



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2010-0043849  
(43) 공개일자 2010년04월29일

(51) Int. Cl.

H04L 12/24 (2006.01) H04L 12/28 (2006.01)

(21) 출원번호 10-2008-0103070

(22) 출원일자 2008년10월21일

심사청구일자 2008년10월21일

(71) 출원인

충남대학교산학협력단

대전광역시 유성구 공동 220번지 충남대학교

서울대학교산학협력단

서울 관악구 신림동 산 56-1

(72) 발명자

이규철

대전 서구 삼천동 가람아파트 6-1303

황윤영

인천 부평구 산곡동 신명스카이뷰 102-303

(뒷면에 계속)

(74) 대리인

김원준

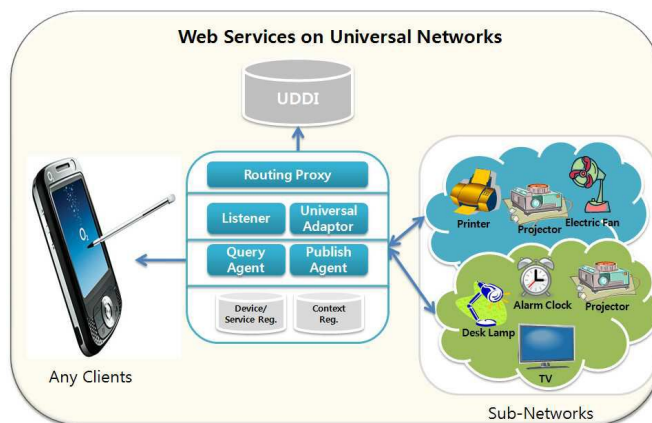
전체 청구항 수 : 총 2 항

(54) 서브네트워크의 서비스 동적 상태 반영을 위한 장치 및 방법

### (57) 요약

본 발명은 유비쿼터스 환경에서 서브네트워크의 서비스 동적 상태 반영을 위한 장치 및 방법에 관한 것으로 더욱 상세하게는, 어댑터, 등록 에이전트, 디바이스/서비스 레지스트리, 컨텍스트 레지스트리 및 UDDI를 포함하는 것으로서, (A) 상기 어댑터는, 각 서브네트워크의 디바이스 및 서비스의 명세 정보와 동적상태 정보를 수집하여 등록 에이전트에 전달하고, 서브네트워크의 서비스를 가상 웹서비스로 변환하여 상기 등록 에이전트에 전달하여 상기 UDDI에 기록하며; (B) 상기 등록 에이전트는, 상기 어댑터로부터 전달받은 상기 명세 정보와 동적상태 정보를 각각 상기 디바이스/서비스 레지스트리와 컨텍스트 레지스트리 및 UDDI 레지스트리에 저장하는; 유비쿼터스에서 서브네트워크의 서비스 동적 상태 반영을 위한 장치 및 이를 이용한 방법에 관한 것이다.

대표도 - 도1



(72) 발명자

**오일진**

대전 대덕구 신탄진동 140-6번지

**임형준**

대전 대덕구 송촌동 선비마을아파트 502-904

이 발명을 지원한 국가연구개발사업

과제고유번호 IITA-2008-C1090-0801-0031

부처명 지식경제부

연구사업명 대학IT연구센터육성지원사업

연구과제명 지능형 유비쿼터스 비즈니스 소프트 인프라 기술

주관기관 서울대학교

연구기간 2002.8~2010.12

---

## 특허청구의 범위

### 청구항 1

어댑터, 등록 에이전트, 디바이스/서비스 레지스트리, 컨텍스트 레지스트리 및 UDDI를 포함하는 서브네트워크의 서비스 동적 상태 반영을 위한 장치에 있어서,

(A) 상기 어댑터는, 각 서브네트워크의 디바이스 및 서비스의 명세 정보와 동적상태 정보를 수집하여 등록 에이전트에 전달하고, 서브네트워크의 서비스를 가상 웹서비스로 변환하여 상기 UDDI에 기록하며;

(B) 상기 등록 에이전트는, 상기 어댑터로부터 전달받은 상기 명세 정보와 동적상태 정보를 각각 상기 디바이스/서비스 레지스트리와 컨텍스트 레지스트리에 저장하는;

것을 특징으로 하는 유비쿼터스에서 서브네트워크의 서비스 동적 상태 반영을 위한 장치.

