

유니버설 네트워크 상에서 웹서비스 프레임워크 구현

(Implementation of Web Services Framework for Web Services on Universal Networks)

임형준[†] 오일진^{**} 황윤영^{**} 이경하^{***}
 (Hyung-Jun Yim) (Il-Jin Oh) (Yun-Young Hwang) (Kyong-Ha Lee)

이강찬^{****} 이승윤^{*****} 이규철^{*****}
 (Kangchan Lee) (Seungyun Lee) (Kyu-Chul Lee)

요약 유니쿼터스 웹서비스는 어떠한 단말/네트워크 환경에서도 다양한 응용 서비스를 연계하여 이용할 수 있도록 하는 미래형 웹서비스 기술로 정의될 수 있다. 또한, 유니쿼터스 환경에서는 계속 유동적인 상태 변화를 갖는 디바이스가 가진 서비스의 동적인 검색과 이용이 요구되며, ad-hoc 환경에서의 검색 방법 등을 보장해야 한다. 이에 관련하여 최근에 트랜잭션, 보안, QoS, 시멘틱과 웹서비스 조합 등 여러 분야로 연구되고 있다. 하지만 최근 컴퓨팅 및 네트워크 기술의 발전으로 유니쿼터스에 대한 관심이 증가하고 있지만 아직 이에 대한 연구는 초기 단계에 있다.

유니쿼터스 환경에서는 기존 프로토콜 중 하나가 각종 디바이스 및 서비스의 위치, 정보, 상태 등의 유동성과 같은 발생하는 문제를 모두 해결하기는 어렵다. 이를 위해, 다양한 이질적인 서비스 디스커버리 미들웨어가 존재함에도 불구하고, 개인 전자기기 및 가전제품에 특화된 새로운 서비스 디스커버리 미들웨어가 생겨나고 있다. 따라서 본 논문에서는 산재한 이질적인 서비스 디스커버리 미들웨어들 즉, 유니버설 네트워크(Universal Networks)간의 상호운용성을 제공하기 위한 프레임워크인 WSUN(Web Services on Universal Networks)을 제안하고 구현한다. 본 논문에서 제안하는 WSUN은 웹서비스의 관점으로 접근하고 있으며 SOA 기반의 유니버설 서비스 브로커(Universal Service Broker)를 통해 서비스 디스커버리 미들웨어들 간의 상호운용성을 제공한다. 또한, 시나리오를 통해 WSUN의 필요성을 도출하였고, 유니버설 서비스 브로커가 지원해야 하는 요구사항과 범위를 정의함으로써 새로운 서비스가 등장하거나 퇴장하는 경우 등의 이벤트를 부가의 작업 없이 바로 적용하여 사용자로 하여금 유니버설 서비스(Universal Service)의 검색 및 연결이 가능하도록 한다.

키워드 : Web Services on Universal Networks, 유니버설 서비스 브로커, 서브네트워크, 상호운용성

Abstract Ubiquitous Web Services is able to be specified future Web Services technology for connecting with various application services in any device and network environments. The devices, in ubiquitous environment, have dynamic characteristic such as location and staturse. So, we must support methods of dynamic service discovery in ad-hoc network. There are many related works at

[†] 학생회원 : 충남대학교 컴퓨터공학과
 hyungjun25@cnu.ac.kr
^{**} 비회원 : 충남대학교 컴퓨터공학과
 victory25@cnu.ac.kr
 yhwang@cnu.ac.kr
^{***} 종신회원 : 충남대학교 소프트웨어연구소 연구원
 bart7449@gmail.com
^{****} 종신회원 : 한국전자통신연구원 표준서비스연구팀 연구원
 chan@etri.re.kr
^{*****} 비회원 : 한국전자통신연구원 표준서비스연구팀 팀장
 syl@etri.re.kr

^{*****} 종신회원 : 충남대학교 컴퓨터공학과 교수
 kclee@cnu.ac.kr
 (Corresponding author임)
 논문접수 : 2007년 9월 6일
 심사완료 : 2007년 12월 26일

Copyright © 2008 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용될 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 컴퓨팅의 실제 및 레터 제14권 제2호(2008.4)

transaction, security, Qos, semantic and Web Services composition with various fields. Recently, the studies are interested in the Ubiquitous by development of computing and network technology. However, they are an early stage.

For this reason, in this paper, we propose a WSUN(Web Services on Universal Networks) for Ubiquitous Web Services. It is a SOA based framework. And this paper extracts necessity of WSUN environment from scenario. The framework is composed of US Broker(Universal Service Broker). It is designed for satisfying the conditions and supports dynamic service discovery using a US Registry (Universal Service Registry). Consequently, clients are able to discover and use Universal Service by protocol stack of the US Broker for Web Services. And it is a strong point which supports interoperability between heterogeneous networks.

Key words : Web Services on Universal Networks, Universal Service Broker, Sub-network, Interoperability

1. 서론

최근 컴퓨팅 및 네트워크 기술의 급격한 발전으로 인해 유비쿼터스 환경에 대한 관심이 증가하고 있다. 이와 같은 추세를 반영하듯 개인 전자기기 및 가전제품에 프로세서를 탑재하여 독자적으로 정보를 처리하는 능력을 갖게 되어 다양한 디바이스 및 서비스가 등장하고 있다. 이들은 서로 다른 프로토콜 및 플랫폼을 기반으로 하고 있기 때문에 사용자가 서비스를 발견하거나 사용하기 위해서는 다양한 환경이 제공되어야 한다. 또한 유비쿼터스와 같은 분산 네트워크 환경에서는 각종 디바이스 및 서비스의 위치, 정보, 상태 등의 유동성에 의해 발생하는 문제를 기존 프로토콜 중 하나가 모두 반영하기는 어렵다. 따라서 사용자의 시간, 장소 등의 환경에 구애 받지 않고 서비스를 발견하고 사용하기 위해 이질적인 디바이스 및 서비스들 간의 상호운용성을 제공하기 위한 방법으로 웹서비스를 적용하고 있다.

초기의 웹서비스 기술은 주로 비즈니스 분야에서 다양한 응용에 대한 통합의 도구로서 이용되어 왔으나, 현재는 광대역통합망(BcN: Broadband convergence Network) 기반의 유무선 통합 응용, 방송/통신 융합, 정보 가전/홈네트워킹, 임베디드 환경 등 IT839 전략의 다양한 분야에서 핵심 연동 기술로 그 활용 범위가 빠르게 확산되고 있다. 이처럼 웹서비스는 단순한 비즈니스 응용 분야의 확장뿐만 아니라 단말, 네트워크, 사용 환경의 다양성과 서비스 환경의 변화/융합을 포괄하는 유비쿼터스 웹서비스로의 발전이 필수적이다[1].

유비쿼터스 웹서비스(Ubiquitous Web Services)는 어떠한 단말/네트워크 환경에서도 다양한 응용 서비스를 연계하여 이용할 수 있도록 하는 미래형 웹서비스 기술로 정의될 수 있다. 유비쿼터스 웹서비스는 계속 동적으로 변하는 디바이스에 대한 능동적인 서비스의 검색과 이용이 요구되고, ad-hoc한 환경에서의 검색 방법 등을 지원해야 한다. 이에 따라 트랜잭션, 보안, QoS, 시멘틱

과 웹서비스 조합 등 여러 분야로 연구되고 있다[2].

최근 유비쿼터스 웹서비스를 지원하기 위한 기술로 BEA, 마이크로소프트, 썬 마이크로시스템즈 등에서 WS-Eventing을 개발, OASIS에서 WS-Base Notification의 표준 개발, IBM, Sonic 등에서 WS-Brokered Notification의 표준 개발 등이 있다. 마이크로소프트에서는 모바일 웹서비스(Mobile Web Services)와 디바이스를 위한 웹서비스(Web Services on Devices) 기술, 유비쿼터스 환경을 위한 Invisible 웹서비스 플랫폼을 개발 및 표준 개발을 더욱 적극적으로 추진하고 있으며, IBM은 유비쿼터스 기술 분야에서의 장점을 활용하여 웹서비스 기술 개발 전략을 추진하고 있다. 국내에서는 IT839 서비스의 융복합을 위해 소프트 인프라웨어(Soft Infracore)로서 웹서비스를 적극적으로 활용할 계획 중이며, 시멘틱 검색 및 서비스 매치메이킹(Match Making) 등의 기술은 대학 및 연구소를 중심으로 연구가 진행되고 있다.

그러나 국내의적으로 유비쿼터스 웹서비스 연구는 초기 상태로 아직까지 다양한 요구 사항을 충분히 반영하지 못하고 있는 실정이다.

