

논문 2010-4-5

임베디드 시스템의 유기적인 동작을 위한 SOA기반의 S/W서비스 설계와 구현

Design and Implementation of SOA based S/W Services for Dynamic Behavior of Embedded System

박원규*, 박용범**

Won-kyu Park*, Young-bum Park**

요약 임베디드 시스템은 시스템의 특성상 사용자의 특정한 요구사항을 가지고 동작하게 되는데, 사용자의 요구사항이 바뀌거나 예상하지 못한 상황이 발생하게 되었을 때 유기적인 동작이 필요하다. 본 논문에서는 임베디드의 유기적인 동작을 위한 SOA(Service-oriented architecture)기반의 웹서비스 설계와 구현을 제안한다. 제안한 기법은 웹서비스를 통하여 임베디드 시스템의 상태를 알아보고 예외 상황이 발생할 경우, 그에 따른 동작을 웹서비스를 통하여 새롭게 업데이트 될 수 있게 하였다. 이를 통하여 사용자는 예외 상황에 대한 부담을 줄일 수 있을 뿐만 아니라 사용상의 편리성을 높일 수 있다.

Abstract As the nature of the embedded system, it is operated by user specified requirements, the dynamic action(behavior) is needed when the user's requirements change or unexpected situations occur. In this paper, it is proposed that design and implementation of SOA(Service-oriented architecture) based S/W services for dynamic behavior of embedded System. In this proposed technique, the status of embedded system can be checked through Web services, and in the cases of exceptional situations, the required proper actions can be newly updated through Web services. Through this technique, the burden of users concerning exceptional situations can be reduced, and the convenience of use can increased as well.

Key Words : WebServices, Embedded, SOA, MiddleWare

1. 서론

유비쿼터스 환경이 확산됨에 따라서, 임베디드 시스템은 우리 주변에 상당히 높은 비중을 차지하게 되었다.^[1] 그러나 현재 시스템의 운용 방식은 임베디드 기기들이 기존에 갖고 있었던 서비스로 동작을 하도록 되어 있어 시시각각 변화하는 사용자의 요구사항에 즉각적으로 적용하기 어렵다. 업데이트를 하는 기기들도 각각의 기

기에 사용자가 직접 업데이트 자료를 다운받고, 장비 간 케이블의 직접 연결을 해야 하는 등 번거롭고 어려운 과정을 거쳐 많은 시간과 비용을 소모한다.^[2]

기존의 연구는 웹서비스 어플리케이션을 통하여 임베디드 시스템을 통합함으로써, 임베디드 시스템간의 동작이 원활히 이루어 질수 있도록 하는 방법^[3]을 사용한다. 이러한 방법을 통해 임베디드 시스템을 통합할 수는 있지만 사용자가 직접적으로 업데이트를 해야 한다. 다른 방법으로는 블루투스 기반의 업데이트 기법^[2]이 있는데, 블루투스 기반으로 업데이트를 하게 되면 사용자가 직접 업데이트를 해야 하는 번거로움을 없앨 수 있지만 그 방

*준회원, 단국대학교 전자계산학과

**정회원, 단국대학교 전자계산학과

접수일자 2010.5.31, 수정일자 2010.7.16

게재확정일자 2010.8.13

법 또한 예외상황이 발생했을 경우의 유기적인 동작이 이루어지기 어렵다.

본 논문에서는 이러한 문제점을 개선하기 위해서 네트워크를 통하여 서비스를 자유롭게 다룰 수 있는 SOA(Service Oriented Architecture)의 한 방법인 웹서비스(Web Service)를 통하여, 임베디드 시스템에서 유기적으로 동작하는 서비스를 제공할 수 있는 시스템을 설계하고 구현하였다.