### 청구항 2

제 1 항에 의한 장치를 이용한 것으로서,

(A) 다음의 단계로 이루어지는 **사용자 등장 및 서비스 발견 과정**

㉠ 상기 장치를 사용하기 위해 사용자가 상기 장치를 발견하고 사용 환경을 구축하는 단계

㉡ 상기 질의 에이전트를 통해 사용자가 UDDI에 등록된 웹서비스 및 유비쿼터스 서비스를 발견하는 단계

㉢ 만약 사용자가 웹서비스를 선택한다면, 사용자에게 웹서비스를 연결해 주고, 유비쿼터스 웹서비스를 선택했다면, 추가적인 정보를 사용자에게 요청하고, 사용자로부터 전달받은 정보에 따라 유비쿼터스 웹서비스를 연결해 주는 단계

(B) 다음의 단계로 이루어지는 **서브네트워크 디바이스 등장 과정**

㉠ 서브네트워크 내에서 사용자와 같이 동작되는 상기 어댑터가 새로 등장하거나 상태가 바뀐 서브네트워크의 디바이스로부터 전달되는 메시지를 받아상기 디바이스 또는 서비스 명세 정보와 동적상태 정보를 수집하는 단계

㉡ 상기 어댑터가, 상기 디바이스 또는 서비스가 이미 UDDI와 유니버설 레지스트리에 등록되어 있는지 확인하는 단계

㉢ 이미 등록되어 있는 디바이스 또는 서비스이면 동적 상태정보만 상기 컨텍스트 레지스트리에 업데이트하는 단계

㉣ 새로운 디바이스 또는 서비스이면 상기 어댑터가 가상 웹서비스로 변환하여 상기 등록 에이전트에 전달하여 상기 UDDI와 디바이스/서비스 레지스트리 및 컨텍스트 레지스트리에 신규 등록하는 단계

(C) 다음의 단계로 이루어지는 **서브네트워크 디바이스의 퇴장 과정**

㉠ 상기 어댑터가 소정의 시간간격으로 서브네트워크의 디바이스들에게 서비스 가능여부를 문의하는 단계

㉡ 상기 서비스 가능여부 문의에 대한 응답이 소정의 시간동안 회신되지 않거나, 연결종료(Bye) 메시지가 있는 디바이스에 대하여, 상기 어댑터가 연결종료 정보를 등록 에이전트에 전달하고, 상기 등록 에이전트는 상기 연결종료 정보를 상기 컨텍스트 레지스트리에 전달하여 상기 디바이스의 상태정보를 변경하도록 하는 단계:

를 포함하는 것을 특징으로 하는 유비쿼터스에서 서브네트워크의 서비스 동적 상태 반영방법.

## 명세서

### 발명의 상세한 설명

## 기술 분야

[0001] 본 발명은 유비쿼터스 환경에서 서브네트워크의 서비스 동적 상태 반영을 위한 장치 및 방법에 관한 것이다.

## 배경 기술

[0002] 유비쿼터스 환경에서는 언제, 어디서든, 어떤 네트워크에 있는 어떤 디바이스의 어떤 서비스도 이용할 수 있어야 한다. 현재, 사용되고 있는 유비쿼터스 환경에서의 서비스 발견 기술들로는 Jini(Java Intelligent Network Infra-structure), UPnP(Universal Plug and Play), DPWS(Device Profile for Web Services), HAVi(Home Audio Video interoperability) 등이 있다. 이들은 자신만의 통신 프로토콜 및 구동 환경을 갖는 서비스만을 사용자가 발견할 수 있도록 하고 있다. 사용자 역시 각각의 통신 프로토콜을 따라야 하며, 구동 환경을 갖추어야 한다. 이와 같은 환경에서 유비쿼터스를 실현하기 위해서 동적으로 서비스를 발견할 수 있는 서비스 발견 방법이 필요하다.