앞서 살펴본 바와 같이, 서로 다른 프로토콜 및 플랫폼에 기반을 둔 디바이스와 서비스의 위치, 상태, 정보 등의 유동성에서 생기는 문제는 서비스 디스커버리 미들웨어(Service Discovery Middleware)를 통해 해결할 수 있다. 본 논문에서는 서비스 디스커버리 미들웨어를 서브네트워크(Sub-network)로 지칭한다. 대표적인 서브네트워크로는 썬 마이크로시스템즈의 JINI(Java Intelligent Network Infra-structure)[3], 마이크로소프트의 UPnP(Universal Plug and Play)[4]와 HAVi(Home Audio/Video Interoperability)[5] 등이 있으며, 디바이스간의 통합을 위한 연구로는 DPWS(Devices Profile for Web Services)[6]가 있다. 하지만 하나의 서브네트워크가 서로간의 상호운용성을 지원해주지 못할 뿐만 아니라 다양한 물리 계층에 대한 이질적인 특성을 모두

반영할 수 없다. 다양한 서브네트워크가 존재하는 반면에 산재한 서브네트워크의 이질성을 극복하는 방안은 아직 미흡한 실정이다.

따라서 본 본문에서는 이질적인 서브네트워크를 유니버설 네트워크(Universal Networks)로 하여 이기종 서브네트워크 환경과 다양한 컴퓨팅 환경이 융합된 환경에서 웹서비스를 연계·이용할 수 있도록 WSUN(Web Services on Universal Networks)을 제안한다. 서브네트워크 및 디바이스, 서비스들 간의 상호운용성을 위한 WSUN의 유니버설 서비스 브로커(US Broker: Universal Service Broker)는 SOA 기반의 미들웨어로 다양한 서비스 검색을 위한 유니버설 서비스 디스커버리 프로토콜(USDP: Universal Service Discovery Protocol)을 포함한다. 또한, 동적인 서비스 검색을 위해 유니버설 서비스 레지스트리(US Registry: Universal Service Registry)를 구성한다. 이를 통해, 플러그 앤 플레이(Plug-and-Play), 프로그램 언어에 독립적, 보안 등의 기능을 지원할 수 있다. 디바이스와 서비스의 위치 및 상태 정보를 반영한 다양한 서비스 발견 방법은 사용자에게 필요한 서비스의 검색을 효율적으로 제공한다. 이러한 프레임워크의 제안은 유비쿼터스 환경의 변화에 대한 적절한 대응 및 전략적 비전을 제시하며, 유비쿼터스 웹서비스를 통해 경쟁력 있는 시스템 개발 및 서비스의 재사용과 웹서비스 응용 서비스의 제공을 기대할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 먼저 서브네트워크의 상호운용성을 제공하기 위해 진행되고 있는 관련 연구들에 대해 소개하고, 3장에서는 시나리오를 통해 WSUN의 필요성을 설명한다. 4장에서 WSUN의 중요한 구성 요소인 유니버설 서비스 브로커의 요구 사항 및 범위, 기능, 프로토콜 스택, 구조에 대한 설계를 기술하고, 5장에서는 설계를 기반으로 WSUN에 대한 구현과 검증을 설명한다. 마지막 6장에서는 결론과 향후 연구방향에 대하여 논한다.

2. 관련 연구

이번 장에서는 이질적인 서브네트워크의 상호운용성을 위한 기존의 연구에 대해 살펴본다.

표 1은 서브네트워크간의 통합을 위한 연구와 디바이스간의 통합을 위한 연구 및 서브네트워크의 특징을 나타낸다.

가장 대표적인 서브네트워크인 JINI와 UPnP는 이질적인 특성을 가장 잘 반영하기 때문에 OSGi(Open Service Gateway Initiative)[8]와 DomoNet(Domotics Network)[9,10] 등에서 사용한다. OSGi는 서비스 게이트웨이로 서비스를 로컬 네트워크나 디바이스에 전달하

표 1 서브네트워크와 디바이스간의 통합을 위한 연구[7]

	OSGi	HAVi	JINI	UPnP	WS	DPWS
Plug-and-Play	-	○	○	○	-	○
Device support	○	○	○	○	-	○
Programming Language independent	-	○	-	○	○	○
Network media independent	-	-	○	○	○	○
Large scalability	○	-	○	-	○	○
Security	○	○	○	-	○	○
High market acceptance	○	-	○	○	○	○

고 전달된 서비스를 운용하는 개방적 표준을 제공한다. DomoNet은 서브네트워크의 서비스를 웹서비스로 변환하여 이질적인 서브네트워크의 상호운용성을 보장한다. 이러한 서브네트워크의 통합을 위한 연구들에서 사용하는 JINI는 자바 기반으로 자바 가상 머신(Java Virtual Machine)이 필요한 단점이 있지만 플러그 앤 플레이와 보안 등 여러 특성을 포함하고 있다. 반면에 UPnP는 프로그래밍 언어에 독립적이지만 광대역의 네트워크 환경을 지원하지 못하며 보안에 취약한 문제를 지니고 있다. 이질적인 환경의 상호운용성을 제공하기 위한 웹서비스는 다수의 특징을 지원하지만 플러그 앤 플레이를 지원하지 못하기 때문에 동적인 서비스 검색이 불가능하다. 또한, 디바이스 관점의 웹서비스를 제공하지 못하기 때문에 유비쿼터스 환경에서 제대로 적용할 수 없다. 웹서비스가 가지는 문제점을 해결하기 위한 일환으로 DPWS가 있으며 디바이스 관점의 웹서비스를 지원하여 유비쿼터스 환경의 특징을 반영한 UPnP의 차세대 기술로 각광받고 있다.

2.1 OSGi의 확장[11]

[11]은 이질적인 서브네트워크의 통합을 OSGi를 이용하여 서비스 프레임워크를 제공한다. 이는 OSGi가 가지는 서비스 게이트웨이와 분산 미들웨어의 장점을 혼합한 형태로 나타난다.

다음은 OSGi의 게이트웨이가 가지는 특성이다.

- 애플리케이션 서비스: 모든 애플리케이션에 적용 가능한 서비스를 제공한다.
- 이식 가능한 애플리케이션: 프레임워크가 애플리케이션 자체로 이식 가능한 코드와 구성 파일들을 포함한다. 즉, 애플리케이션과 라이브러리의 구성 요소들을 프레임워크가 관리한다.
- 애플리케이션 관리: 프레임워크는 설치, 시작, 중지, 변경, 삭제의 과정을 어떤 애플리케이션이나 프로그램으로 동작한다.
- 인터페이스 서비스: 코드의 재사용을 위한 애플리케이션

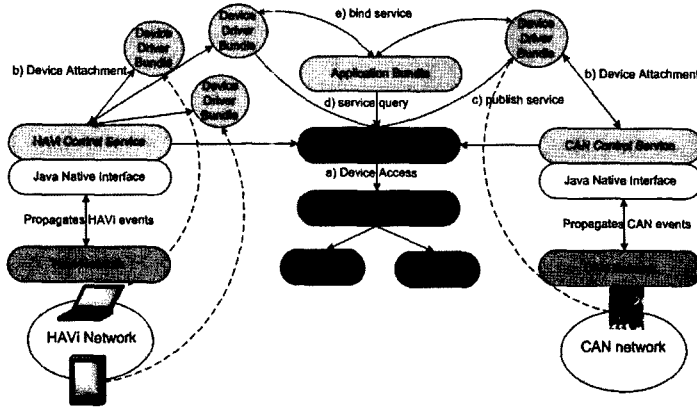


그림 1 OSGi를 확장한 구조

선 간의 상호운용을 위해 애플리케이션을 모듈화 한다.
 • 안전성 증대: 에러나 시스템 결점을 발견하여 이에 대한 해결을 한다.

그림 1은 컨트롤 서비스(Control Services)를 통한 디바이스 연동을 위한 OSGi 확장한 구조이다. 이는 OSGi 레지스트리와 디바이스 위치기, 디바이스 선택기로 구성된 OSGi 디바이스 매니저 등을 통해 서브네트워크와 연결을 한다. [11]에서는 OSGi의 게이트웨이 가지는 특성을 이용하기 위해 서브네트워크마다 컨트롤 서비스를 생성하여 OSGi의 게이트웨이와 연결하는 인터페이스를 제공한다. 또한, 이벤트에 대한 처리를 담당하여 OSGi의 프레임워크에서 처리하지 못하는 메시지를 변환하여 전달한다.

2.2 DomoNet

DomoNet는 서브네트워크의 서비스를 웹서비스로 변환하여 이질적인 서브네트워크의 상호운용성을 보장하기 위해 SOA 기반으로 접근하고 있다. 이를 위해, 하나의 서비스를 웹서비스로 변환하는 기능을 담당하는 TM (TechManager)가 존재한다. TM은 그림 2와 같이 각 서브네트워크마다 위치하여 모든 서비스에 대해 변환하는 과정을 서브네트워크에 종속적으로 동작하기 때문에 효율적으로 사용된다. DomoNet 네트워크와 서브네트워크의 연결을 담당하는 TM은 서브네트워크에서 제공하는 서비스들을 DomoNet 환경에서 사용 가능하도록 하

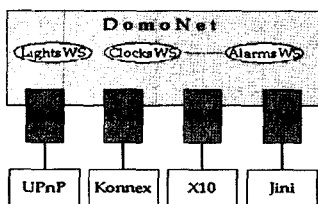


그림 2 DomoNet 구조

기 위해 서브네트워크의 상태에 따라 웹서비스로 변환하는 동작을 한다.

