.

■ 중심어 : □웹 서비스□URC 로봇□임베디드 시스템□원격 모니터링□유비쿼터스□

Abstract

In this paper, we propose a new remote control and monitoring technique using web-service technology for URC robot. URC robot needs the architecture which can be applied all over the variety hardware and software platform for supporting the several interface with external world in the ubiquitous environment. In this paper, web-service technology is preferentially deliberated how to adopt into the embedded environment and the remote control and monitoring technology based on web-service technology for URC robot is designed and implemented as to support the interaction with agent programs. It has been carried out through simulating and implementing this technology into the target robot called NETTORO and proved its practical worth.

■ keyword : □Web Service□URC Robot□Embedded System□Remote Monitoring□Ubiquitous□

I. 서 론

유비쿼터스 세상은 시간과 공간의 간격을 좁힌다는 의미로서, 서비스 사용자와 제공자는 주변의 여건에 가능한 의식하지 않아야 한다. 이와 함께 관련 기술에 대한 끊임없는 기술개발이 진행되면서 URC(Ubiquitous Robot Companion)라는 새로운 개념의 로봇이 태동하게 되었다[9]. URC로봇은 네트워크기술을 기반으로 언

접수번호 : #061016-003
접수일자 : 2006년 10월 16일
심사완료일 : 2006년 11월 02일
교신저자 : 김주만, e-mail : joomkim@pusan.ac.kr

제 어디서나 사용자와 함께 하며 사용자에게 필요한 다양한 서비스를 제공하는 로봇을 말한다. URC 로봇은 기존의 컴퓨터가 사용자의 사용여부에 따라 서비스를 제공하는 수동적인 측면이 강했다면, 이젠 직접 사용자에게 다가가 사용자가 원하는 서비스를 제공하는 능동적인 서비스를 제공한다.

본 논문은 URC 로봇과 사용자간의 상호 작용을 지원하는 정합 도구로써 실시간 원격 감시제어 기법을 설계하고 구현하는데 있다. 사용자는 로봇을 떠나 로봇의 행동 감시 및 제어는 물론 주변 환경에 대한 영상 모니터링을 웹 브라우저 및 응용 에이전트 뷰를 통하여 관찰하거나, 제어할 수 있도록 한다. 로봇의 원격 모니터링에 관한 관련 연구로 웹 서버를 통한 방법[10][11]이 있으나, 이 방식은 웹 브라우저를 통하여야 하며, 또한 로봇이라는 임베디드 환경에 웹 서버의 기능성 확장에 많은 제약이 있다.

웹 서비스 기술은 프로바이더에 의해 자동 생성되는 WSDL 언어를 배포함으로써 응용 개발자가 플랫폼에 제약받지 않고 독립적으로 응용프로그램 개발이 용이하게 하여 응용간의 통신 방식의 표준화를 달성할 수 있다[8]. 내장형 웹 서비스 기반 로봇 감시제어는 언제 어디서나 인터넷으로 연결된 컴퓨터와 이 컴퓨터에 설치된 웹 브라우저 또는 응용 프로그램만 있으면 사용할 수 있다는 장점이 있다.

본 논문에서는 웹 서비스 기술을 임베디드 실시간 리눅스인 Q+위에 적용하였으며, 네트워크상의 URC 서버에 웹 서버를 구축하여 AXIS 엔진에 의한 gSOAP 메시지 기반 통신 프로토콜을 구축하였다. URC 로봇은 웹 서비스 서버를 탑재하여 사용자에 의한 웹서버 또는 응용 에이전트 프로그램의 요청에 응답한다. 실시간 영상정보는 웹 서버를 통한 재경로 설정 후, URC 로봇의 포트를 통해 클라이언트 자바 애플릿에 의해 직접 표시하도록 한다.

본 논문의 구성은 2장에서 시스템 구성을 보이고, 3장에서 웹 서비스 기술과 관련 세부 기술을 소개한다. 4장에서 원격 모니터링 시스템 설계 방법론에 대해여 기술하고, 5장에서 그 구현 방법과 결과에 대하여 기술하여 6장에서 결론을 맺는다.

II. 시스템 구성

URC 로봇은 최소한의 자원을 가진 임베디드 시스템 구조이며, ZigBee 혹은 PLC를 통한 가전 기기 제어를 포함한 로봇 원격 관리 지원 서버로서 웹 서비스 서버를 두어 네트워크 상의 URC 서버와 연동하도록 설계 된다. 사용자는 웹 브라우저 또는 응용 에이전트 GUI를 통하여 로봇 서버와 정합하기 위한 방법으로 gSOAP에 의한 XML 메시지 교환 방식을 적용한다.