[0003] 유비쿼터스 환경에서 요구되는 서비스 발견 방법은 기존의 서비스 발견 방법과는 다른 특징을 가져야 한다. 이 서비스 발견 방법은 각각의 네트워크 특성을 반영해야 할 뿐만 아니라 유비쿼터스 환경에서 나타나는 동적인 요소도 고려해야 한다. 유비쿼터스 환경에서의 동적인 요소(동적 상태)로는 서비스 및 사용자의 위치 변화, 서비스 상태 변화 등이 해당된다. 또한, 사용자 및 서비스가 통신 프로토콜 및 구동환경에 구애받지 않도록 해야 한다.

[0004] 유비쿼터스 환경에서 사용되는 서비스 발견 기술에는 Jini, UPnP, DPWS, HAVi 등이 있으며, 웹 서비스 발견 기술에는 WS-Discovery가 있다.

[0005] Jini 시스템은 클라이언트(client), 서비스 제공자(service provider), 룩업 서비스(lookup service)로 구성되며 이들 삼자간의 상호 통신으로 이루어진다. Jini 서비스 구조의 핵심은 발견(discovery), 조인(join), 룩업(lookup) 프로토콜이다. 발견 프로토콜은 서비스나 클라이언트가 네트워크상에서 룩업 서비스를 찾는 과정으로 서비스는 자신을 등록하기 위해, 클라이언트는 등록된 서비스를 이용하기 위해 각각 룩업 서비스를 찾는다. 조인은 서비스 객체가 룩업 서비스에 위치한 후 Jini 공동체에 참여하는 과정이다. 룩업 프로토콜은 클라이언트나 사용자가 룩업 서비스를 통해 자신이 원하는 서비스를 가져오는 과정이다.

[0006] Jini 공동체에 포함된 클라이언트들이 서비스를 이용하기 위해서는 다음과 같은 과정을 거쳐야 한다. 서비스 제공자는 로컬 네트워크상에서 멀티캐스트 요청을 통해 룩업 서비스를 발견하여 서비스 객체 및 서비스 속성을 룩업 서비스에 등록한다. 서비스 객체는 서비스 실행을 시작할 수 있는 방법과 함께 서비스를 위한 Java 프로그래밍 인터페이스를 포함하고 있다. 클라이언트가 Java 유형이나 서비스의 속성 중에 따라 서비스를 요청하면 서비스 객체의 사본이 클라이언트로 이동한다. 이후, 클라이언트와 서비스 제공자간의 통신이 이루어진다.

[0007] UPnP는 다양한 미디어 및 복수의 벤더가 존재하는 홈 네트워킹 환경에서 운영체제, 언어 및 하드웨어에 독립적인 서비스 환경을 제공하는 미들웨어로 단순하고 유연하며 표준에 기반 한 peer-to-peer 방식은 연결성을 제공한다. 이를 통해, 사용자는 단지 장치를 네트워크에 연결시켜 주면 네트워크상에 연결된 기존의 장치들이 자동으로 새로 추가된 장치를 발견하여 제어하거나 다른 장치가 가진 서비스를 찾을 수 있다.

[0008] UPnP는 서비스 발견을 위해 SSDP(Simple Service Discovery Protocol)를 이용한다. SSDP는 Jini의 발견, 조인, 룩업 프로토콜과 유사한 기능으로 다른 장치에게 자신의 존재를 알리거나 다른 장치 및 서비스를 발견하는 데 이용되는 프로토콜이다. 네트워크상에 새로 추가된 장치가 자신의 서비스를 알리기 위해 컨트롤 포인트에게 멀티캐스트 메시지를 보내면 이후 장치는 서비스들의 잠재적인 클라이언트가 된다. 반면에, 새로운 컨트롤 포인트가 네트워크상에 추가되면 SSDP는 검색을 위한 멀티캐스트 메시지를 보내고 메시지를 수신한 네트워크 상의 장치들은 유니캐스트 응답을 보낸다.

[0009] DPWS는 UPnP 차세대 버전이라고 할 수 있다. 따라서, UPnP와 동일하게 peer-to-peer 통신 방법을 제공하고 있다. 그러나, DPWS는 웹서비스의 발견에 목적을 두고 있으며, 이로 인해, 뒤에서 설명할 WS-Discovery 프로토

콜을 사용하고 있다.

