UPnP와 웹 서비스를 연계한 임베디드 미들웨어 설계 및 구현

정덕원°, 윤태웅, 김성민, 민덕기
건국대학교 컴퓨터 · 정보통신공학과
{dwchung°, taewoong, realight, dkmin}@konkuk.ac.kr

Design and Implementation of Embedded Middleware Service for Integrating UPnP and Web Service

Duckwon Chung°, Taewoong Yoon, Sungmin Kim, Dugki Min
Dept. of Computer Science & Engineering, Konkuk University

요 약

내부 시스템을 통합하는 비즈니스 관점에서 발전하던 웹 서비스가 모든 가정 내 통신과 가전제품을 연결하는 홈 네트워크와 같은 외부 시스템으로의 적용이 확대되면서 점차 보편화 되어가고 있다. 이런 외부 시스템으로의 적용을 위해서는 웹 서비스와 홈 네트워크 장치 간의 정보를 변환하고 제어하는 미들웨어 개발이 필요하다. 본 논문에서는 홈 네트워크의 자동 탐지 및 제어를 위한 UPnP 표준 기술과 서로 다른 플랫폼이나 언어로 작성된 어플리케이션 통합에 대한 표준 메커니즘을 제공하는 웹 서비스 기술을 이용하여 홈 네트워크나 유비쿼터스 환경에서의 분산 어플리케이션 개발 및 연계를 용이하게 하는 미들웨어를 설계 하고 구현하였다. 이 미들웨어를 임베디드 타입으로 개발함으로써 셋탑박스 또는 가정 내 모든 디바이스들에 내장 형태로 구성이 가능해짐으로 확장성을 제공한다.

1. 서 론

홈 네트워크[1]는 집안의 각 공간을 연결하여 정보를 전달하는 것을 의미한다. 이러한 홈 네트워크가 활성화되기 위해서는 네트워크 기술의 발전과 이를 바탕으로 운영되는 정보가전기기 그리고 이러한 통합 환경을 원활히 작동시켜주는 응용 소프트웨어의 발전이 함께 이루어져야 하기 때문에 무엇보다 관련기술의 완성도와 표준화 작업이 중요하다. 그러나 아직은 표준화에 부족한 부분이 있다. 그 부족한 부분이 바로 홈 네트워크를 다른 인터넷 기반의 분산 컴퓨팅 프로토콜과의 연계이다.

본 논문에서는 홈 네트워크의 디바이스 간 연결의 표준 미들웨어인 UPnP[2] 기술과 인터넷 기반의 분산 컴퓨팅 표준 프로토콜인 웹 서비스[3] 기술의 연계를 시도한다. UPnP 표준 기술과 웹 서비스 표준 기술을 연계한 미들웨어 서비스 개발을 통해서 다음과 같은 이점을 얻을 수 있다. 세부 통신 인프라와 미들웨어 서비스를 통하여 어플리케이션 개발이 용이해지고 앞으로 협력 미들웨어 프레임워크 개발을 통하여 시장의 적시성 확보할 수 있으며 기존의 웹 서비스의 분야에서 사용하는 기술을 그대로 UPnP를 사용한 홈 네트워크에서 사용할 수 있다.

이 논문의 구성은 2장 관련연구에서는 UPnP의 웹 서비스로의 발전과 웹 서비스 적용분야를 설명하고 3장에서는 논문에서 제시하는 홈 네트워크와 웹 서비스를 연계하는 아키텍처와 아키텍처를 구성하는 각 모듈들에 대하여 설명한다. 4장에서는 본 논문에서 제시하고 설계한 임베디드 미들웨어의 구현 예제를 보여주고 5장에서는 결론 및 앞으로의 향후 과제에 대하여 설명한다.

2. 관련연구

본 논문에서 제시하고 있는 홈 네트워크의 기술표준의 하나인 UPnP의 웹 서비스로의 발전과 웹 서비스 적용 분야를 설명한다.