II. 관련 연구

1. 웹 서비스

웹 서비스는 어플리케이션 컴포넌트로서, 개방형 표준 데이터 표현 기법인 XML (eXtensible Markup Language)과 인터넷 프로토콜을 결합시킨 새로운 패러다임에 의해서 탄생된 분산 컴퓨팅 기술이다.^[4]

웹서비스의 사용은 역할에 따라 “서비스 제공자”, “서비스 사용자”, “서비스 등록자”로 나눌 수 있는데, 서비스 제공자는 자사의 자원을 개방형 표준에 의해 서비스화 하여 등록자장소에 공개한다. 웹 서비스에서는 WSDL SOAP의 핵심 기술을 활용한다.

가. SOAP(Simple Object Access Protocol)

SOAP은 XML 프로토콜 표준으로 Web Services의 요청 및 응답에서 사용되는 XML 메시지 형식과 통신 프로토콜과의 바인딩 방법을 정의하고 있다. SOAP은 분산된 컴퓨팅 환경에서 서비스 요청자와 서비스 제공자 사이에 구조적이고 유형화된 정보를 교환하는데 있어서 필요한 간단하면서도 경량의 XML 기반 프로토콜이라는 특징을 가지고 있으며, HTTP 프로토콜을 그대로 이용할 수 있어 기존 웹환경을 확장한 형태로 분산컴퓨팅 환경을 구현할 수 있다. 또한 방화벽에 의한 통신 장애에 대한 문제에 대해 고려할 필요가 없으며, 소켓 설정 같은 것도 필요하지 않다는 장점을 가지고 있다.

나. WSDL(Web Services Description Language)

WSDL은 웹 서비스를 기술하는 문법이다. 웹서비스가 수행하는 작업, 호출 가능한 메소드, 메소드에 전달해야 하는 파라미터, 파라미터 타입, 사용되는 바인딩 프로토콜 등 기술정보들을 정의하는데 사용된다. WSDL은

네트워크 상에서 메시지를 교환할 수 있는 종신점들의 집합을 정의할 수 있도록 XML문법을 사용함으로써 Web Services의 기술에 필요한 정보를 정의할 수 있게 한다.

Web Service는 일반 사용자나 어플리케이션의 서비스 요청자가 요구하는 서비스에 대한 메시지를 작성 한 후, SOAP이나 다른 특정 프로토콜을 사용하여 전달하고, 이 메시지를 받은 서비스 응답자는 응답 메시지를 작성하여 다시 서비스 요청자에게 보내는 일련의 과정으로 설명할 수 있다. SOAP을 통한 Web Services 이용시 WSDL을 이용하여 Web Services 정보를 기술하고, 전송에 사용되는 SOAP 메시지는 여러 파트들로 구성되는데 각 메시지 파트들은 XML Schema를 이용해 정의된 데이터 형들을 이용하게 된다.^[5]

2. Embedded MiddleWare

MiddleWare는 서로 다른 운영체제와의 호환성이나 이기종의 통신 프로토콜을 사용하는 네트워크 간의 접속, 네트워크 자원에 대한 접근 등을 연결해 단일한 사용자 환경으로 만들어 주기 위한 목적으로 등장한 소프트웨어이다.

Middleware의 한 종류인 Embedded MiddleWare는 두 가지 관점에서 분류할 수 있는데, 클라이언트와 서버 간 통신을 매개해 주는 관점에서 WAS(Web Application Server)와 CTI(Computer Telephony Integration) 미들웨어로, OS와 애플리케이션을 매개하는 관점에서 모바일 플랫폼 미들웨어, DMB(Digital Multimedia Broadcast) 미들웨어, 홈 네트워킹 미들웨어 등으로 나눌 수 있다.^[6]

III. 시스템 설계

본 장에서는 본 논문에서 제안하고 있는 임베디드 시스템의 유기적인 동작을 위한 전체적인 아키텍처와 각각의 모듈들을 설명하고자 한다.