- [0010] HAVi는 홈 네트워크용 미들웨어로서, IEEE 1394를 따르고 있다. IEEE 1394는 A/V 정보를 전송하기 위해 제안되었고, HAVi는 IEEE 1394를 따르는 A/V 기기들을 하나로 묶는 홈 네트워크 기능을 제안하고 있다. 또한, HAVi는 IEEE 1394에서 지원하는 버스 리셋과 노드 어드레싱을 사용하여 hot plugging과 Plug and Play를 지원하고 있다.
- [0011] 앞서 살펴본 기존 유비쿼터스 서비스 발견 기술들은 서로간의 통신 및 연동이 되지 않는다. 따라서 사용자는 각각의 서비스를 발견하기 위해서는 각 서비스 발견 기술을 사용하기 위한 환경을 구축해야 한다. 또한, 이들은 유비쿼터스 서비스 특징인 동적인 요소, 즉, 위치정보와 상태정보는 고려하지 않고 있다.
- [0012] 웹서비스 및 유비쿼터스 서비스를 모두 사용자가 발견하고 사용할 수 있도록 하기 위해 기존의 웹서비스 발견 프로토콜을 확장하여 사용하는 방법이 가능하다. 기존의 웹서비스 발견 기술에는 WS-Discovery가 있다.
- [0013] WS-Discovery는 멀티캐스트 방식으로 서비스를 발견하는 프로토콜이다. 사용자는 서비스를 발견하기 위해 필요한 서비스의 타입(type)과 스코프(scope)을 명시한 메시지를 멀티캐스트로 보내게 된다. 해당 메시지를 받은 서비스는 자신이 해당되는 서비스인 경우, 사용자에게 응답 메시지를 보내게 되고, 그렇지 않을 경우는 응답하지 않는다. 그 후, 사용자는 응답되어 온 서비스 목록 중 원하는 서비스를 선택해서 사용하게 된다.
- [0014] 그러나 WS-Discovery는 웹서비스만을 검색 대상으로 하고 있어, 유비쿼터스 서비스는 발견할 수 없다. 이는 사용자가 사용불가능하고, 사용자와 멀리 떨어져 있는 서비스인지 확인할 수 없는 상황에서 적당하지 않은 서비스 목록까지 사용자에게 모두 제공된다는 단점을 가진다.

## 발명의 내용

### 해결 하고자하는 과제

- [0015] 본 발명은 유비쿼터스 환경에서 사용 가능한 동적 서비스 발견을 위한 장치 및 그를 이용한 방법을 제공하는 것을 목적으로 한다.
- [0016] 본 발명은 유비쿼터스 환경에서 사용자가 자신 주변의 사용가능한 모든 서비스를 발견할 수 있도록 하는 장치 및 방법을 제공하고자 한다. 사용자 주변의 사용가능한 모든 서비스란 사용자 위치에 근접하고 사용자가 바로 사용할 수 있는 모든 서비스를 뜻한다.

### 과제 해결수단

- [0017] 전술한 과제를 해결하기 위한 본 발명은, 유비쿼터스 환경에서 서브네트워크의 서비스 동적 상태 반영을 위한 장치 및 방법을 제공한다.
- [0018] (1) 서비스 동적 상태 반영을 위한 장치
- [0019] 본 발명은, 어댑터, 등록 에이전트, 디바이스/서비스 레지스트리, 컨텍스트 레지스트리 및 UDDI를 포함하는 것으로서, (A) 상기 어댑터는, 각 서브네트워크의 디바이스 및 서비스의 명세 정보와 동적상태 정보를 수집하여 등록 에이전트에 전달하고, 서브네트워크의 서비스를 가상 웹서비스로 변환하여 상기 등록 에이전트에 전달하여 상기 UDDI에 기록하며; (B) 상기 등록 에이전트는, 상기 어댑터로부터 전달받은 상기 명세 정보와 동적상태 정보를 각각 상기 디바이스/서비스 레지스트리와 컨텍스트 레지스트리 및 UDDI 레지스트리에 저장하는; 유비쿼터스에서 서브네트워크의 서비스 동적 상태 반영을 위한 장치에 관한 것이다.
- [0020] (2) 서비스 동적 상태 반영을 위한 방법
- [0021] 또한 본 발명은 상기 장치를 이용한 것으로서, (A) ① 상기 어댑터가 새로 등장하거나 상태가 바뀐 서브네트워크의 디바이스로부터 전달되는 메시지를 받아 상기 디바이스 또는 서비스 명세 정보와 동적상태 정보를 수집하