[그림 1]은 사용자가 로봇을 원격 모니터링 하기위한 운용환경으로써, 언제 어디서나 웹을 통하거나, 전용 응용 에이전트 프로그램을 통해 직접 로봇에 접속이 가능함

그림 1. URC 로봇 원격 모니터링 운영 시스템

을 보여준다. 사용자는 웹 브라우저 혹은 에이전트 응용 프로그램을 통하여 원격의 URC 로봇 서버에 접속할 수 있다. 사용자는 웹 브라우저를 통하여 웹 서버가 구동되는 URC 서버에 접속하면, 웹 서버의 후면 처리기인 AXIS 엔진을 통하여 gSOAP 메시지 프로토콜에 의해 URC 로봇의 웹 서비스 서버와 상호 연동한다. URC 로봇은 로봇 자체 기능뿐만 아니라, 홈 네트웍을 통한 가전기기 제어를 위해 PLC 및 ZigBee 통신 제어 모듈을 탑재하고 있다. 응용 프로그램은 URC 서버에서 다운로드 가능하여 전용 에이전트 응용 프로그램의 설치를 자동화 한다.

III. 웹 서비스 기술

1. 웹 서비스 구조

웹 서비스는 웹상에서 정의된 모듈화 된 소프트웨어 컴포넌트로서, 개방형 표준 데이터 표현 기법인 XML (eXtensible Markup Language)과 인터넷 프로토콜을 결합시킨 새로운 패러다임에 의해서 탄생된 분산 컴퓨팅 기술이다[8][12][13].

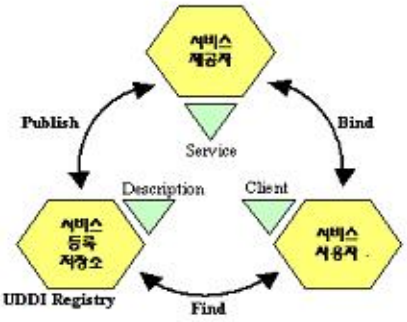

그림 2. 웹 서비스 오퍼레이션

[그림 2]는 웹 서비스 세가지 기본 오퍼레이션으로 웹 서비스의 사용은 역할에 따라 "서비스 제공자", "서비스 사용자", "서비스 등록저장소"로 나눌 수 있는데, 서비스 제공자는 자신의 자원을 개방형 표준에 의해 서비스 화 하여 등록저장소에 공개한다. 서비스 사용자는 서비스 등록저장소로부터 웹 서비스를 탐색하여 찾을 수 있고, 필요한 웹 서비스를 발견하면 서비스 사용자는 서비스를 호출하여 실행할 수 있다. 이를 위하여 웹 서비스에서는 WSDL, UDDI, SOAP의 세 가지 핵심기술을 활용한다.

1.1 SOAP(Simple Object Access Protocol)

SOAP는 XML로 인코드된 데이터를 전달하기 위한 표준적인 방법을 정의한 메시지 구조에 대한 명세이다[8][12]. 이것은 또한 메시지를 주고받는 양 끝단 간에 SOAP 메시지를 전달하기 위한 기반 커뮤니케이션 프로토콜로서 HTTP에 바인딩 하는 방법을 정의하고 응용 프로그램들의 연동이 가능하도록 한다.

1.2 WSDL(Web Service Description Language)

웹 서비스를 네트워크 단말 즉, 포트의 집합으로 정의하여 기술한다. 인터넷을 통하여 교환되는 데이터와 포트 타입에 대한 추상적인 기술을 메시지 추상화라 하며, WSDL은 메시지 추상적 정의를 재사용할 수 있도록 한다. 그리고 포트 타입이란 작업들의 추상적 모임이다. 특정 포트타입에 대한 구체적인 통신 프로토콜과 데이터 형태의 명세는 바인딩을 구성하며, 네트워크 주소를 바인딩 하도록 정의된다. 그리고 포트의 모임은 하나의 웹 서비스를 정의하며 응용 프로그램이 대상 웹 서비스를 사용할 수 있도록 한다[8][13].

1.3 UDDI(Universial Description, Discovery, and Integration)

서버의 사용자 또는 제공자가 웹 서비스를 발견하거나 등록할 수 있는 표준적인 방법을 제공하며 웹 서비스 제공자가 등록하는 웹 서비스의 정보를 저장하기 위하여 Business Entity, Business Service, Binding Template, Technical Model과 같은 핵심 데이터 구조를 정의하며 웹 기반의 글로벌 레지스트리 기능을 제공한다[8].