그림1은 전체적인 동작 흐름도를 나타내는 것으로 Embedded System에서는 자신의 상태를 파악하고 그 상태에 맞게 서비스를 제공하는데, 자신의 상태가 변화했을 때 그에 맞는 서비스를 찾지 못할 경우 Middle Ware에 현재 상태를 나타내는 데이터를 bluetooth를 이용하여 보내고 이에 대한 응답을 기다린다.

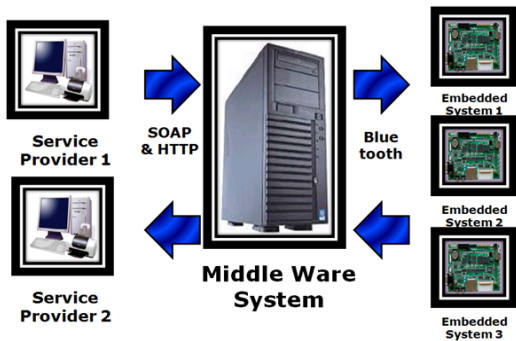


그림 1. 임베디드 시스템의 유기적인 동작을 위한 웹서비스 설계
 Fig.1 S/W service design for dynamic behavior of embedded System

MiddleWare에서는 받은 데이터를 네트워크를 통하여 서비스를 자유롭게 다룰 수 있는 웹서비스의 표준기술인 XML기반의 SOAP메세지로 변환하여, 인터넷 데이터 통신규약의 하나인 HTTP(Hypertext transfer protocol)를 통하여 인터넷으로 Service Provider로 보내고 이 Data를 받은 Service Provider는 그 상황에 맞는 서비스를 SOAP를 통하여 다시 MiddleWare로 보내어 Embedde System이 문제를 해결할 수 있도록 하였다.

1. Service Provider

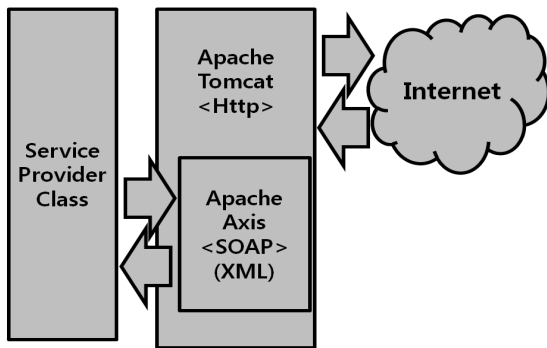


그림 2 Service Provider의 구조
 Fig. 2 Structure of Service Provider

Service Provider에서는 Internet을 통하여 받은 SOAP메세지를 Axis에서 객체로 변환 시킨 뒤 Load된 Service Class로 넘겨준다. 객체를 넘겨받은 Class는 개체 내부에 포함되어 있는 데이터로 Consumer가 필요로 하는 작업을 확인하여, 반환 값을 일정한 객체로 만들어 Axis에 제공하고, Axis는 다시 객체를 SOAP 메세지 로

변환하여 Tomcat를 통하여 Internet으로 전송한다.

2. MiddleWare 및 Embedded Device

Embedded Device에서 데이터를 얻기 위해 Middle Ware로 데이터를 전송하고, 전송된 데이터는 Middle Ware의 Network에 관련된 정보를 Class에서 전달 받고 이 데이터는 Embedded Device 마다 배정된 Thread에게로 보내진다.

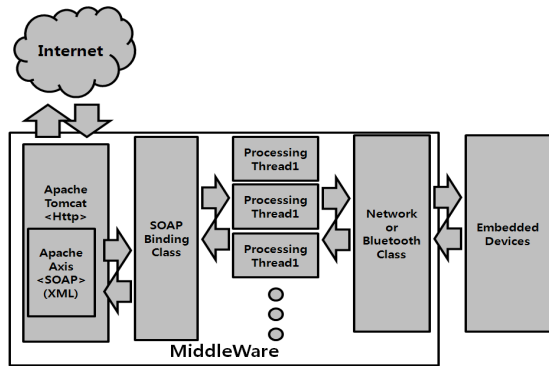


그림 3. MiddleWare의 구조
 Fig. 3 structure of MiddleWare

데이터를 받은 Thread는 자신의 Embedded Device가 필요로 하는 Service의 SOAP Binding Class를 찾아 매개변수로 사용할 데이터를 넘겨주며, 데이터를 넘겨받은 Class는 데이터를 객체로 만들게 되고 Axis를 통해 SOAP 메시지를 Service Provider로 보낸다.