2.3 연구 동기

본 장에서 설명한 바와 같이 서브네트워크의 상호운용성을 제공하기 위한 연구들이 있다. 관련 연구들은 이질적인 서브네트워크의 문제점을 인지하고 문제를 해결하기 위한 접근 방법은 다르지만 각각 해결할 수 있는 방안을 제시하였다.

표 2는 이질적인 서브네트워크의 상호운용성을 제공하기 위한 요구 사항으로 관련 연구들과 본 논문에서 제안하는 WSUN을 비교한다.

[11]은 OSGi를 연동하기 위해 각 서브네트워크의 디바이스마다 디바이스 드라이버 번들(Device Driver Bundle)이 존재한다. 이 디바이스 드라이버 번들은 [11]에서의 컨트롤 서비스와 1:N으로 연결이 되어 동작하는데 OSGi의 특성을 반영하고 있기 때문에 새로운 디바이스가 생길 때마다 수동적으로 생성해야 한다. 따라서 동적인 서비스 발견은 지원하지 못한다. 반면에 DomoNet과 WSUN은 각각 TM과 유니버설 어댑터를 통해 실행 단계에서의 플러그 앤 플레이를 지원한다.

관련 연구와 본 논문에서 제안하는 시스템은 모두 현

표 2 이질적인 서브네트워크의 상호운용성을 위한 연구

	[11]	DomoNet	WSUN
Plug-and-Play	-	△	△
Transparent access to each device	○	○	○
Management of events and errors	△	-	△
Maintaining a consistent description	○	○	○
discover available service	-	-	○

재의 서브네트워크 특성을 변형하지 않고 서브네트워크의 상호운용성을 제공하고 있기 때문에 디바이스의 접근할 수 있는 투명성(transparency)을 제공한다. 또한, 디바이스의 명세 정보를 유지하는 레지스트리를 사용한다.

[11]은 각 서브네트워크에서 발생하는 이벤트 정보를 컨트롤 서비스를 통해 중재하고 있으며, WSUN도 서비스의 상태 정보 등을 이벤트 메시지를 받아 처리를 하고 있다. 하지만 WSUN에서는 서비스의 상태 정보를 이벤트로 받아 레지스트리에 서비스 사용가능 정보를 담고 있기 때문에 사용자가 서비스 검색을 할 경우 사용가능한 서비스만을 검출하여 제공하는 이점이 있다.

이 이외에도 [11]은 OSGi의 특성인 자바 기반으로 구성되어 자바 환경을 갖추고 있지 않으면 서비스를 제공하지 못할 뿐만 아니라 다른 언어와의 연동을 위해 Java Native Interface(JNI)를 부가적으로 사용해야 하는 단점이 있다. 또한, DomoNet은 서비스의 수정이 일어나면 매번 웹서비스로 변환해야 하는 번거로운 작업이 있으며 다양한 서비스 검색 방법을 제공하지 못한다.

본 논문에서는 이러한 문제를 해결하기 위한 방안으로 유니버설 네트워크를 위한 웹서비스 프레임워크인 WSUN을 제안한다. 본 논문은 WSUN을 통해 플랫폼이나 언어에 독립적이고 서비스의 상태 정보를 반영한 동적인 서비스 검색 방법을 제공할 수 있는 구조를 설계 및 구현한다.

3. WSUN: Web Services on Universal Networks

앞서 살펴본 것과 같이, 하나의 서브네트워크가 서로 간의 상호운용성을 지원해주지 못할 뿐만 아니라 다양한 물리 계층에 대한 이질적인 특성을 모두 반영할 수 없다. 따라서 다양한 서브네트워크가 존재하기 때문에 일관적인 서비스 검색 방법이나 이벤트 처리가 중요하

다. 또한, 사용가능한 서비스에 대한 지원과 서브네트워크에서 제공하는 서비스들의 조합이 필요하다. 이를 위해, 본 논문에서는 웹서비스 관점으로 접근하고 있으며 WSUN을 제안한다. WSUN은 사용자가 유니버설 서비스 브로커를 통해 서비스를 검색하고 사용할 수 있도록 한다. 그림 3은 SOA 기반의 환경으로 유니버설 서비스 브로커, 유니버설 서비스 레지스트리, 유니버설 어댑터(UniA: Universal Adaptor) 등으로 이루어져 있다. 이러한 구성 요소를 통해 WSUN은 사용자에게 각 서브네트워크의 서비스를 가상의 웹서비스(Virtual Web Services)로 변환한 유니버설 서비스(Universal Service)를 제공한다. 각 서브네트워크의 서비스를 가상의 웹서비스를 통해 제공하기 때문에 사용자는 어떠한 서비스라도 웹서비스를 사용하는 것과 같은 검색 방법 및 사용이 가능하다. WSUN의 각 구성 요소에 대한 상세 설명은 다음 장에서 하도록 하고, 이번 장에서는 시나리오를 통해 WSUN의 필요성을 설명한다.

3.1 유니버설 서비스 브로커가 존재하지 않은 경우

사용자는 회의를 진행하기 위해 회의 장소 내에 있는 서비스들을 이용할 수 있다. 예를 들어, 프린터, 프로젝터, LCD TV 등이 있다. 다음의 그림 4는 유니버설 서비스 브로커가 없는 경우로 사용자는 이러한 서비스들을 이용하기 위해서는 각 서브네트워크에서 지원하는 서비스의 종류와 통신 방법에 대해 모두 알고 있어야 한다. 즉, 모든 서브네트워크와 통신할 수 있는 환경을 갖추어야 하고, 질의 방법을 알아야 한다. 사용자는 각 서브네트워크에 맞는 질의와 통신 방법을 이용하여 서비스를 검색한다. 첫 번째로 사용자는 프린터를 이용하기 위해 질의를 한다. 프린터 서비스가 어떤 서브네트워크에 존재하는지 모르기 때문에 모든 서브네트워크마다 질의를 하게 되는데 각각의 서브네트워크의 질의에 대한 결과를 토대로 프린터 서비스를 이용할 수 있다. 하