는 단계 ⑥ 상기 어댑터가, 상기 디바이스 또는 서비스가 UDDI와 유니버설 레지스트리에 등록되어 있는지 확인하는 단계 ⑦ 이미 등록되어 있는 디바이스 또는 서비스이면 동적 상태정보만 상기 컨텍스트 레지스트리에 업데이트하는 단계 ⑧ 새로운 디바이스 또는 서비스이면 상기 어댑터가 가상 웹서비스로 변환하여 상기 등록 에이전트에 전달하여 상기 UDDI와 디바이스/서비스 레지스트리 및 컨텍스트 레지스트리에 신규 등록하는 단계들로 이루어지는 서브네트워크 디바이스 등장 과정과, (B) ③ 상기 어댑터가 소정의 시간간격으로 서브네트워크의 디바이스들에게 서비스 가능여부를 문의하는 단계 ④ 상기 서비스 가능여부 문의에 대한 응답이 소정의 시간동안 회신되지 않거나, 연결종료(Bye) 메시지가 있는 디바이스에 대하여, 상기 어댑터가 연결종료 정보를 등록 에이전트에 전달하고, 상기 등록 에이전트는 상기 연결종료 정보를 상기 컨텍스트 레지스트리에 전달하여 상기 디바이스의 상태정보를 변경하도록 하는 단계들로 이루어지는 서브네트워크 디바이스의 퇴장 과정을 포함하는 유비쿼터스에서 서브네트워크의 서비스 동적 상태 반영방법에 관한 것이다.

## 효 과

- [0022] 본 발명에 의한 유비쿼터스 환경에서 사용가능한 동적 서비스 발견을 위한 장치 및 방법에 의하면, 서브네트워크의 서비스 및 사용자의 환경(동적 상태) 변화에 따른 서비스 발견이 가능하게 되고, 서비스의 구동환경과 통신 프로토콜에 상관없이 서비스 발견이 가능하게 된다.
- [0023] 본 발명에 의해 산재된 많은 유비쿼터스 기술들의 서비스뿐만 아니라 웹서비스까지도 발견 할 수 있게 된다.