Service Provider에서 응답한 SOAP메시지는 Axis 및 SOAP Binding Class를 통해 데이터로 변환되어 Thread로 전송된다. 전송된 데이터를 Embedded Device에 맞게 Header를 붙여 Network Class로 전송하면 Embedded Device가 데이터를 받아 동작 또는 업데이트를 실행하게 된다.

IV. 구현 및 평가

본 논문에서 제안한 시스템의 구조를 간단한 Embedded Devices로서 교육용으로 사용되는 Embedded Kit인 Robo Kit를 통하여 적용시키고 원활한 동작이 이루어지도록 구현하였다.

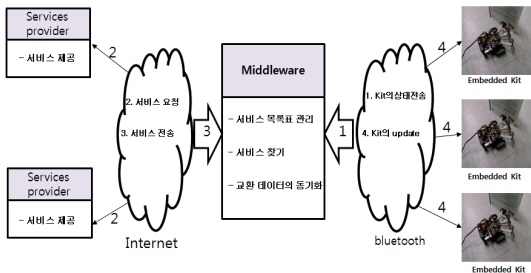


그림 4. 시스템 구현도
Fig. 4 materialization of system

그림4은 전체적인 구현도를 나타내었다. 임베디드 킷에서 예외상황이 발생하면 현재 상태를 블루투스를 통하여 Middleware로 전송되며, Middleware는 서비스의 목록표를 가지고 있어, 해당 상태에 대한 서비스를 찾아서 서비스를 요청한다. 서비스의 요청은 Internet를 통하여 되며, 서비스 요청을 받은 Service Provider은 해당 서비스를 Internet를 통하여 제공하게 된다. MiddleWare는 이 서비스를 제공받아서 블루투스를 통하여 Embedded kit으로 보내게 되면, Embedded kit의 업데이트가 완료되어 동작을 한다.

1. Embedded Devices의 구현

Embedded kit은 3개로 구성되어 있으며 각각의 Kit에는 4개의 적외선센서가 부착되어 있어 센서 값에 대한 동작을 하게 되어 있는데, Kit중에 2개는 라인을 따라 가는 동작을 하는 Kit이며, 1개는 앞에 있는 장애물을 인식하여 비켜갈 수 있는 Kit이다. 또한 이 Embedded kit들은 현재 자신의 동작 상태들이 입력되어 있으며, 예외 상황이 오면 언제든지 지금 상태를 블루투스를 통하여 Middle Ware에 전송할 수 있게 되어있다.

상태에 대한 프로토콜은 아래와 같으며, 한 번에 총 5byte의 Data가 전송된다.

표 1. Embedded Kit의 시리얼 프로토콜
Table 1. Serial Protocol of Embedded Kit

STX	모터 좌	모터 우	부저	ETX
0X02	0X00	0X00	0X00	0X03

2. Service Provider의 구현

본 논문에서는 2개의 Service Provider를 구축하여, Embedded kit에 대한 상태에 대한 Service를 각각의

Provider에서 할 수 있도록 하였으며, 하나의 Service Provider가 종료가 되더라도 다른 하나의 Service Provider의 서비스를 통해서 Embedded kit에 새로운 동작을 추가 시킬 수 있게 하였다.

```
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface SparePartServiceIF extends Remote
{
    public SparePartBean getSparePart(String PartSKU)
        throws RemoteException;
    public String addSparePart(SparePartBean spBean)
        throws RemoteException;
}
```