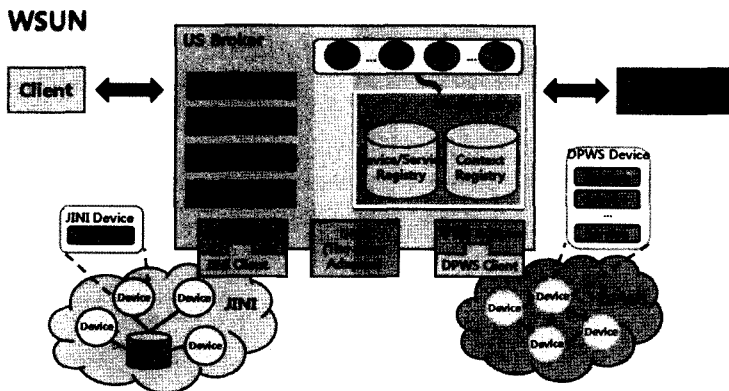


그림 3 Web Services on Universal Networks

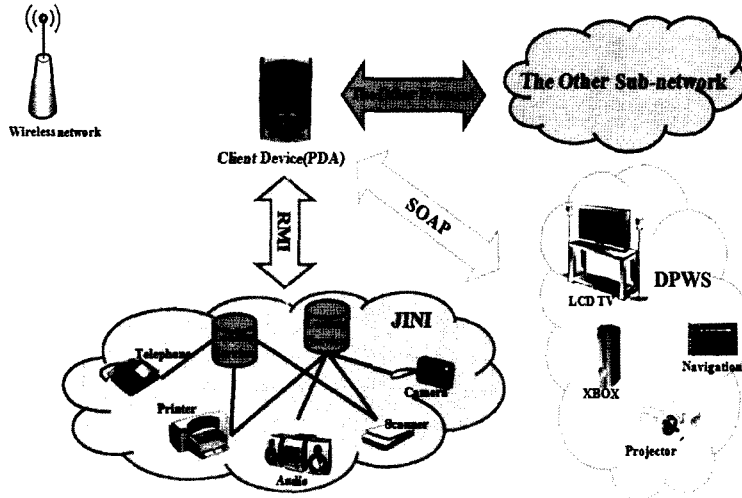


그림 4 유니버설 서비스 브로커가 존재하지 않는 경우

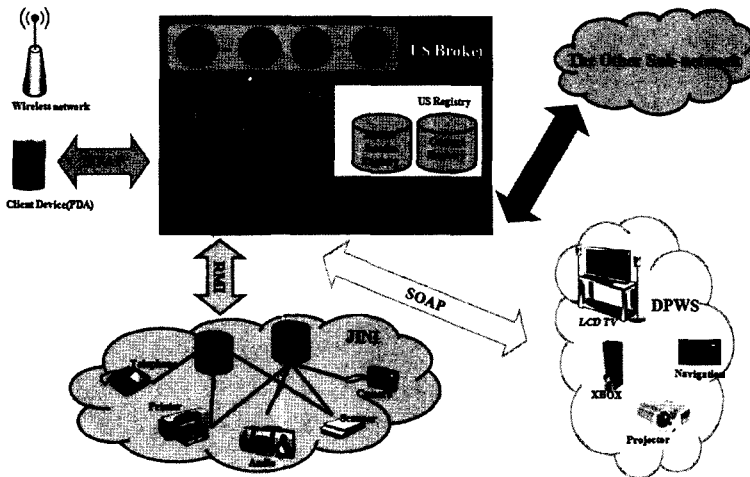


그림 5 유니버설 서비스 브로커가 존재하는 경우

지만 사용자가 필요한 서비스가 JINI에서 제공하는 프린터일 경우, JINI 서브네트워크의 환경을 갖추고 있지 않다면 서비스를 이용할 수 없다. 또한, 다른 사용자가 JINI의 프린트 서비스를 사용하고 있다면 서비스를 사용하기 위해 기다리거나 다른 프린터를 검색해야 한다. 즉, 서비스를 찾더라도 서비스의 상태 정보를 알 수 없기 때문에 다른 사용자가 서비스를 사용하고 있다면 서비스를 이용할 수 없고 다시 질의를 해야 하는 단점이 있다. 이런 문제와 함께 각각의 서브네트워크에 모두 질의를 하기 때문에 서비스 검색 시간이 증가하는 문제가 있다.

3.2 유니버설 서비스 브로커가 존재하는 경우

사용자는 회의를 진행하기 위해 회의 장소 내에 있는 서비스들을 이용할 수 있다. 예를 들어, 프린터, 프로젝

터, LCD TV 등이 있다. 다음의 그림 5는 유니버설 서비스 브로커가 존재하는 경우로 사용자는 서브네트워크마다 지원하는 서비스의 종류를 알고 있지 않아도 웹서비스를 검색하는 것과 같이 서비스를 검색할 수 있다. 이는 유니버설 서비스 브로커에서 각 서브네트워크의 서비스를 가상의 웹서비스로 변환하여 제공하기 때문에 모든 서비스가 웹서비스처럼 보이는 유니버설 서비스를 제공한다. 즉, 사용자는 유니버설 서비스 브로커를 인식하지 않고 웹서비스와 같이 각 서브네트워크의 서비스 검색 및 연결이 가능하다. 또한, 각 서브네트워크의 환경을 고려하지 않더라도 서비스 검색 및 연결이 가능하다. 유니버설 어댑터는 각각의 서브네트워크에서 사용하는 메시지 통신 방법을 유니버설 서비스 브로커에서 변환하여 전송하므로 사용자의 환경에 제약을 받지 않는

다. 유니버설 레지스트리는 디바이스 및 서비스의 메타 데이터 정보와 서비스의 상태 정보를 포함하여 사용가능한 동적인 서비스 검색을 지원한다. 예를 들어, 프로젝트 서비스를 질의를 할 경우, 사용가능한 서비스만 검색되기 때문에 연결이 가능하다. 이와 함께 사용자의 위치 정보 등과 같은 사용자 프로파일을 이용하여 서비스 검색이 이루어진다.

4. 유니버설 서비스 브로커

유니버설 서비스 브로커는 WSUN의 중심이 되는 구성 요소로 산재한 이질적인 서브네트워크의 상호운용성을 지원하는 SOA기반의 미들웨어이다. 유니버설 서비스 브로커는 서비스 사용자와 제공자 사이에 위치하여 서브네트워크의 서비스 및 WSUN 디바이스가 제공하는 서비스를 검색 및 사용할 수 있도록 한다. 이번 장에서는 서비스 검색, 등록 및 관리하는 과정에서 생기는 유니버설 서비스 브로커의 요구 사항과 범위를 제시하고, 그에 따른 기능과 동작에 필요한 프로토콜 스택을 설명한다. 또한, WSUN을 위한 유니버설 서비스 브로커의 구조와 구성 요소를 명세하여 사용자의 질의 방법을 통해 서비스 검색 및 연결 과정에 대해 살펴본다.

4.1 유니버설 서비스 브로커의 요구 사항 및 범위

유니버설 서비스 브로커는 서비스 발견, 등록 및 관리하는 과정에서 생기는 다음과 같은 요구 사항을 모두 해결할 수 있어야 한다.

- 사용자 등장: 사용자가 WSUN의 환경에서 서비스를 검색하거나 사용할 수 있도록 하기 위해 유니버설 서비스 브로커와 통신을 지원해야 한다. 즉, 유니버설 서비스 브로커는 서비스를 검색하는 방법을 사용자에게 제공해야 한다. 이 과정에서 사용자는 유니버설 서비스 브로커를 사용하기 위해 별도의 환경을 구축하지 않아도 된다. 그림 6은 사용자가 등장하여 유니버설 서비스 브로커를 발견하고, 유니버설 서비스 브로커와 통신하는 과정을 나타낸다.

- 서브네트워크의 등장 및 퇴장: JINI나 UPnP와 같은 서브네트워크가 WSUN에 등장하거나 퇴장하였을 경우 디바이스 및 서비스 정보를 유니버설 서비스 브로커에 반영해야 한다. 유니버설 서비스 브로커의 유니버설 서비스 레지스트리에 서브네트워크의 디바이스 및 서비스 메타정보를 저장하여 동적인 서비스 검색이 가능하도록 한다. 단, DPWS와 같은 P2P를 지원하는 서브네트워크는 서브네트워크 단위로 WSUN에 등장하거나 퇴장하는 경우는 없다.

- 서브네트워크의 디바이스 등장 및 퇴장: 서브네트워크의 디바이스의 위치는 유동적이다. 디바이스의 등장 및 퇴장에 따라 디바이스에서 제공하는 서비스의 상태가 종속적이기 때문에 이에 대한 상황을 감지하고 적절한 해결 방안을 제시할 수 있어야 한다. 즉, 유니버설 서비스 브로커는 사용자에게 사용 가능한 서비스만 검색할 수 있도록 제공할 수 있어야 한다.

- WSUN 디바이스 등장 및 퇴장: WSUN 디바이스는 WSUN의 환경을 지원하는 디바이스로서, 유니버설 서비스 브로커가 WSUN 디바이스의 등장 및 퇴장에 대한 지원이 가능해야 한다.

- 사용자의 서비스 검색 및 연결: 사용자가 서비스 검색 질의를 유니버설 서비스 브로커에 요청하면 그에 대한 검색 결과를 알려주고, 검색한 서비스 중 사용자가 원하는 서비스를 사용할 수 있도록 연결해야 한다.

4.2 유니버설 서비스 브로커의 기능

유니버설 서비스 브로커는 앞서 살펴본 것과 같이 다양한 요구 사항을 만족해야 한다. 이를 위해, 유니버설 서비스 브로커는 다음과 같은 기능들을 담당한다.

- 서비스 검색: 유니버설 서비스 브로커의 레지스트리에 서브네트워크의 디바이스 및 서비스 메타 정보를 저장하여 동적인 서비스 검색이 이루어지도록 한다. 이를 위해, JINI, UPnP, DPWS 등 각 서브네트워크에 존재하는 디바이스 및 서비스를 검색하는 기능을 담당한다.

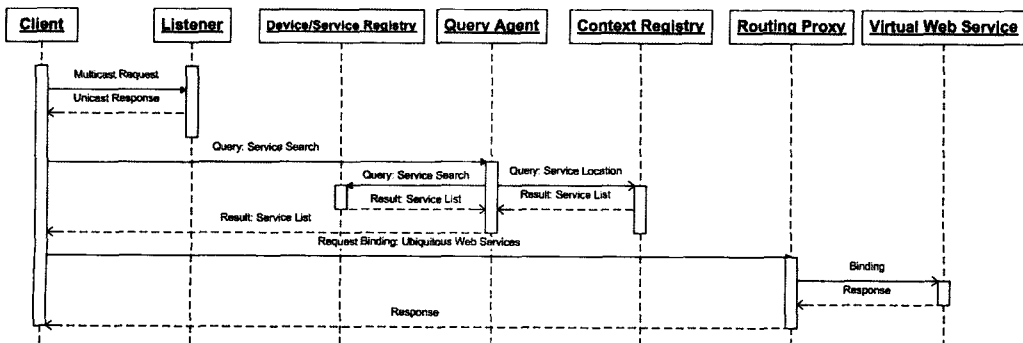


그림 6 사용자 등장에 따른 유니버설 서비스 브로커 동작 과정

- 서비스 등록 및 관리: 각 서브네트워크의 디바이스 및 서비스 정보를 검색한 후 유니버설 서비스 레지스트리의 디바이스/서비스 레지스트리와 컨텍스트 레지스트리에 등록하고 서비스의 사용 가능 여부에 따라 관리를 한다.
- 서비스 연결: 사용자의 서비스 검색 후 원하는 서비스를 사용할 수 있도록 연결을 담당한다. 사용자는 가상의 웹서비스를 사용하는 것으로 인식하기 때문에 실제로 사용하게 될 서브네트워크의 서비스를 사용하도록 지원한다.
- 가상의 웹서비스 생성: 서브네트워크의 서비스는 유니버설 서비스 브로커를 통해 가상의 웹서비스로 사용자에게 제공된다. 사용자는 웹서비스를 사용하는 것으로만 인식하도록 하여 사용자 및 서브네트워크의 환경에 제약을 받지 않고 서비스를 사용할 수 있다.
- 서비스의 이벤트 처리: 동적인 서비스 검색 및 사용을 위해 사용 가능한 서비스만 제공하도록 해야 한다. 디바이스의 위치나 사용자의 서비스 사용이 종료될 경우 서비스의 상태에 변화가 생기므로 이에 따른 정보를 반영한다.