2. AXIS 개요

AXIS는 W3C의 SOAP의 표준을 구현한 라이브러리로서, C++과 JAVA 버전으로 배포되고 있다. 웹 서비스 구현의 핵심적인 기능을 지원하며, JAVA 에서 WSDL을 자동 생성하거나, WSDL 파일에서 웹 서비스 클라이언트 자바 코드를 생성하는 기능이 있어, 웹 서비스 개

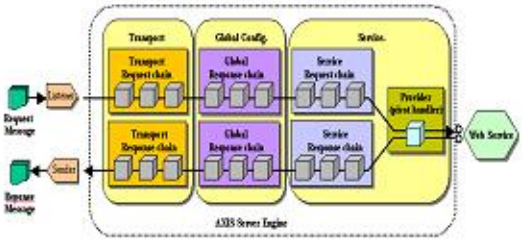

그림 3. AXIS 엔진 구조

발 주기를 단축할 수 있다. AXIS 엔진은 SOAP 메시지 처리 시스템으로 설정된 순서대로 체인을 호출하여

SOAP에서 정의한 내용이 정확히 지켜지는지 확인하는 역할을 수행한다. 또한 필요한 핸들러를 등록하여 웹 서비스 호출 및 응답과정에서 커스터마이징 할 수 있는 구조를 가진다.

[그림 3]에서처럼, 여러 개의 체인으로 결합되며, 각 체인들 간의 관계는 체인의 역할과 요청과 응답 이라는 흐름에 따라 결정된다. AXIS는 웹 서비스의 유연한 처리를 위하여 핸들러 기반의 서버 확장 기능의 제공과 SOAP 요청 메시지에 따른 웹 서비스 호출을 담당하는 프로바이더를 제공한다.

IV. 원격 모니터링 시스템 설계

1. 소프트웨어 모듈 구성

본 논문은 URC 로봇에 탑재되는 웹 서비스 서버와 인터넷상에 URC 서버에 탑재되는 웹 서버로 구성된다. 두 서버 간에는 gSOAP 통신 프로토콜에 의해 정합되며 웹 서버 하단에는 AXIS 엔진을 두어 XML을 gSOAP과 바인딩 할 수 있는 핸들러 기반의 서버 확장 기능을 수행한다. 웹 서비스 서버는 로봇의 3가지 주요 모듈을 수행하기 위해 WSDL을 작성 배포한다. WSDL은 웹 서버의 AXIS 핸들러로 구현되며, 또한 다른 응용 프로그램에서 통신 핸들러로 적용될 수 있다.

[그림 4]에서처럼, URC 로봇 서버는 로봇 제어 모듈, 로봇 모니터링 모듈 및 홈서버 연동 모듈로 구성된다. 각 모듈은 공통 모듈을 통하여 DBMS나 라이브러리를 통한 실시간 운영체제인 Q+에 정합한다. 로봇 엔진은 로봇 서버와 순차 접속을 하며, 공통 모듈은 여러 종류의 라이브러리를 제공한다.

2. AXIS를 이용한 웹 서비스 개발

AXIS를 이용하여 웹 서비스를 개발하고 배치하기 위하여 비교적 단순한 절차를 따른다. 본 연구에서는 JAVA를 이용하여 웹 서비스를 개발하므로, 웹 서비스 배치를 위한 WSDD(Web Service Deployment Descriptor)를 이용하였다. WSDD는 웹 서비스 배치는 물론, 핸들러와 체인을 등록할 수 있는 기능을 가지며, 웹 서비스 제공자와 사용자가 웹 서비스 제공 및 호출 환경을 커스터마이징 할 수 있도록 한다.

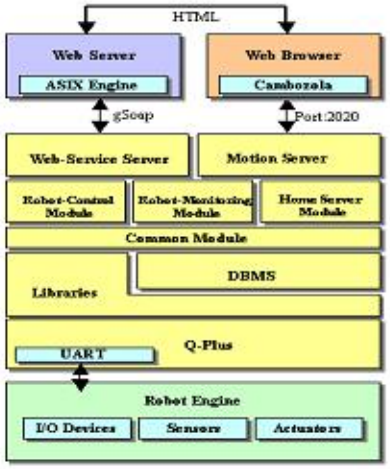

그림 4. 소프트웨어 모듈 구조

3. gSOAP을 이용한 웹 서비스 개발

gSOAP은 C/C++ 환경에서 웹 서비스 제공과 사용을 위한 라이브러리로서 본 연구의 URC 로봇 환경이 임베디드 환경으로 인해 JVM을 구축할 수 있는 자원이 부족하여 C/C++만을 지원하기위해 선택된 웹 서비스 도구인 gSOAP을 채택하였다.