그림 5. 서비스의 추가를 위한 인터페이스 작성
Fig. 5. Interface drawing up for the addition of service

3. Middleware의 구현

Service의 목록들을 가지고 있어서 Embedded kit과 해당되는 Service를 찾아주고, 해당 Service를 가지고 있는 Service Provider와 Embedded kit의 Data 교환에 동기화를 시켜줘서 원활한 Data교환이 이루어질 수 있도록 한다. 여기서 Data는 두 가지로 나뉘는 데, 하나는 Embedded kit의 현재 상태를 나타내는 것을 Service Provider에게 보내는 Data이다. 다른 하나는 Service Provider에서 상태를 확인하여 새로운 형태의 동작을 나타내는 것을 Embedded kit에 보내는 Data를 말한다.

4. 모니터링을 통한 구동 확인

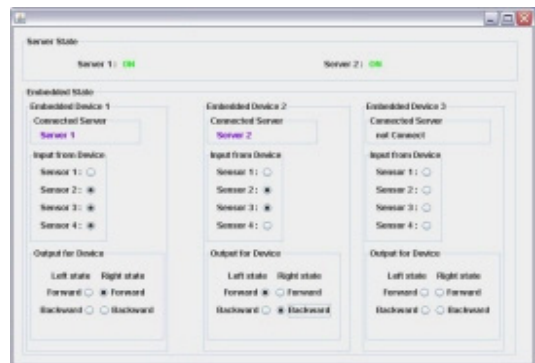


그림 6. 모니터링 화면
Fig. 6 Screen of Monitoring

그림 6은 실시간으로 모니터링이 가능한 상태를 나타낸다. 이를 통하여 확인할 수 있는 사항으로는 첫째, Service Provider가 정상적으로 동작하고 있는지를 확인할 수 있으며, 각각의 Embedded kit이 현재 연결되어있는 Service Provider도 확인할 수 있다. 둘째, 해당 Service Provider가 멈추었을 때 다른 Service Provider로 바뀌는 것도 확인이 가능 하다. 셋째, 각각의 Embedded kit 센서들의 상태를 확인하여 그에 따른 동작이 원활하게 이루어지고 있는지를 확인할 수 있다. 마지막으로 Embedded kit에 대한 예외 상태의 센서가 동작을 하였을 경우에는 모니터링을 통하여 확인 하고 새로운 동작을 추가 시킬 수 있고 그것을 바로 확인할 수 있다.

5. 평가

본 논문에서 제안한 웹서비스를 이용한 임베디드시스템의 설계방식을 구현하여 평가를 실행하였다.

표2는 기존의 연구에서 사용했던 기술과 본 연구에서 사용한 기술에 대한 요약표이다. 기존의 연구 1은 블루투스를 이용하여 임베디드 S/W의 업데이트를 자동적으로 해주어 사용자 편리성을 증가시키기 위하여 제안된 연구이며, 기존의 연구 2는 웹 어플리케이션을 통하여 임베디드 시스템을 통합함으로써 여러 기종의 임베디드 시스템을 제어가 가능하도록 제안된 연구이다.

본 연구에서 웹서비스를 사용하여 생기는 이점으로는 실제로 먼 거리에 있는 임베디드 장치에서도 동작이 이루어진다는 것과 원래 Service를 받고 있던 Service Provider가 멈춰도 다른 Service Provider의 같은 Service를 이용하여 Error를 막을 수 있는 이점이 있다.

또 다른 이점으로는 모니터링을 통하여 임베디드 시스템의 전반적인 상태를 알 수 있어 기기들의 상태를 관리자가 한눈에 파악할 수 있기 때문에 예외 상황이나 문제가 발생했을 시에 빠른 대처가 가능하였고, 임베디드 장비에서 센서의 상태를 파악해서 알려주기 때문에 다른 센서를 이용하거나 정보수집 장비를 이용한다면 실시간의 정보 수집이 가능하다는 이점도 발견할 수 있었다. 또한 블루투스를 사용함으로써 기존의 연구 1이 갖는 사용자의 편리성도 갖게 되었다. 그렇지만 한정된 임베디드 기기와 적은 수의 명령으로는 시스템이 커질 경우에도 원활히 동작하는지는 알 수 없었다.