4.3 유니버설 서비스 브로커 프로토콜 스택

유니버설 서비스 브로커는 서브네트워크의 서비스를 가상의 웹서비스로 변환하여 사용자에게 제공하기 때문에 유니버설 서비스 브로커에 적절한 웹서비스 프로토콜 스택을 따른다. 그림 7은 유니버설 서비스 브로커의 프로토콜 스택으로 USDP[12], WS-Eventing[13], WS-Addressing[14], SOAP(Simple Object Access Protocol) [15], WSDL(Web Services Description Language) [16], XML Schema[17]로 구성된다.

USDP	WS-Eventing
WS-Addressing	
SOAP	
WSDL, XML Schema	

그림 7 유니버설 서비스 브로커의 프로토콜 스택

- WSDL: 서비스의 인터페이스와 연결을 위한 전송 프로토콜의 명세 정보를 포함한다.
- XML Schema: 서비스를 사용하기 위한 송수신 메시지의 데이터 형식을 정의한다.
- SOAP: WSDL에 정의된 메시지 형식의 전송 프로토콜로 웹서비스를 위한 다양한 웹서비스 프로토콜을 통합할 수 있도록 한다. 현재는 SOAP을 이용하지만 유비쿼터스의 경량화된 환경에 맞도록 SOAP을 변형하여 simple SOAP인 gSOAP [18]으로 확장한다.

- WS-Addressing: WS-Addressing은 SOAP과 밀접한 관련을 가지고 있다. 송수신 되는 모든 메시지의 SOAP의 헤더 부분에 주소 정보를 포함하고 있다. 이를 통해, HTTP, SMTP, TCP와 UDP 등 어떤 전송 프로토콜이라도 메시지를 전송할 수 있다.
- USDP: 서비스를 검색하기 위한 프로토콜로 플러그 앤 플레이 및 네트워크에 연결된 ad-hoc 환경에서의 검색을 제공한다. WSUN 디바이스의 등장 및 퇴장에 대한 처리와 함께 유니버설 서비스 브로커를 통한 유니버설 서비스 검색이 가능하도록 한다.
- WS-Eventing: 서비스의 위치나 상태 정보 등의 유동성에서 생기는 이벤트를 처리하기 위한 프로토콜이다. 예를 들어, 사용자가 서비스의 사용이 종료되었을 경우도 서비스의 상태 정보가 변하게 되는 것이므로 WS-Eventing을 통해 유니버설 서비스 브로커에 전달하게 된다.

4.4 유니버설 서비스 브로커 구조

앞서 살펴본 것과 같이, 유니버설 서비스 브로커는 사용자에게 유니버설 서비스의 검색을 지원하기 위한 그림 8과 같은 구조를 가진다. 유니버설 서비스 브로커는 리스너, 질의 에이전트, 등록 에이전트, 라우팅 프록시, 유니버설 서비스 레지스트리, 유니버설 어댑터로 구성된다.

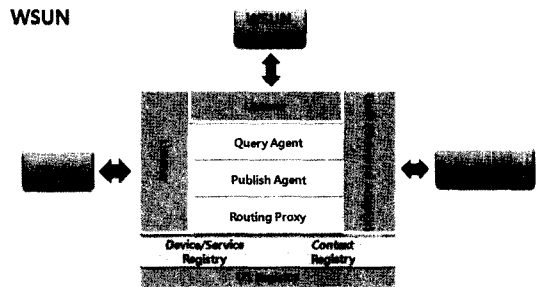


그림 8 유니버설 서비스 브로커 구조

- 리스너(Listener): WSUN 디바이스와 사용자가 WSUN에 등장하였을 때, 유니버설 서비스 브로커와 통신을 하기 위한 유니버설 서비스 브로커의 기본 정보를 제공한다. 예를 들어, 유니버설 서비스 브로커의 주소, 호스트 등의 정보가 되며 WSUN 디바이스의 등록이나 사용자가 서비스를 검색, 발견할 수 있도록 환경을 만들어 주어야 한다.
- 질의 에이전트(Query Agent): 사용자의 서비스 검색 질의를 처리한다. 질의 에이전트는 사용자의 서비스 검색 질의를 받게 되면 유니버설 서비스 레지스트리의 디바이스/서비스 레지스트리의 메타데이터와 컨텍스트 레지스트리의 서비스의 위치 정보를 검색하여

사용자에게 알려준다.

- 등록 에이전트(Publish Agent): WSUN의 환경에 필요한 메타데이터 정보를 유니버설 서비스 레지스트리에 등록한다. WSUN 디바이스나 서브네트워크에 존재하는 메타데이터 정보가 유니버설 어댑터에 의해 수집되면 유니버설 서비스 레지스트리의 디바이스/서비스 레지스트리와 컨텍스트 레지스트리에 각각 저장하는 기능을 한다.
- 라우팅 프록시(Routing Proxy): 사용자가 서비스 검색 결과에서 원하는 서비스를 연결을 유니버설 서비스 브로커에 요청할 경우 라우팅 프록시는 가상의 웹서비스의 실제 서비스인 서브네트워크 서비스를 연결시켜주는 기능을 담당한다.
- 유니버설 서비스 레지스트리(US Registry): 유니버설 서비스 레지스트리는 디바이스/서비스 레지스트리와 컨텍스트 레지스트리로 구성된다.
 - 디바이스/서비스 레지스트리(Device/Service Registry): WSUN의 환경에서 사용되는 메타데이터 즉, 각 서브네트워크의 디바이스 및 서비스의 명세 정보와 WSUN 디바이스의 명세 정보를 저장한다. 이를 통해, 유니버설 서비스 브로커는 사용자의 서비스 검색 질의가 오면 서브네트워크에 존재하는 디바이스 및 서비스를 검색하지 않아도 서비스 검색이 가능할 뿐만 아니라 서비스 상태 정보도 포함하고 있기 때문에 동적인 서비스 검색이 가능하다. 디바이스/서비스 레지스트리에는 서브네트워크의 디바이스와 서비스 정보 중 공통되고, 정적이며, 필수적인 요소만 저장한다. 이러한 요소만을 추출하기 위해서 JINI, UPnP, DPWS로 범위를 한정하여 디바이스 및 서비스 정보를 표 3과 같이 추출하였다.

이와 같은 요소를 통해, 서비스 검색 시의 효율적인 특성을 나타낸다. 유니버설 서비스 레지스트리가 존재하지 않으면 서비스의 검색 질의가 오면 모든 서브네트워

크나 WSUN 디바이스에 검색을 하는 중개자(Mediator) 방식을 이용해야 한다. 이는 각 서브네트워크와 WSUN 디바이스가 제공하는 서비스를 관리하는 모듈이 필요 없기 때문에 동적인 서비스 검색이 가능한 장점이 있지만, 서비스의 상태 정보를 포함할 수 없으며 검색 시간이 증가하는 문제점이 있다. 반면에 서브네트워크나 WSUN 디바이스의 모든 메타데이터 정보를 레지스트리에 저장하고 관리하는 데이터 웨어하우스(Data warehouse) 방식이 있다. 중개자 방식에서 생기는 서비스 검색 시간의 증가를 해결할 수 있지만 모든 메타데이터 정보를 레지스트리에 등록, 삭제 또는 수정 과정에 대한 부담이 가중된다. 그리하여 유니버설 서비스 레지스트리의 디바이스/서비스 레지스트리는 각 서브네트워크의 디바이스 및 서비스의 명세 정보 중 공통적이고 정적이며 필수적인 요소만을 추출하여 관리하는 하이브리드(Hybrid) 방식을 취한다.

- 컨텍스트 레지스트리(Context Registry): 유니버설 서비스 환경에서는 디바이스의 위치가 자주 변하기 때문에 컨텍스트 레지스트리는 디바이스의 위치 정보를 저장하여 사용자의 위치 정보에 맞는 서비스를 검색하도록 상황 정보로 쓰인다. 예를 들어, 회의 장소 내에서 사용 가능한 프린터를 검색하는 것을 들 수 있다. 이는 추후 사용자의 프로파일 등의 온톨로지 검색으로 확장할 것이다.
- 유니버설 어댑터[19]: 서브네트워크와 유니버설 서비스 브로커 사이의 중계 역할을 담당한다. 이질적인 서브네트워크와 유니버설 서비스 브로커가 통신을 하기 위한 브리지 역할로 서브네트워크의 서비스를 가상의 웹서비스로 변환하여 유니버설 서비스를 검색을 지원한다. 그림 9는 유니버설 어댑터의 구성 요소로 다음과 같은 기능을 한다.