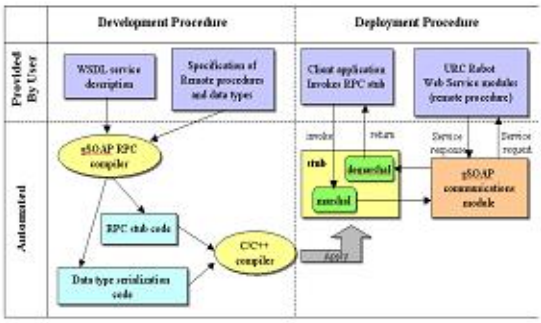

그림 5. gSOAP 개발 및 호출 절차

[그림 5]에서처럼 gSOAP를 이용하여 웹 서비스를 개발하기 위해서 서비스 응용프로그램은 RPC를 통해 해당하는 서비스를 제공하여야 하며, 이를 통해 외부에서

들어오는 웹 서비스 요청을 처리한다. 이 때 RPC 함수의 구현은 gSOAP에서 제공하는 gSOAP WSDL Importer를 이용하여 자동적으로 해당 코드를 만들 수가 있는데, 생성된 코드에는 서비스 함수의 프로토타입과 함수에서 사용하는 데이터 타입이 선언되어 있다. 또한 gSOAP Compiler는 서비스 RPC Skeleton과 데이터 타입 순차 코드를 자동적으로 생성하여 서비스 응용프로그램이 웹 서비스로서 실행될 수 있도록 해준다.

4. 웹 서비스 구축 방안

본 연구에서는 AXIS와 gSOAP를 이용하여 URC 로봇 기능을 웹 서비스화 한다.

로봇의 제어 및 모니터링은 gSOAP 를 통하여 구현되고, 사용자 권한 인증은 JAVA 클래스로 구현되어 AXIS 핸들러에 등록하여 처리하도록 설계 한다. 웹 서비스는 제한된 자원의 임베디드 시스템 환경인 로봇 서버에 구현되어야 함으로써, JAVA 환경 보다는 비교적 자원의 소모가 적은 C/C++ 개발 환경에 적응하는 gSOAP를 이용하여 구축한다. URC 서버는 인터넷상에서 필요시 URC 로봇의 서비스를 제공하는데, 그 방법으로 자바 톰캣 웹 서버를 탑재하고 있다. URC 로봇과의 gSOAP 통신을 구현하기 위하여 AXIS 엔진과 웹 서비스 호출 클라이언트 통신 모듈을 구축하여 구현 하였다.

5. URC 로봇 서버 구성

URC 로봇은 웹 서비스 서버와 영상 서버를 탑재 운영한다. [그림 6]은 URC 로봇 서버 구성 모듈로써, 웹 서비스 서버는 gSOAP 통신 모듈을 외부 인터페이스로 메시지 송/수신을 담당하며, 하부에 로봇 기능인 URC 로봇 제어 모듈, URC 로봇 모니터링 모듈 그리고 홈 서버 연동 모듈로 구성된다.

파일 DB는 플래시 메모리에 임베디드 실시간 DB 기능을 구현하여 로봇의 모니터링 정보의 추적 감시를 위한 로깅 및 로봇 좌표 맵 관리, 로봇의 위치 추적, 청소 영역 추적등을 위해 구현한다. 타겟 로봇인 NETTORO는 실시간 운영체제인 Q+를 탑재하여 NETTORO API를 제공함으로써 API 규격에 맞게 작성되어 실시간 감시 제어가 가능한 데몬 프로세스를 구동하고 있다. 영상 서버는 2020 포트를 개방하여 인증 서버로부터의 접속과 동시에 프레임 전송을 가능하게 한다. 웹 캠으로부터 발취된 영상은 램 디스크인 이미지 프레임 버퍼에 큐 타입으로 스트리밍 하여 포트를 통해 출력한다.