## 발명의 실시를 위한 구체적인 내용

- [0024] 이하 첨부된 도면을 참조하여 본 발명을 보다 상세히 설명한다. 그러나 이러한 도면은 본 발명의 기술적 사상의 내용과 범위를 쉽게 설명하기 위한 예시일 뿐, 이에 의해 본 발명의 기술적 범위가 한정되거나 변경되는 것은 아니다. 또한 이러한 예시에 기초하여 본 발명의 기술적 사상의 범위 안에서 다양한 변형과 변경이 가능함은 당업자에게는 당연할 것이다.
- [0025] 도 1은 본 발명에 의한 장치의 예시적 구조를 보여주는 블록도이다. 여기서 상기 동적상태 반영을 위한 장치(이하 '본장치'라 함)는 어댑터, 등록 에이전트, 질의 에이전트, 디바이스/서비스 레지스트리, 컨텍스트 레지스트리 및 UDDI를 포함하여 구성된다. 이들의 기능은 다음과 같다.,
- [0026] (A) 어댑터 : 어댑터는, 서브네트워크와 상기 본장치 사이의 중계 역할을 담당한다. 이질적인 서브네트워크와 본장치가 통신을 하기 위한 브리지 역할로 서브네트워크의 서비스를 가상 웹서비스로 변환하여 유니버설 서비스를 검색을 지원한다.
- [0027] 서브네트워크는 자신의 영역안에 있는 서브네트워크 사용자에게 자신의 이벤트 정보를 전달하게 된다. 따라서 상기 어댑터는 마치 서브네트워크 내의 사용자처럼 보여지도록 해서 서브네트워크의 이벤트 정보를 전달받는다.
- [0028] 상기 어댑터는 다음과 같은 기능을 갖는다.
- [0029] ① 메시지 전송 (Message Transfer) : 서브네트워크와 본장치에 필요한 정보들은 전송하는 기능을 담당한다. 이는 본장치와 서브네트워크의 통신 프로토콜 등이 이질적이기 때문에 어댑터를 통해 중재된다.
- [0030] ② 서브네트워크 서비스의 검색 (Sub-network Services Discovery) : 서브네트워크의 서비스를 가상 웹서비스로 변환하고 레지스트리에 저장할 정보들을 검색한다. 검색한 정보들은 가상 웹서비스 생성 후 등록 에이전트로 보내 디바이스/서비스 레지스트리와 컨텍스트 레지스트리 및 UDDI 레지스트리에 저장한다.
- [0031] ③ 가상 웹서비스 생성 (Virtual Web Services Creation) : 사용자단말은 서브네트워크의 존재를 인식하지 않고 서비스를 검색하고 사용하기 때문에 서브네트워크의 서비스를 웹서비스 형태로 변환한다. 변환된 서비스가 외부에 가상 웹서비스로 사용자단말에게 노출된다. 이를 통해, 사용자단말은 웹서비스를 사용하는 것으로 보이지만 실제로 유니버설 어댑터를 통해 서브네트워크의 서비스가 제공된다
- [0032] ④ 이벤트 및 알림 (Events and Notifications) : 서브네트워크 및 서브네트워크의 디바이스의 등장과 퇴장이 발생하였을 경우 이를 반영하여 플러그 앤 플레이를 지원한다. 또한, 사용자단말이 서비스의 사용을 종료하였을

경우, 서비스의 상태 정보를 변경하여 서비스가 사용 가능하도록 한다.

- [0033] ⑤ 웹서비스와의 연동(Interoperability with Web Services) : 서브네트워크내 사용자에게 UDDI 레지스트리에 등록된 서비스 목록을 제공해 주고, 사용자가 웹서비스 사용을 요구할 시 바로 사용할 수 있도록 연결해 주는 역할을 담당한다.
- [0034] (B) 등록 에이전트 (Publish Agent) : 상기 어댑터에 의해 수집된 메타데이터 정보를 레지스트리에 등록한다. 서브네트워크에 존재하는 디바이스의 메타데이터 정보가 어댑터에 의해 수집되면 디바이스/서비스 레지스트리와 컨텍스트 레지스트리에 각각 저장하는 기능을 한다.
- [0035] (C) 질의 에이전트 (Query Agent) : 새로 등장하거나 퇴장했다가 다시 등장하는 디바이스 또는 서비스인지를 확인하기 위해서 상기 어댑터에서 서비스 등록 여부를 요청하면, 서비스 ID를 이용하여 디바이스/서비스 레지스트리에서 서비스 검색을 하는 기능을 한다.
- [0036] (D) UDDI (Universal Description, Discovery and Integration) : 사용자에게 사용가능하고 근접한 서비스를 제공하고, 유비쿼터스 서비스 뿐 아니라 웹서비스까지도 모두 제공해 주기 위해서 유비쿼터스 서비스를 모두 웹 서비스로 변환하여 저장하고 관리한다. 웹서비스로 변환된 유비쿼터스 서비스는 UDDI에 저장되어 관리되는데, 이 때, Business Entity에 유비쿼터스 서비스인지 아닌지에 대한 내용을 명시하도록 한다. 또한, UDDI에 등록되어 관리 될 수 없는 유비쿼터스 서비스의 부가적인 내용은 레지스트리에 등록되어 관리되도록 한다.
- [0037] (E) 디바이스/서비스 레지스트리 (Device/Service Registry) : 메타데이터 즉, 각 서브네트워크의 디바이스 및 서비스의 명세 정보와 WSUN 디바이스의 명세 정보를 저장한다. 이를 통해, 본장치는 사용자단말의 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. 하기 표 1과 같은 디바이스 및 서비스 정보를 추출하는 것이 바람직하다.