표 3 디바이스/서비스 레지스트리 메타데이터 정보

명세 정보	설명
Sub-networkType	유니버설 서비스 브로커에서 다수의 서브네트워크를 관리하기 위한 타입
sServiceID	서브네트워크에서의 서비스 ID
uServiceID	유니버설 서비스 브로커에서의 서비스 ID
uDeviceID	유니버설 서비스 브로커에서 관리하기 위한 각 서브네트워크의 디바이스 ID
serviceName	서비스 이름
deviceName	디바이스 이름
categoryName	서비스의 카테고리 정보
serviceStatus	서비스의 상태 정보
deviceDescription	디바이스의 명세 정보
serviceDescription	서비스의 명세 정보

Universal Adaptor (UniA)

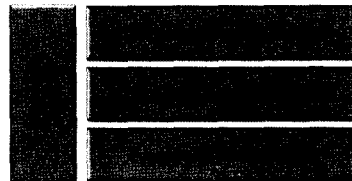


그림 9 유니버설 어댑터 구조

- 메시지 전송(Message Transfer): 서브네트워크와 유니버설 서비스 브로커에 필요한 정보들은 전송하는 기능을 담당한다. 이는 유니버설 서비스 브로커와 서브네트워크의 통신 프로토콜 등이 이질적이기 때문에 유니버설 어댑터를 통해 중재된다.

- **서브네트워크의 서비스 검색(Sub-network Services Discovery):** 서브네트워크의 서비스를 가상의 웹서비스로 변환하고 유니버설 서비스 레지스트리에 저장할 정보들을 검색한다. 검색한 정보들은 가상의 웹서비스 생성 후 등록 에이전트로 보내 유니버설 서비스 레지스트리에 저장한다.

- **가상의 웹서비스 생성(Virtual Web Services Creation):** 사용자는 서브네트워크의 존재를 인식하지 않고 서비스를 검색하고 사용하기 때문에 서브네트워크의 서비스를 웹서비스 형태로 변환한다. 변환된 서비스가 외부에 가상의 웹서비스로 사용자에게 노출된다. 이를 통해, 사용자는 웹서비스를 사용하는 것으로 보이지만 실제로 유니버설 어댑터를 통해 서브네트워크의 서비스가 제공된다.

- **이벤트와 알림(Events and Notifications):** 서브네트워크 및 서브네트워크의 디바이스의 등장과 퇴장이 발생하였을 경우 이를 반영하여 플러그 앤 플레이를 지원한다. 또한, 사용자가 서비스의 사용을 종료하였을 경우, 서비스의 상태 정보를 변경하여 서비스가 사용 가능하도록 한다.

이러한 기능들은 설계 단계와 실행 단계로 구분되어 다음과 같은 동작을 나타낸다.

- **설계 단계 - Hook-up 과정**

그림 10과 같이 각 서브네트워크마다 존재하는 유니버설 어댑터는 네트워크에 연결이 되면 존재하고 있는 서브네트워크의 디바이스 및 서비스 정보를 검색한다. 검색이 완료된 후 서브네트워크의 서비스를 가상의 웹서비스로 변환하여 등록 에이전트를 통해 디바이스/서비스 레지스트리와 컨텍스트 레지스트리에 각각 저장한다. 이러한 일련의 과정이 종료되면 이 단계의 동작이 완료

된다.

- **실행 단계 - 서브네트워크의 디바이스가 등장할 경우**
그림 11은 서브네트워크의 디바이스가 등장함에 따라 등장 메시지를 유니버설 어댑터에게 보내면 유니버설 어댑터는 디바이스와 서비스의 메타데이터를 요청한다. 요청한 정보를 이용하여 질의 에이전트를 통해 유니버설 서비스 레지스트리에 기존에 존재했던 디바이스인지 새로운 디바이스인지 판단하여 새로운 디바이스인 경우는 설계 단계와 같은 동작을 한다. 반면에 기존에 존재했던 디바이스의 재등장인 경우는 서비스의 상태 정보와 위치 정보만을 변경하여 서비스의 검색 및 사용이 가능하도록 한다.

- **실행 단계 - 서브네트워크의 디바이스가 퇴장할 경우**
서브네트워크의 디바이스가 퇴장하면서 퇴장 메시지를 보내게 된다. 이를 감지하여 해당하는 유니버설 어댑터는 퇴장하는 디바이스의 서비스 상태 정보를 등록 에이전트를 통해 그림 12와 같이 사용 불가능도록 변경한다.

- **실행 단계 - 서브네트워크의 디바이스가 퇴장할 경우 (Error)**

서브네트워크의 디바이스가 퇴장하는 경우는 퇴장 메시지를 보내는 경우와 시스템의 결함으로 인한 경우가 있다. 그림 13은 후자의 경우로 서비스에 일정한 주기로 응답 메시지를 요청한다. 일정 시간동안 응답 메시지가 오지 않으면 디바이스가 사라진 것으로 간주하여 서비스의 상태 정보를 변경하게 된다.

- **실행 단계 - 서브네트워크의 서비스의 상태 정보가 변경된 경우**

그림 14와 같이 사용자가 서비스의 사용을 종료하면 서비스는 이벤트를 통해 서비스의 상태 정보가 변경되었음을 알리고 디바이스/서비스 레지스트리에 반영한다.

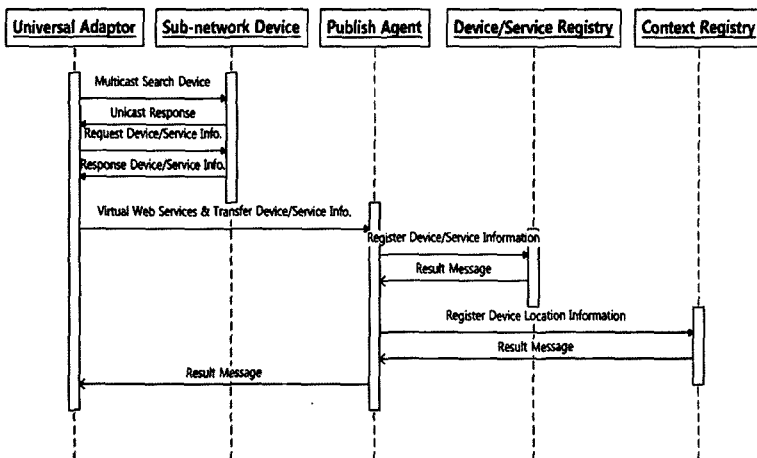


그림 10 Hook-up 과정

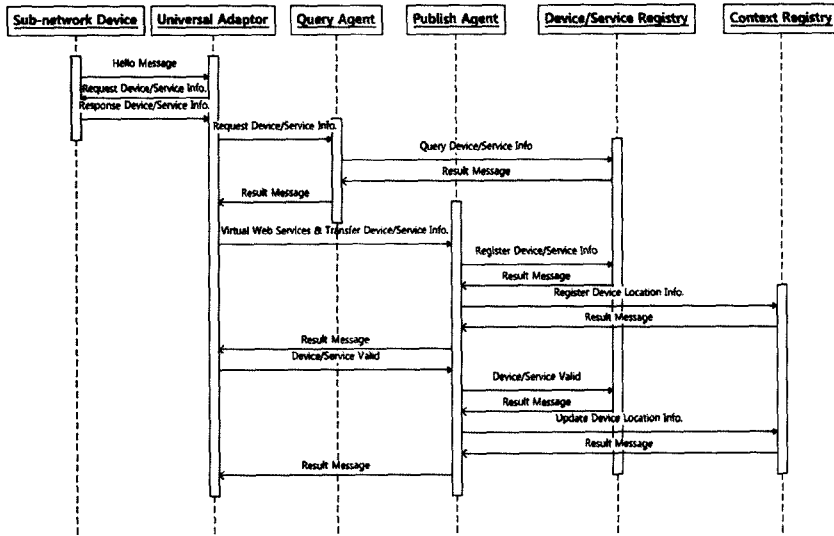


그림 11 서브네트워크의 디바이스가 등장할 경우

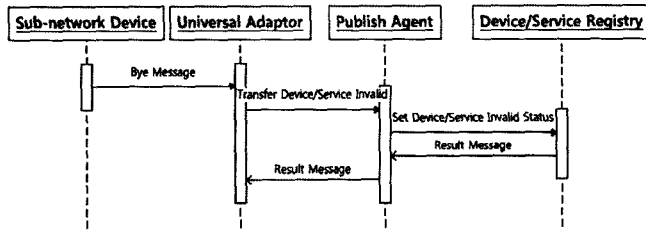


그림 12 서브네트워크의 디바이스가 퇴장할 경우

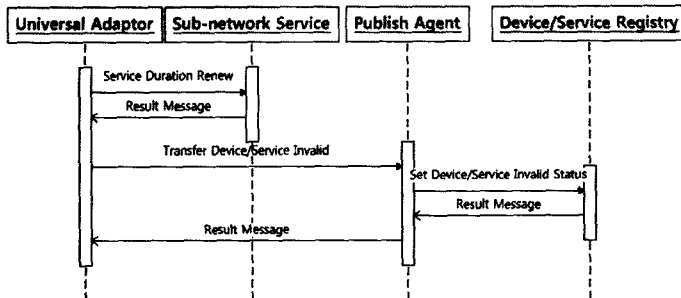


그림 13 서브네트워크의 디바이스가 퇴장할 경우 (Error)