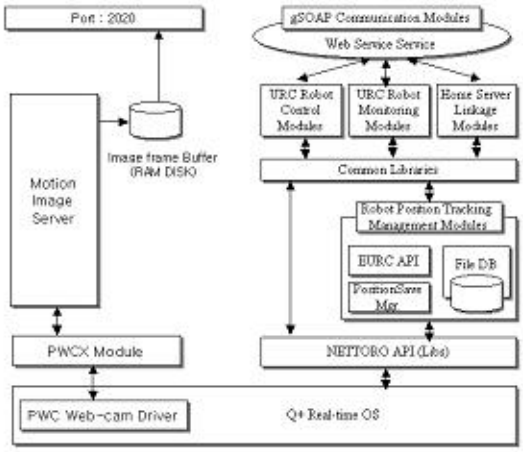

그림 6. URC 로봇 서버 구성 모듈

6. 설계 방법론

사용자와 관리자의 관점에서 Actor 맵을 설계하고, 각 기능별 User Case 및 그 Diagram을 도시하여 모듈간의 관계를 규정하였고, 각 모듈에 대한 Class 정의 및 Class Diagram을 도시하여 정의를 명확히 하였고, 각 엑터 별로 메시지의Sequence Diagram을 도시하여 각 모듈 간 흐름도를 설계하였고, 이를 바탕으로 사용자 인터페이스(UI) 흐름과 UI 화면 설계를 통하여 전체적인 구성모듈 및 모듈간 상세성과 명확성을 반영한 설계가 되도록 하였다.

7. 웹 서버

웹 서버는 URC 로봇을 위한 네트워킹상의 지원 서버로서 본 연구와 관련하여서는 웹 서비스 클라이언트로서 AXIS 엔진을 탑재한 JAVA 환경의 TOMCAT 서버이다. 웹 서버는 원격 제어 및 모니터링을 위한 다양한 웹 페이지로 구성되며, 로봇 사용자로부터 웹 서버에 내

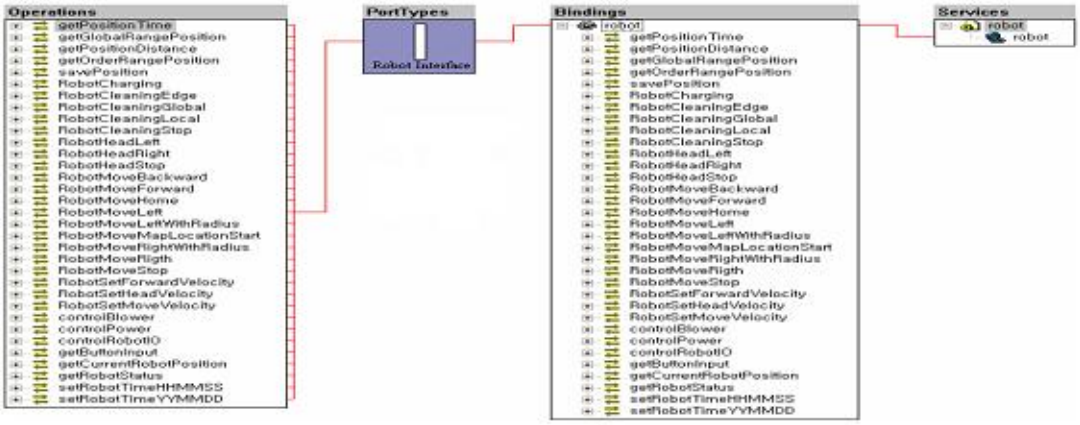

그림 7. 로봇 서버를 위한 웹 서비스 포트 정의

```
<?xml version="1.0" encoding="UTF-8"?>
<WSDL:definitions xmlns:tns="http://robot.pusan.ac.kr/robot.wsdl"
    <WSDL:types>
        <schema targetNamespace="urn:robot"
        </schema>
    </WSDL:types>
    <message name="getRobotStatusResponse">
        <part name="mode" type="xsd:int"/>
        <part name="battery" type="xsd:int"/>
        <part name="driving" type="xsd:int"/>
        <part name="head" type="xsd:int"/>
        <part name="error" type="xsd:int"/>
        <part name="reserved1" type="xsd:int"/>
        <part name="reserved2" type="xsd:int"/>
    </message>
    <message name="getRobotStatusRequest">
    </message>
    <portType name="RobotInterface">
        <operation name="getRobotStatus">
            <input name="getRobotStatusRequest" message="tns:getRobotStatusRequest"/>
            <output name="getRobotStatusResponse" message="tns:getRobotStatusResponse"/>
        </operation>
    </portType>
    <binding name="robot" type="tns:RobotInterface">
        <SOAP:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="getRobotStatus">
            <SOAP:operation style="rpc"/>
            <input>
                <SOAP:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:robot"/>
            </input>
            <output>
                <SOAP:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:robot"/>
            </output>
        </operation>
    </binding>
    <service name="robot">
        <port name="robot" binding="tns:robot">
            <SOAP:address location="http://168.188.60.160:8000/robotserver"/>
        </port>
    </service>
</WSDL:definitions>
```

그림 8. WSDL 정의 소스코드 (Operation : getRobotStatus)

려진 명령이 웹 서버와 URC로봇의 웹 서비스가 연동되도록 AXIS기반의 웹 서비스 클라이언트 모듈을 사용하여, 웹 서비스의 생성, 배치, 호출에 필요한 기능을 포함하였다.