2.1 UPnP의 웹 서비스로의 발전

UPnP는 SOAP[4]을 기반으로 한 현재 버전 1.0에서 2.0으로 업그레이드 되면서 WSDL 및 기타 웹 서비스 표준들을 수용함으로 더욱 웹 서비스 기반으로 나아가고 있다. UPnP 버전 1.0은 SOAP을 기반으로 하되 전체 프로토콜을 단순화함으로써 쉽게 적용할 수 있는 것을 목표로 하였기 때문에 단기간에 실제 응용에 적용하여 상용화를 이끌어 낼 수 있었다. 하지만, 복잡한 서비스를 표현하는 데 한계가 있다는 단점이 있었다. 이런 단점을 극복하기 위하여 차세대 UPnP 구조는 디바이스 디스크립션을 위해서 WSDL 지원, IPv6 지원, 디바이스 보안 모델 지원, 이벤팅을 SOAP으로 지원 하는 등 웹 서비스를 기반으로 변화하고 있다.

2.2 웹 서비스 적용 분야

HTTP와 XML을 기반으로 한 웹 서비스 기술은 Grid나 모바일 기술 2.1절에서 설명한 홈 네트워크로 까지 확장되고 있다. 본 절에서는 웹 서비스가 점차 확장되면서 적용되는 그리드와 모바일 분야에 대하여 설명한다.

(1) 그리드 웹 서비스

GGF[5] (Global Grid Forum) 에선 2004년 1월 20일 기존 OGSI를 대체할 웹 서비스 표준인 웹 서비스 WS-Resouce Framework[6]을 발표함으로서 기존 OGSA가 가지고 있었던 문제점을 해결 하려 하고 있다. WSRF는 웹 서비스를 변형 또는 확장하는 것이 아니라 WS-Addressing, WS-Resource Properties등의 표준을 수용, 개발함으로서 OGSI를 웹 서비스 형태로 바꾸는 것이다. WSRF는 WSDL 2.0 (2004년 2월 현재 Draft 상태)를 기반으로 하고 있고 6개의 핵심 표준인 WS-Resource Properties, WS-Resource Lifetime, WS-Notation, WS-Service Group, WS-Renewable Reference, WS-Base Fault 중 전자 세 개의 표준이 개발 완료된 상태이다. WSRF 표준이 개발되는 동시에 OGSI 기반의 OGSA를 WSRF 기반의 OGSA로 바꾸는 작업이 진행될 예정이다.

(2) 모바일 웹 서비스
모바일 웹 서비스는 기존의 웹 환경에서 제공되는 서비스들을 무선인터넷 환경에서도 도입하기 위한 서비스 기술이다. 모바일 웹 서비스 기술은 모바일 사용자의 능동적이며 지능적인 서비스 요구사항을 지원하며, 차세대 지능형 유무선 연동 서비스 시스템 설계를 위해 요구된다. 현재, 기존의 모바일 및 웹 서비스 기술은 다양한 종류의 플랫폼, 프로토콜, 어플리케이션 및 데이터 포맷을 기반으로 동작하며, 동종 시스템간에 서비스를 제공한다고 볼 수 있다. 또한, 모바일 서비스 기술은 모바일 단말의 하드웨어 성능 및 무선네트워크 특성 등을 고려한 설계가 필요하며, 웹 서비스 기술과 연동을 위한 다양한 인터페이스 기술이 요구된다.

3. 시스템 설계
본 장에서는 UPnP와 웹 서비스를 연계하는 전체 시스템의 동작을 설명하고 미들웨어 아키텍처와 각각의 모듈들을 설명한다.

3.1 시스템 동작 구조도
UPnP와 웹 서비스를 연계하는 시스템의 동작은 그림 1과 같으며 오른쪽 위와 왼쪽 위는 회사와 가정의 UPnP지원 디바이스와 UWS(UPnP and Web Service) 게이트웨이가 이루는 내부 네트워크 망을 의미한다. 이러한 여러 개의 가정이나 회사의 내부 UPnP 명령을 UWS 게이트웨이가 인터넷상에 존재하는 UWS 서비스 센터를 통해서 또 다른 외부에 있는 관리자가 제어한다. 그리고 오른쪽 아래에 있는 UWS 벤딩머신 네트워크는 이러한 외부 관리자가 UPnP기반의 모바일 장비를 통해서 접근하는 것을 의미한다. 또한 외부 관리자는 UPnP기반의 소형 모바일 장비 외에 외부에서 웹 서비스를 통해서 UWS 서비스 센터로 접근할 수 있으며 적절한 관리자 인증절차를 거쳐서 가정이나 회사의 디바이스 조정권을 얻게 되며 UWS 서비스 센터는 외부의 웹 서비스 접근을 위한 웹 서비스로 디바이스 제어에 대한 인터페이스가 노출되어 있으며 디바이스 제어뿐만 아니라 UPnP기반 디바이스의 이벤트 또한 외부에 있는 관리자에게 전달된다.