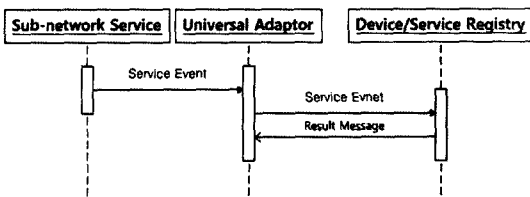


그림 14 서브네트워크의 서비스의 상태 정보가 변경된 경우

유니버설 어댑터의 기능 및 동작 과정을 살펴보았다. 위에서 설명한 유니버설 어댑터는 공통적인 특성을 추

출하여 설명하였다. 본 연구에서는 JINI와 DPWS만을 고려하여 구현 중으로 JINI 어댑터와 DPWS 어댑터로만 구성이 되어 있다. 각 서브네트워크의 어댑터가 동작하는 과정은 서브네트워크의 특성을 포함한다.

4.5 서비스 검색 및 연결

이번 절에서는 그림 15와 같이 WSUN의 환경에서 유니버설 서비스를 검색 및 연결하기 위한 과정을 살펴본다.

사용자는 유비쿼터스 환경에서 서비스를 위해서는 일단 유니버설 서비스 브로커를 찾아야 한다. 멀티캐스트

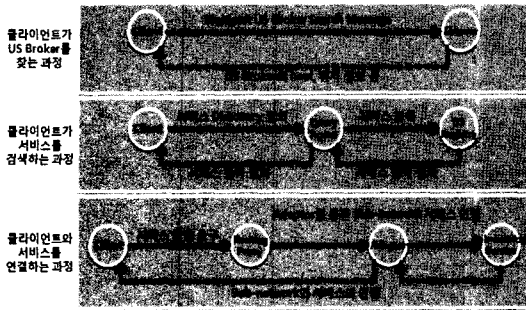


그림 15 사용자의 서비스 검색 및 연결 과정

메시지로 유니버설 서비스 브로커의 리스너를 통해 유니버설 서비스 브로커의 주소 및 호스트 등의 정보를 얻는다. 유니버설 서비스 브로커와 통신이 이루어지면 사용자는 서비스 검색이 가능해진다. 또한, 사용자의 위치 정보와 서비스의 상태 정보는 자동적으로 사용자 질의에 포함되어 서비스 검색이 이루어진다. 사용자의 위치 정보는 컨텍스트 레지스트리의 디바이스의 위치 정보와 비교하며 서비스를 사용 가능한 것만 검색을 하여 사용자에게 최적의 검색 결과를 제공한다.

이러한 일련의 과정으로 사용자의 서비스 검색 질의가 질의 에이전트로 보내지면 질의에 따라 디바이스/서비스 레지스트리와 컨텍스트 레지스트리를 각각 검색한다. 사용자는 검색 결과에 대해 메시지를 받으면 원하는 서비스에 대한 연결을 유니버설 서비스 브로커에 요청한다. 사용자는 모든 서비스가 웹서비스로 인식하기 때문에 실제 서비스를 제공하는 서브네트워크의 서비스를 연결해야 한다. 이 과정에서 라우팅 프록시는 해당하는 서브네트워크 서비스의 유니버설 어댑터와 사용자를 연결을 담당한다. 실제로 사용자는 웹서비스를 사용하는 것으로 보이도록 유니버설 어댑터가 중재를 한다. 이렇게 하여 유니버설 서비스 브로커는 사용자에게 산재한 이질적인 서브네트워크의 서비스 검색 및 연결이 가능하도록 상호운용성을 지원한다.

5. WSUN 구현

5.1 구현 환경

본 논문에서는 WSUN을 위한 서브네트워크로 JINI와 DPWS를 사용하였다. JINI, UPnP와 DPWS는 이질적인 특성을 가장 잘 반영한 서브네트워크이다. DPWS는 UPnP의 차세대 버전으로 디바이스간의 통합을 위해 디바이스 관점의 웹서비스를 제공한다. 따라서 WSUN과 서브네트워크인 JINI, DPWS의 구현 환경은 다음과 같다.

- WSUN

- OS: Windows Server 2003 R2 Enterprise SP2
- Java: J2SDK 1.4.2
- MySQL: MySQL 4.1
- Tomcat: Tomcat 5.0.30
- Axis2: Axis2 1.3
- JINI
 - OS: Windows XP Professional SP2
 - Java: JDK 1.5
 - 개발 툴: Eclipse 3.3
 - 룩업서비스 (Lookup Service): IncaX5.0, JINI2.1
- DPWS
 - OS: Windows Vista Enterprise K Edition
 - 개발 툴: Microsoft Visual Studio 2008
 - .NET Framework 3.0
 - SDK: Microsoft® Windows® Software Development Kit(SDK) for Windows Server code name "Longhorn" and .NET Framework 3.5

5.2 구현

4장에서 WSUN의 설계에 따라 구현된 부분을 테스트하기 위한 시나리오는 다음과 같다.

1. 사용자는 리스너를 통해 유니버설 서비스 브로커를 검색한다.
2. 리스너는 사용자가 사용할 수 있는 유니버설 서비스 브로커의 주소와 포트 정보를 전송한다.
3. 사용자는 회의 자료를 출력하기 위해 흑백 프린터를 검색한다.
4. 사용자가 원하는 흑백 프린터가 존재하지 않아 외부의 프린터 서비스가 새로 등록된다.
5. 다시 사용자는 흑백 프린터를 검색한다.
6. 사용자는 회의 자료를 출력한 후 회의 진행을 위한 프로젝트 서비스를 검색하여 사용한다.
7. 회의가 종료되면 회의에서 찍은 사진을 컬러 프린터 서비스를 통해 출력한다.

위의 시나리오 중 과정 3은 질의 에이전트를 통해 서비스를 검색하고, 과정 4가 실행 단계에서 서비스의 종류에 상관없이 새로 생겨나면 등록 에이전트와 유니버설 어댑터를 이용하여 WSUN에서 부가의 작업 없이 바로 사용할 수 있도록 한다. 이때 새로 생긴 디바이스와 서비스의 메타데이터 정보를 유니버설 서비스 레지스트리에 저장하여 관리한다. WSUN의 유니버설 서비스 브로커로 인해 사용자는 어떤 서브네트워크에서 서비스를 제공하는지 인식하지 않고도 서비스를 이용하게 된다. 이는 각 서브네트워크의 질의 방법과 통신 프로토콜을 고려하지 않고도 유니버설 서비스 브로커를 통해 사용가능한 서비스만 제공한다.

그림 16은 사용자 입력 화면으로 서비스 이름을 필수

Universal Service Discovery Protocol

속성명	값	서비스 카테고리
서비스 이름	PrinterService	Printer <input type="checkbox"/> Projector <input type="checkbox"/>
서비스 설명정보	Color Printer Service	

[검색 결과]
 아래의 서비스 검색 결과 중 원하는 서비스를 선택하십시오.
 ■ 검색 조건
 - 서비스 이름 : PrinterService
 - 서비스 설명정보 : Color Print Service
 - 서비스 카테고리 : Print

1	91a36c73-bdee-4c39-9ee3-6be36ue739ek	PrinterService	168.188.46.37#91a36c73-bdee-4c39-9ee3-6be36ue739ek
---	--	----------------	--

그림 16 사용자 검색 인터페이스

적인 요소로 하여 서비스의 카테고리 및 서비스의 상세 명세를 부가적인 요소로 서비스 검색을 할 수 있다. 유니버설 서비스 브로커를 통해 검색된 결과 목록 중 사용자는 원하는 서비스를 선택하여 사용한다.

6. 결론 및 향후 연구

유비쿼터스 웹서비스는 어떠한 단말/네트워크 환경에서도 다양한 응용 서비스를 연계하여 이용할 수 있도록 하는 미래형 웹서비스 기술로 정의될 수 있다. 또한, 계속 유동적으로 변하는 디바이스에 대한 동적인 서비스의 검색과 이용이 요구되고, ad-hoc한 환경에서의 검색 방법 등을 지원해야 한다. 따라서 산재한 서브네트워크의 이질성 문제는 반드시 해결해야 한다.