표 1

| 명세 정보              | 설 명                             |
|--------------------|---------------------------------|
| Sub-networkType    | 본장치에서 다수의 서브네트워크를 관리하기 위한 타입    |
| sServiceID         | 서브네트워크에서의 서비스 ID                |
| uServiceID         | 본장치에서의 서비스 ID                   |
| uDeviceID          | 본장치에서 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                      |

- [0038]
- [0039] (F) 컨텍스트 레지스트리 (Context Registry) : 유비쿼터스 환경에서는 디바이스의 위치가 자주 변하기 때문에 컨텍스트 레지스트리는 디바이스의 상태 및 위치 정보를 저장하여 사용자단말의 상태 및 위치 정보에 맞는 서비스를 검색하도록 상황 정보로 쓰인다. 예를 들어, 회의 장소 내에서 사용 가능한 프린터를 검색하는 것을 들 수 있다. 이는 추후 사용자단말의 프로파일 등의 온톨로지 검색으로 확장할 것이다.
- [0040]
- [0041] 본장치는 이질적이고 산재한 서브네트워크의 상호운용성을 지원하는 미들웨어로서의 기능과 역할을 하게 된다. 본장치는 서비스 사용자단말과 서비스 제공자 사이에 위치해서 서브네트워크의 서비스와 디바이스가 제공하는 서비스를 검색하고 사용할 수 있다.

[0042] 본장치는 서비스 발견, 등록 및 관리하는 과정에서 다음과 같은 기능을 하게 된다.

[0043] (A) 사용자 등장 및 동적 서비스 발견 시

[0044] 사용자는 상기 장치를 사용하기 위해, 멀티캐스트 메시지를 이용하여 상기 장치를 찾는다. 사용자의 메시지를 받은 상기 장치의 리스너는 상기 장치를 사용하는데 필요한 정보를 사용자에게 전달한다. 이 후에 사용자는 자신이 필요한 서비스를 상기 장치를 이용할 수 있다. 사용자는 서비스 이름, 설명정보 등을 이용하여 서비스를 검색할 수 있다. 서비스 검색 요청을 받은 상기 장치의 에이전트는 UDDI 레지스트리를 검색하여 검색된 결과를 사용자에게 전달한다. 검색된 서비스 목록에는 유비쿼터스 서비스 및 기존 웹서비스가 모두 포함되어 있다. 만약, 사용자가 기존 웹서비스의 사용을 요구한다면, 사용자에게 기존 웹서비스와의 바인딩을 지원한다. 그러나, 유비쿼터스 서비스를 요청할 경우에는, 유비쿼터스 서비스를 사용할 때 추가적으로 필요하게 되는 정보를 사용자로부터 전달받아, 서비스를 연결해 주게 된다. 도 2에 본 발명에 의한 장치 및 방법이 적용된 시스템에서 사용자가 동적 서비스를 발견하는 과정의 흐름을 도시하였다.

[0045] (B) 유비쿼터스 네트워크 및 서비스의 등장시

[0046] 유비쿼터스 네트워크의 등장은 peer-to-peer 통신 네트워크에서는 발생하지 않는다. 그러나 Jini와 같이 서비스 레지스트리를 가지고 있는 미들웨어 방식의 네트워크에서는 서비스 레지스트리의 등장이 곧 유비쿼터스 네트워크의 등장이라고 할 수 있다. 유비쿼터스 네트워크 및 서비스의 등장과정의 흐름도를 도 3에 도시하였다.