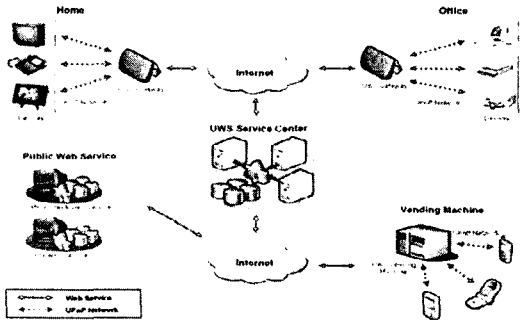

그림 1 UPnP와 웹 서비스 연계 시스템 동작 구조도

3.2 UWS 미들웨어 아키텍처
UWS 미들웨어 아키텍처는 그림 2와 같이 크게 한개의 메인 매니져 모듈과 6개의 각 기능 모듈로 구성된다. 본 논문에서 제시한 UWS 미들웨어는 하나의 통합된 미들웨어로 개발한 후에 UWS 게이트웨이와 UWS 벤딩머신에서 사용하였다.

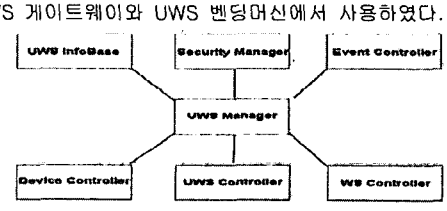

그림 2 UWS 미들웨어 아키텍처

(1) UWS Manager
UWS Manager는 UWS 미들웨어의 메인 모듈이다. UWS Manager는 UWS 게이트웨이와 UWS 벤딩 머신 양쪽에 다 포함되는 모듈이다. 나머지 6개의 기능 모듈에 대한 마더보드 같은 역할을 하며 외부의 컨트롤에 대한 접근 포인터가 된다.
가정의 회사의 인트라넷의 디바이스 제어를 위한 UPnP 컨트롤 포인터로써의 역할을 담당해야 한다. 또한 인터넷을 통한 웹 서비스 호출을 내부의 UPnP로 변환하거나 내부 인트라넷의 UPnP의 호출을 외부 인터넷의 웹 서비스로 변환하는 역할을 담당하기 위해서 웹 서비스를 제공해야 하고 처리해야 한다. 다시 말해야 UWS Manager는 UPnP 프로토콜의 Control Point가 되어야 하고 웹 서비스 제공자여야 한다. 이러한 외부 명령에 대한 Access Point로써의 역할을 한다.
나머지 기능 모듈을 사용해서 UWS Manager가 UPnP 명령을 웹 서비스 명령으로 변환하거나 웹 서비스 호출을 UPnP 컨트롤 명령으로 변환하는 작업과 그 밖에 이벤트 처리 작업을 하게 된다. UPnP와 웹 서비스 제어 명령 처리에 관해서는 UWS Converter를 통해서 하게되고 이벤트 처리 과정은 Event Controller를 통해서 하게된다.

(2) UWS Converter
웹 서비스 제어 명령과 UPnP 디바이스 제어 명령을 변환하는 일을 담당한다. 웹 서비스 제어 명령 즉, 웹 서비스 호출은 Device Controller를 사용하며 UPnP디바이스 제어 명령은 Device Controller를 사용한다. UWS 미들웨어의 기본 기능인 웹 서비스 및 UPnP 프로톨콜 변환을 UWS Converter에게 위임하는 이유는 차후에 웹 서비스 혹은 UPnP 프로토콜 외에 다른 프로토콜을 지원하기 위해서 이다. 왜냐하면 다른 프로토콜을 지원하게 변경할 경우 UWS Converter를 사용하는 측의 코드는 변경하지 않고 Device Controller와 WS Controller만을 변경하면 되기 때문이다.

(3) Device Controller
UPnP를 지원하는 디바이스를 제어하기 위한 모듈이다. 실제로는 디바이스 호출을 위한 UPnP SDK를 래핑하는 역할을 한다. UPnP SDK가 변경될 경우 미들웨어의 다른 부분을 변경하게 하지 않게 하기 위해서이다. 따라서 Device Controller에 기능은 UPnP SDK의 기능으로 전달된다.