표 2. 구현기술 요약표

Table 2. Summary table of materialization technical

구현	연구	기존 연구 ^[2]	기존 연구 ^[3]	현재연구
웹서비스		X	O	O
미들웨어		X	X	O
블루투스		O	X	O
모니터링		O	X	O

V. 결론

본 논문에서는 임베디드 시스템의 유기적인 동작을 위하여 SOA의 한 기술인 Web Service라는 기술을 도입하여, 임베디드 시스템들의 유기적인 동작이 가능하도록 시스템을 설계 및 구축을 해보았다. 그 결과 임베디드 시스템들의 업데이트 시간과 비용을 단축시킬 수 있었으며, 프로액티브(Proactive)한 기계를 만들 수 있고 서비스 제공자가 서비스를 중단하여도 다른 서비스 제공자를 통하여 지속적인 업데이트를 가능하게 하여 좀 더 효율적인 운용을 할 수 있다. 향후 제안된 설계가 다양한 임베디드 환경에서 적용 가능하도록 확장 연구 및 개발이 필요하다.

참고 문헌

- [1] 하태정, “국내 임베디드소프트웨어(ESW) 산업의 혁신특성”, 2009년 과학기술정책, pp.96-102P
- [2] 유길상, 남영진, 서대화, “블루투스 기반의 임베디드 S/W 자동 업데이트 기법 설계”, 제27회 한국정보처리학회 춘계학술발표대회 논문집 제14권 제1호, PP952-954P, 2007
- [3] 박영서, 박용범, “웹서비스 어플리케이션에서의 임베디드 장치의 통합 및 재구성에 관한 설계 및 구현”, 단국대학교 정보아키텍처 연구실
- [4] 한동수, “웹 서비스를 위한 예외상황기반 동적 서비스 연결 프레임 워크”, 2006년 정보과학학회 논문집, 소프트웨어 및 응용 제 33권 제 8호, pp668-680p, 2006
- [5] 이태희, “웹서비스 시반 원격 감시제어 S/W 플랫폼에 관한 연구”, 2006년도 한국콘텐츠학회 추계

- 종합학술대회 논문집, Vol4 제2호, pp.70-72P, 2006
- [6] 김정환, “임베디드 소프트웨어: 세계 모바일 미들웨어 시장”, 정보통신연구진흥원 학술정보 주간 기술동향 1167호
- [7] 조희형, 안성수, 안부영, 김경수, 박형선, “CCBB Web Services를 이용한 생명정보 데이터 분석”, 한국콘텐츠학회, 제32권, 제4호, 200-207쪽, 2007년 4월
- [8] Agranat, “Embedded Web Servers in Network Devices,”Communication Systems Design, pp.30-36, March 1998.
- [9] Simeon Simeonov 외 6인, 자바를 이용한 웹서비스 구축
- [10] 류광택, “웹 서비스 확산 방안발전 연구”, [한국전산원], (2004)

※ 본 연구는 2009학년도 단국대학교 대학연구비 지원으로 연구되었습니다.

저자 소개

박 원 규(준회원)



- 2009년 단국대 컴퓨터과학과 (학사)
 - 2010년 - 현재 동 대학원 전자계산학과 석사과정
- <주관심분야 : WebServices, 실시간 시스템, 정보아키텍처>

박 용 범(정회원)



- 1985년 서강대 전자계산학과(학사)
- 1987년 Ph.D N.Y.Polytechnic Univ(석사)
- 1991년 Ph.D N.Y.Polytechnic Univ(박사)
- 현재 단국대학교 교수

<주관심분야 : 패턴인식, 정보아키텍처, 멀티미디어 보안>