본 논문에서는 이기종 서브네트워크 환경과 다양한 컴퓨팅 환경이 융합된 환경에서 웹서비스를 연계·이용하여 유니버설 서비스를 제공하기 위하여 WSUN(Web Services on Universal Networks)를 제안하였다. 이를 위해 유니버설 서비스 브로커의 요구 사항 및 범위를 정의하여 USDP를 정립함으로써 유니버설 서비스 브로커의 검색 프로토콜을 제공한다. 유니버설 서비스 브로커의 구성 요소인 리스너, 등록 에이전트, 질의 에이전트, 라우팅 프록시, 유니버설 어댑터와 유니버설 서비스 레지스트리를 설계하였다.

이를 통해, 관련 연구에서 살펴본 [11]이나 DomoNet와는 달리 디바이스/서비스의 등장 및 퇴장에 따른 이벤트 메시지로 사용가능한 서비스만 검색할 수 있도록 지원한다. 디바이스 관점에서의 상호운용성과 함께 디바이스에서 제공하는 여러 서비스에 대한 접근도 가능하다. 이질적인 서브네트워크의 상호운용성을 위해 웹서비스를 통해 해결 방법을 찾기 때문에 일반 웹서비스와 유니버설 네트워크에서의 서비스 즉, 가상의 웹서비스를

지원할 수 있다.

본 연구는 추후 UPnP, HAVi 등의 서브네트워크로 확장하여 유니버설 서비스를 제공할 필요가 있으며, 각 서브네트워크에서 사용하고 있는 서비스/디바이스의 카테고리에 대한 연관 관계를 정립함으로써 다양한 방법으로 서비스 검색을 할 수 있도록 제공해야 할 것이다. 추후 유니버설 레지스트리의 구성을 UDDI에 적용하여 웹서비스와 가상의 웹서비스인 유니버설 서비스를 조합하여 제공하는 것을 기대할 수 있다. 현재는 컨텍스트 정보를 서비스의 위치 정보로만 사용하고 있으나 사용자의 프로파일 등을 이용하여 서비스 조합 및 대체 서비스를 제공하기 위한 연구가 필요하다. 또한, 서브네트워크의 이벤트와 에러에 대해 감지하여 처리할 수 있는 부분을 향후 연구되어야 할 것이며 트랜잭션, 보안, QoS, 시맨틱과 웹서비스 조합 등의 연구로 유비쿼터스 웹서비스에 최적의 프레임워크를 제공할 것으로 기대된다.

참고 문헌

- [1] 이규철, 이호경, "유비쿼터스 환경의 서비스 융·복합 아키텍처", 한국정보과학회, 정보과학회지 제25권 제1호, pp. 17-26, 2007. 2.
- [2] Malcolm Attard, "Ubiquitous Web Services," <http://www.cs.um.edu.mt/~csaw/CSAW03/Proceedings/UbiquWebServices.pdf> COMPUTER SCIENCE ANNUAL RESEARCH WORKSHOP, 2003.
- [3] Sun Microsystems. JINI Specifications Archive v2.1, 2005.
- [4] UPnP Forum. UPnP Device Architecture v1.0.1, 2 December 2003.
- [5] Havi Consortium. HAVi Specification v1.1, 15 May 2001.
- [6] Shannon Chan et. Al, "Devices Profile for Web Services," February 2006.
- [7] Hendrik Bohn, Andreas Bobek, Frank Golasowski,

"SIRENA - Service Infrastructure for Real-time Embedded Networked Devices: A service oriented framework for different domains," April 2006.

- [8] OSGi Alliance. About the OSGi Service Platform, Technical Whitepaper Revision 4.1, 7 June 2007.
- [9] Vittorio Miori, Luca Tarrini, Maurizio Manca, "An Open Standard Solution for Domotic Interoperability," IEEE Transactions on Consumer Electronics, vol. 52, NUMB 1, pp. 97-103, February 2006.
- [10] Vittorio Miori, Luca Tarrini, Maurizio Manca, "DomoNet: A Framework and a Prototype for Interoperability of Domotic Middlewares Based on XML and Web Services," ICCE 2006 on Consumer Electronics, pp. 117-118, January 2006.
- [11] Michael Ditzte, Guido Kamper, Isabell Jahnich, Reinhard Bernhardt-Grisson, "Service-based access to distributed embedded devices through the open service gateway," INDIN 2004, pp. 493-498, June 2004.
- [12] Yun-Young Hwang Il-Jin Oh, Hyung-Jun Im, Kangchan Lee, Seungyun Lee, Kyu-Chul Lee, "Universal Service Discovery Protocol," ICCIT 2007, November 2007.
- [13] Don Box, Microsoft et. Al, "Web Services Eventing," March 2006.
- [14] Don Box, Microsoft et. Al, "Web Services Addressing," August 2004.
- [15] Don Box, Microsoft et. Al, "Simple Object Access Protocol 1.1," May 2000.
- [16] Erik Christensen, Microsoft et. Al, "Web Services Description Language 1.1," March 2001.
- [17] XML Schema Working Group, "XML Schema," October 2004.
- [18] Robert van Engelen, "gSOAP 2.7.9 User Guide," September 2007.
- [19] Il-Jin Oh, Hyung-Jun Yim, Yun-Young Hwang, Kangchan Lee, Seungyun Lee, Kyu-Chul Lee, "Design of the Adaptor for interoperability between sub-networks in US Broker," NEXT 2007, pp. 335-338, October 2007.



임형준

2003년~2007년 삼성 SDS IT 우수동아리. 2003년~현재 한국 Microsoft Student Club. 2007년 충남대학교 전기정보통신공학부 컴퓨터전공 학사. 2007년~현재 충남대학교 대학원 컴퓨터공학 석사과정. 관심분야는 차세대 웹, 웹 2.0, 웹서비스, 유비쿼터스 웹서비스(UWS)



오일진

2006년 공주대학교 컴퓨터공학과 학사
2006년~현재 충남대학교 대학원 컴퓨터공학과 석사과정. 관심분야는 XML, 웹서비스, 유비쿼터스 웹서비스(UWS)



황윤영

2002년 충남대학교 컴퓨터공학과 학사
2004년 충남대학교 대학원 컴퓨터공학과 석사. 2004~현재 충남대학교 대학원 컴퓨터공학과 박사과정. 관심분야는 서비스 지향 아키텍처, 유비쿼터스 웹서비스(UWS)



이경하

1998년 충남대학교 정보통신공학 학사
2000년 충남대학교 대학원 정보통신공학 석사. 2006년 충남대학교 대학원 컴퓨터공학 박사. 2006년 7월~12월 한국전자통신연구원 디지털융연구단 Post-doc. 2007년 1월~현재 충남대학교 소프트웨어연구소 전임연구원. 관심분야는 데이터베이스, 정보 검색, XML, 웹서비스



이강찬

2001년~현재 ETRI 표준연구센터, 선임연구원. 2002년~현재 W3C 대한민국 사무국 부국장. 2002년~현재 TTA 국제표준전문가. 2004년~현재 TTA 웹프로젝트 그룹(PG401) 부의장. 2006년~현재 ITU-T SG13 Editor. 현재 한국전자통신연구원 표준연구센터 서비스융합표준연구팀 선임연구원. 관심분야는 차세대 웹, 디바이스 코디네이션, 유비쿼터스 웹, 웹2.0, 모바일 웹



이승운

1999년~현재 ETRI 표준연구센터, 선임연구원. 2003년~현재 ETRI 표준연구센터 서비스융합표준연구팀장. 2004년~현재 TTA 국제표준전문가. 2006년~현재 TTA IT융용기술위원회(TC04) 부의장 2006년~현재 TTA 웹프로젝트 그룹(PG401) 의장. 2006년~현재 ITU-T SG13 Editor. 현재 한국전자통신연구원 표준연구센터 서비스융합표준연구팀장. 관심분야는 차세대 웹 표준, 유비쿼터스 웹서비스(UWS) 표준, 모바일 웹 표준, 웹 2.0 표준

이 규 철

1984년 서울대학교 컴퓨터공학 학사. 1986년 서울대학교 컴퓨터공학 석사. 1990년 서울대학교 컴퓨터공학 박사. 1989년~현재 충남대학교 컴퓨터공학과 교수. 1989년 3월~1994년 6월 미국 IBM Almaden Research Center 초빙연구원. 1995년 8월~1996년 8월 미국 Syracuse University, CASE Center 초빙 교수. 1997년 1월~1998년 1월 교육부 학술진흥재단 부설 첨단학술센터 파견 교수. 1997년 7월~현재 한국정보과학회 논문편집위원. 2000년 2월~2004년 2월 산업자원부 한국 ebXML 전문위원회 위원장. 2003년 3월~현재 한국전자거래학회 이사. 2003년 3월~현재 조달청 목록자문위원. 2003년 8월~현재 웹 코리아포럼 부위원장. 2005년 1월~현재 한국기록관리학회 이사. 2006년 2월~현재 충남대학교 소프트웨어연구소 소장. 2006년 8월~현재 대통령자문 정부혁신 지방분권위원회 위원. 관심분야는 데이터베이스, XML, 정보 통합, 멀티미디어 시스템, e-비즈니스 시스템, 유비쿼터스 웹서비스(UWS)