(4) WS Controller
웹 서비스 호출을 담당하는 모듈이다. 이 모듈 역시 Device Controller와 마찬가지로 웹 서비스 SDK 혹은 라이브러리를 래핑하는 역할을 한다. 웹 서비스 호출과 관련된 라이브러리 변경에 따른 UWS 미들웨어의 변경을 최소화 하기위한 래퍼이다.

(5) UWS InfoBase
UWS Manager에서 사용하는 시스템 정보관리를 위한 정보를 체계적으로 저장하고 추출하고 검색하는 역할을 담당한다. UWS Manager에서 사용하는 시스템 정보 관리란 UPnP 네트워크에 존재하는 UPnP 지원 디바이스에 대한 디바이스 고유 이름과 디바이스 Description, 디바이스가 제공하는 서비스 리스트 정보, 각 서비스가 제공하는 액션 리스트가 저장된다. 디바이스가 제공하는 액션이란 디바이스를 제어할 수 있는 기능을 의미한다. 이러한 액션 정보는 액션 이름, 파라미터 정보, 리턴 값 정보가 포함된다. 또한 웹 서비스 호출과 관련해서 웹 서비스 호출에 대한 엔드 포인트 정보들이 부가적으로 저장된다.

(6) Event Controller
UPnP기반 디바이스에서 디바이스의 상태가 변함에 따라서 발생하는 이벤트를 UWS 미들웨어를 통해서 인트라넷이 아닌 인터넷을 통해서 멀리 떨어져 있는 곳까지 전달하기 위한 이벤트 처리 모듈이다. 이벤트 처리를 위한 이벤트 구독 신청이나 이벤트 전달을 위한 웹 서비스 인터페이스를 노출하고 있다. 이 웹 서비스 인터페이스를 통해서 인터넷을 통한 이벤트전달을 하게 된다. 즉, 외부에서는 UWS Manager를 통해서 이벤트 등록하고 등록 시 설정한 URL로 이벤트를 전달 받게 된다.

(7) Security Manager
UWS 미들웨어의 보안관리를 담당하는 모듈이다. 기본적으로 관리자들의 인증과 권한을 제어하기 위한 역할을 담당한다.

4. 구현 및 시나리오 테스트

UWS 미들웨어를 임베디드 타입으로의 구현을 설명하고 이를 적용한 예를 헬스센터에서의 시나리오를 기반으로 설명한다.

4.1 UWS 미들웨어 구현

구현은 플랫폼에 독립적인 이유와 UPnP SDK의 공개 버전인 자바 SDK(CyberLink SDK)를 위해서 자바로 개발되었다. 또한 게이트웨이나 벤딩머신에 탑재될 수 있게 하기 위해서 자바 1.1.8(퍼스널 자바)기반의 임베디드로 구현하여 그림 3과 같이 샤프 자우르스 PDA에 탑재하였다.

임베디드로 구현하기 위해서는 기존의 AXIS 웹 서비스 SDK이나 SOAP SDK로는 불가능하였다. 웹 서비스관련 SDK는 현재 PDA를 지원하는 웹 서비스 엔진이 없는 관계로 본 논문에서는 kSOAP 2.0[7]을 사용해서 PDA와 같은 모바일 장비에서 사용가능한 웹 서비스 SDK와 웹 서비스 클라이언트를 위한 라이브러리를 개발하였다. 또한 UPnP SDK도 Cyberlink SDK[8]를 기반으로 개발하였다.

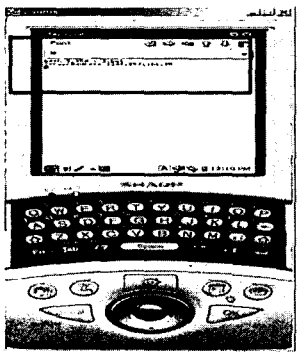

그림 3 UWS 미들웨어 구현

4.2 헬스 센터 적용 예

시스템 동작을 테스트하기 위하여 헬스 센터 시나리오를 가지고 실험하였다. 시스템 설계에서 UWS InfoBase를 통해서 로컬 네트워크에 접근할 수 있는 모든 UPnP 기반 디바이스에 대한 정보를 실시간으로 업데이트하는 기능이 UWS 미들웨어에 내장되어 있다고 설명하였다. 본 헬스 케어 시나리오는 그 실시간 정보관리에 대한 테스트를 하기 위함이며 시나리오는 다음과 같다.