V. 웹 서비스 구현

1. WSDL 정의 구현

WSDL은 웹 서비스의 입출력과 바인딩 그리고 서비스 위치에 대한 정보를 기술하는 표준 언어로 XML 문법으로 표현된다. WSDL의 정의는 메시지, 오퍼레이션, 포트타입, 바인딩 및 서비스로 구성된다. [그림 7]은 WSDL에서 오퍼레이션, 포트타입, 바인딩 및 서비스와의 관계를 도시하고 있다. [그림 8]에서는 오퍼레이션이 getRobotStatus에 대한 WSDL 작성 예를 보여주고 있다. 그림에서 WSDL를 설계하기 위해 먼저 PortType의 Operation을 정의하였으며, 각각의 입출력을 위한 메시지를 설계하였다. 메시지에서는 각각의 웹 서비스에서 사용되는 입출력의 파라미터와 타입이 정의된다.

Binding은 PortType에서 정의된 Operation의 인코딩 방식을 정의하며, Services는 웹 서비스의 물리적 접점 정보를 가지고 있다. gSOAP에 의해 전달되는 메시지는 로봇 제어 및 모니터링과 홈 서버 관리를 위한 서비스 호출 명령을 포함하여 34개의 오퍼레이션으로 정의하였다.

2. WSDL에 의한 웹 서버 코드 생성

URC 로봇 서버를 위하여 정의된 WSDL 코드는 웹 서비스를 호출하는 웹 서버 또는 에이전트 응용 프로그램의 웹 서비스 클라이언트 코드를 생성하는데 입력으로 사용된다. 본 연구에서 적용하는 웹 서버는 자바 톰캣 서버로서 내부에 AXIS 엔진을 탑재하고 있다.

AXIS는 WSDL 코드를 이용하여 자바 클라이언트 코드를 생성하기 위해 사용되는 도구로서 [표 1]은 AXIS의 WSDL2JAVA기능을 이용하여 만들어낸 자바파일의 리스트와 그 내용을 설명한다.

표 1. WSDL에 의한 웹 서비스 클라이언트 코드

JAVA 파일	설 명
Robot.java	URC 로봇 웹 서비스 호출 상위 인터페이스
RobotLocator.java	로봇 웹 서비스의 서비스 엘리먼트 이름과 접점 정보를 제공하는 클래스
RobotPortType.java	웹 서비스의 PortType에 포함된 Operation 인터페이스
RobotStub.java	RobotPortType에 대한 구현
RobotTestCase.java	RobotStub에 대한 JUnit기반 테스트 클래스

3. URC 로봇 영상 서버

URC 로봇의 눈의 역할은 웹 캠을 통하여 구현된다. [그림 6]에서 보듯, 실시간 운영체제 Q+에 PWC 와 PWCX 웹캠 디바이스 드라이버를 두고 상위에 Motion 영상 서버를 두었다. 클라이언트의 자바 애플릿 코드에 의해 URC 로봇 서버의 포트 2020을 엑세스하면, 초당 30 프레임의 영상을 스트리밍 처리할 수 있다. 또한 브라우저의 캠보졸라인 자바 애플릿은 웹 서버를 통하여 로봇 서버에 접근하지만 자바 통신의 보안 규정을 위배하게 된다. 따라서 본 연구에서는 Signed-JAVA를 적용하여 다음 [그림 9]처럼 URC 서버의 인증을 확보한 다음, URC 로봇 서버와 클라이언트 간에 직접 영상 전송이 가능하도록 하였다.