(1) 컨트롤 포인트 등록
임베디드로 구현한 UWS 미들웨어는 그림 3과 같이 PDA에 탑재하였다. UWS 미들웨어는 컨트롤 포인트를 의미하며 운동기구에 대한 제어와 총 운동량에 대한 이벤트를 받아서 UWS 서비스 센터에 전달하는 기능을 담당한다.

(2) UPnP 디바이스 등록
헬스 센터 내부에 UWS 게이트웨이를 작동하고 나서 실제 운동에 필요한 UPnP로 구현한 런닝머신 및 스테퍼를 가동한다. 런닝머신 및 스테퍼를 가동하면 가동 즉시 UWS 게이트웨이 즉 UWS 미들웨어에 UPnP 프로토콜로 등록되고 컨트롤 포인트로 제어가 가능한 상태가 된다. 그림 4는 UPnP로 구현한 디바이스들을 등록하는 화면이다.

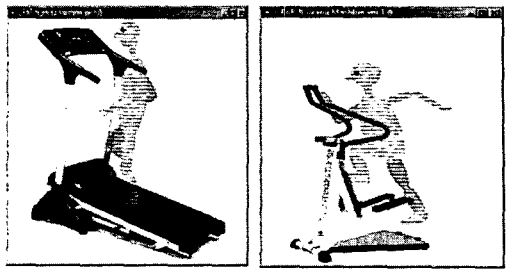

그림 4 UPnP 디바이스 등록

(3) UWS서비스 센터를 통한 통합된 총 운동량 확인
UWS 서비스 센터에서는 다양한 그리고 새롭게 추가될 UPnP 운동 디바이스에 대한 정보를 수집해서 저장하고 그 결과를 표나 그래프로 분석하는 일을 한다. 이렇게 사용자는 자신이 현재 운동한 총 운동량이 목표량과 어떻게 다른지 그리고 어떤 운동기구에서 어떻게 운동을 했는지를 정확히 알 수 있게 된다. [그림 5]는 UWS 서비스 센터를 통해서 현재 자신의 정확한 운동량을 분석하는 화면이다.

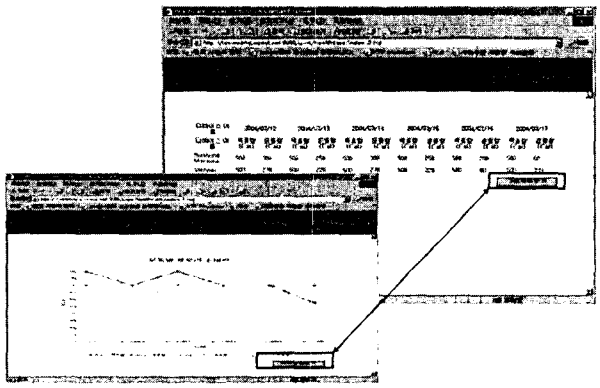

그림 5 UWS 서비스 센터를 통한 일별 운동량 확인

5. 결 론

UPnP와 웹 서비스를 연계하는 미들웨어를 설계하고 구현함으로써 서로 다른 플랫폼이나 언어로 작성된 어플리케이션 통합에 대한 표준 메커니즘을 제공하는 웹 서비스 기술을 이용하여 홈 네트워크나 유비쿼터스 환경에서의 분산 어플리케이션 개발 및 연계를 용이하게 해준다. 그리고 이런 미들웨어를 임베디드 타입으로 개발함으로써 가정 내 모든 디바이스들에 내장 형태로 구성이 가능해짐으로 해서 확장성을 제공한다.

6. 참고문헌

[1] Bill Rose, "Home Networks: A Standards Perspective" IEEE Communications Magazine, December 2001
[2] UPnP available at URL "http://www.upnp.org"
[3] Henry Bequet, "Beginning Java Web Service", WroxPress, 2003
[4] Don Box, "Simple Object Access Protocol (SOAP) 1.1" available at URL "http://www.w3.org/TR /SOAP/"
[5] GGF available at URL "http://www.ggf.org"
[6] WSRF available at URL "www-unix.globus.org/toolkit/docs/development/wsrf/"
[7] KSOAP available at URL "http://www.ksoap.org"
[8] Cyberlink SDK available at URL "http://www.cybergarage.org/net/upnp/java/index.html"