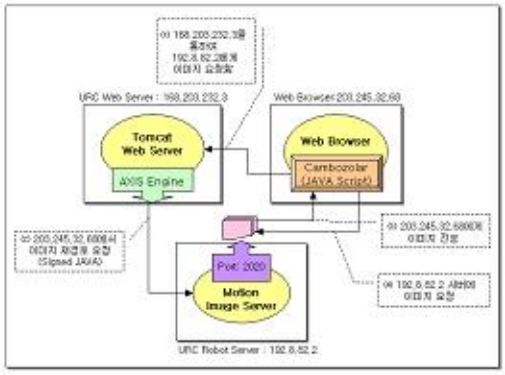

그림 9. URC 로봇 영상 전송 방식

4. 시뮬레이션

웹 서비스는 WSDL 정의 및 AXIS 엔진의 메시지 핸들러 구현으로 완성된다. 로봇 서버의 데몬 프로세스는 gSOAP 메시지 도착을 받아 처리하여 하단의 로봇 엔

진에 접속하게 된다. 로봇 서버는 로봇 엔진 프로세서와 시리얼 통신으로 접속되며, 본 연구의 타겟 구현을 위해 [그림 10]과 같은 시뮬레이션을 통하여 사전 검증 하였다.

[그림 10]은 로봇의 센서 및 엑츄에이터, 좌표, 상태 정보등 목표 시스템인 NETTORO URC 로봇의 대부분의 기능을 시뮬레이션 할 수 있도록 구성되었다.

그림 10. URC 시뮬레이터

그림 11. 시뮬레이션 과정 시연

[그림 11]에서 보는바와 같이 시뮬레이션을 위해 타켓 보드로 PXA270 보드에 URC 로봇 웹 서비스 서버를 탑재하고, 보드의 COM1 포트를 통하여 데스크 탑에 탑재된 [그림 10]의 시뮬레이터를 연결한다. URC 서버는 시뮬레이터가 수행 중인 데스크 탑에 웹 서버를 구축하였다. 시뮬레이터 입력 값이 원격 모니터링의 결과로 반영이 되는지를 확인하고, 원격 로봇 이동 제어를 통하여, 실제 시뮬레이터에서 좌표 이동이 제대로 동작 하는지를 확인 한다.

3. 구현 결과 : 모니터링 페이지

로그인 페이지에서 관리자로 로그인이 성공적으로 이루어지면, 웹 서버는 클라이언트에게 세션 번호를 부여하게된다. [그림 12]는 구현 결과물 중 대표적 화면으로서, 사용자 인증, 초기 접속에 따른 환경설정, 로봇의 영상, 로봇의 상태, 로봇의 움직임을 파악할 수 있는 위치정보 미니맵, 그리고 홈 네트웍을 통한 가전기기 제어 등에 대한 사용자 GUI 콘텐츠를 보여준다.

그림 12. URC 로봇 원격 모니터링 GUI

3.1 영상 정보 처리

로봇 영상은 초당 15 프레임에서 30프레임까지 해상도 조정이 가능하다. 30프레임 이상일 때, 무선 랜의 영향으로 끊김 현상이 있으나, 20 프레임 정도는 끊김없이 영상 표시가 원만함을 확인할 수 있었다.

3.2 로봇 맵 관리

미니맵은 웹 서버의 멀티 프레임 기술을 적용하고 프레임 수를 50x50으로 구성하여, 로봇의 움직임 좌표 및 청소 영역 관리를 목적으로 구현되었다. 로봇의 현재 위치가 표시되며, 맵 상의 특성 마우스 클릭으로 로봇을 이동하고, 이동 경로를 DB가 관리하여 필요시 맵에 표시되도록 하였다. 특정 시간동안 움직임이나 청소 영역을 표시함으로써 가사 로봇으로서 원격 관리의 효율성을 부가하였다.

3.3 로봇 모니터링

로봇의 현재 수행 모드, 현재 충전 상태, 이동 상태 및

머리 움직임 상태와 마지막 로봇에 반영된 명령어가 실시간으로 표시된다.

3.4 로봇 제어

로봇의 구동부 제어와 머리의 카메라 제어부로 구성된다. 구동부는 전방, 후방 및 좌, 우의 방향 전환 및 이동이 가능하며, 머리의 카메라 위치를 좌, 우 제어가 가능하도록 하였다. 기타 충전을 위한 도킹 스테이션 복귀 명령, 영화 보기, 음악 감상 등의 버튼을 가진다.

VI. 결 론

본 논문에서는 이기종간의 표준 방식과 응용 개발을 용이하게 하고, 웹 서비스 기술을 임베디드 환경에 최적화하여 URC 로봇에 적용함으로써 향후 국민 보급형 로봇인 URC 로봇의 원격 모니터링을 위한 표준 소프트웨어 플랫폼을 제안하며, 그 결과를 설계 및 구현하였다.

언제 어디서나 접속이 가능한 웹 브라우저 인터페이스를 위해 자바 환경의 톰캣 웹 서버를 구축하였으며, AXIS 엔진을 장착하여 로봇 서버와 gSOAP 프로토콜을 통한 웹 서비스 기술을 원격 제어 및 경량화가 요구되는 임베디드 환경에서 적용 가능함을 시뮬레이션과 타겟 URC 로봇에 구현하였으며, 2005년 소프트웨어 엑스포에 참가하여 그 가능성에 기반 한 실용성을 검증하였다.

URC 로봇의 웹 서비스화를 위해 34개의 로봇 이동 제어 및 상태 제어, 영상 제어, 청소 관리 제어 등의 제어와 홈 서버의 부가 기능을 제공하여 원격에서 웹 기반의 표준적인 방법을 통해 접근 가능하도록 구성하였으며, 향후 다양한 로봇 기능의 확장 시에도 유연하게 대처할 수 있는 설계 기법을 제공하게 되었다. 또한 웹 브라우저의 제한성을 고려하여 전용 응용 에이전트 프로그램을 통한 웹 서비스 기술 적용도 가능하게 하여 차세대 이동 단말기에 쉽게 적용할 수 있도록 하였다.

참 고 문 헌

[1] B. McCombie, "Embedded Web Servers now and in the future," Real-Time Magazine, No.1, pp.82-83, Mar. 1998

[2] A. Wilson, "The Challenge of Embedded Internet," Electronic Product Design, pp.31-34, Jan. 1998.

[3] I. Agranat, "Embedded Web Servers in Network Devices," Communication Systems Design, pp.30-36, Mar. 1998.

[4] 최재우 노방현, 이창근, 차동현, 황희융, "임베디드 웹 서버를 이용한 원격 감시 및 제어 시스템 구현", 한국산학기술학회 논문지, 제4권, 제3호, pp.301-306, 2003.

[5] I. Agranat, "Engineering Web Technologies for Embedded Application," IEEE Internet Computing, Vol.2, No.3, pp.40-45, May 1998.

[6] M. F. Zakaria, S. H. M. Amin, and R. Marmat, "Design and Development of Control System for Internet-based Telerobotics", Proc. TENCON 2000, Vol.2, pp.338-342, 2000.

[7] 정종근, 천승환, "웹 카메라 기반 홈 서버용 모니터링 기술에 관한 연구", 한국해양정보통신학회 논문지, 제9권, 제8호, pp.1754-1761, 2005.

[8] 이강찬, 이승윤, "웹 서비스 표준 기술 동향과 전망", 한국정보과학회지 웹서비스 특집, 제22권, 제10호, 2004(10).

[9] 이승익, 장철수, 정승욱, 김중배, "로봇 소프트웨어 아키텍처의 연구 동향과 현황", 전자통신동향분석 제20권, 제2호, 2005.

[10] 이태희, 임동선, 김주만, "URC 로봇 원격 모니터링 설계 및 구현", 한국콘텐츠학회 학술대회논문집, 제4권, 제1호, pp.136-139, 2006(4).

[11] 김주만, "URC 로봇 원격 모니터링기술 개발", 한국콘텐츠학회논문지, 제6권, 제8호, pp.8-19, 2006(8).

[12] http://www.cs.fsu.edu/~engelen/soap.html

[13] http://www.w3.org/TR/wsdl20/

저 자 소 개

임 성 호(Sung-Ho Im) 정회원

- 1986년 : 전북대 전기공학과(공학사)
- 1988년 : 전북대 전기공학과(공학석사)
- 2003년 : 전북대 메카트로닉스공학과(박사수료)
- 1988년~현재 : ETRI 로봇/텔레매틱스S/W플랫폼연구팀 책임연구원

<관심 분야> : 임베디드시스템, 지능형로봇, 텔레매틱스시스템

김 주 만(Joo-Man Kim) 정회원

- 1984년 : 숭실대 전자계산학(공학사)
- 1998년 : 충남대 컴퓨터공학(공학석사)
- 2003년 : 충남대 컴퓨터공학(공학박사)
- 1985년 1월~2000년 2월 : ETRI 책임연구원(운영체제연구팀장)
- 1995년 7월~1996년 6월 : Novell Inc. Research Center 방문 연구원
- 2000년 3월~2006년 2월 : 밀양대학교 정보통신공학부 교수
- 2006년 3월~현재 : 부산대학교 바이오정보전자 교수

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