[0047] 유비쿼터스 네트워크의 서비스를 웹서비스로 변환하기 위해서는 먼저, 서비스 정보를 가져오는 과정이 필요하다(1~3). 서브네트워크의 디바이스는 자신의 등장을 US Broker에게 알리게 되는데, 이 메시지는 어댑터에 의해 수신된다. 디바이스 등장 메시지를 받은 어댑터는 서비스 등록에 필요한 서비스 정보를 서브네트워크 디바이스에게 요청한다. 서비스 정보 요청 메시지를 받은 서브네트워크 디바이스는 자신의 상세 정보를 어댑터에게 다시 송신하게 된다. 이 과정이 끝나면 가져온 서비스 목록이 이미 웹서비스 변환되어 저장되어 있는지 확인하는 절차가 요구된다(4~7). 이를 위해, 어댑터는 질의 에이전트에게 등장한 서비스 ID를 전달하고, 서비스 검색을 요구한다. 질의 에이전트는 서비스 ID를 이용하여 디바이스/서비스 레지스트리에서 일치하는 서비스가 있는지 검색하고, 있으면 해당 서비스 ID를, 없으면 검색결과가 없다는 메시지를 어댑터에게 보낸다. 만약, 이미 등록되어 있는 서비스라면 서비스 상태만 사용가능한 상태로 변환하고(9~12), 새로운 서비스라면 해당 서비스의 웹서비스를 생성하고(13) UDDI 및 디바이스/서비스 레지스트리와 컨텍스트 레지스트리에 저장한다(14~22). 가상 웹서비스를 생성하는 데는 많은 시간 및 비용이 소요되므로, 새로 등장한 서비스에 대해서만 가상 웹서비스를 생성하도록 하는 것이 바람직하다. 서비스 정보외에 서비스 및 디바이스의 위치 정보도 컨텍스트 레지스트리에 함께 저장하는데, 이는 사용자에게 근접한 서비스 목록을 제공하기 위함이다.

[0048] (C) 유비쿼터스 네트워크 및 서비스의 퇴장시

[0049] 유비쿼터스 네트워크 및 서비스 퇴장의 경우는 크게 2가지로 나뉜다. 퇴장하는 네트워크나 서비스가 그 사실을 통보하고 퇴장하는 경우와 그렇지 않은 경우이다. 통보하고 퇴장하는 경우는 해당 서비스 상태를 사용불가능으로 변환하면 된다(도 4). 후에, 같은 네트워크 및 서비스가 재등장 할 경우, 가상 웹서비스 생성 비용을 줄이기 위해, 서비스 상태만을 사용불가능으로 변환되도록 하는 것이 좋다. 그러나, 반대의 경우, 오류에 의한 경우에는 해당 서비스에게 메시지를 보내서 정해진 시간 안에 응답 메시지가 오지 않으면, 퇴장했다고 보고 서비스 상태를 사용불가능으로 변환한다(도 5).

[0050] (D) 동적 서비스 발견 방법을 지원하는 서비스의 등장시

[0051] 동적 서비스 발견 방법을 이미 지원하고 있는 서비스의 경우에는 웹서비스로의 변환이 필요 없다. 도 6에 도시된 것처럼, 이 경우에는 자신이 이미 등록되어 있는 경우와 그렇지 않은 경우만을 판단하여, 등록되어 있는 경우에는 유비쿼터스 네트워크 및 서비스의 등장과 동일하게 서비스의 상태만을 사용가능으로 변환한다. 이미 등록되어 있지 않은 서비스의 경우에는 바로 해당 서비스를 UDDI와 유니버설 레지스트리에 등록한다.

[0052] (E) 동적 서비스 발견 방법을 지원하는 서비스의 퇴장시

[0053] 동적 서비스 발견 방법을 지원하는 서비스의 퇴장은 앞서 기술한 유비쿼터스 서브네트워크 및 서비스의 퇴장과 동일하다. 서비스가 자신의 퇴장 사실을 통보하면, 해당 서비스의 상태를 사용불가능으로 변환하게 된다(도 7).

### 도면의 간단한 설명

[0054] 도 1은 본 발명에 의한 장치의 예시적 구조를 보여주는 블록도.

[0055] 도 2는 본 발명에서 사용자 등장 및 서비스 발견 과정을 나타낸 흐름도

[0056] 도 3은 본 발명에서 유비쿼터스 네트워크 및 서비스의 등장과정을 나타낸 흐름도.

[0057] 도 4은 본 발명에서 유비쿼터스 네트워크 및 서비스의 정상적인 퇴장과정을 나타낸 흐름도.

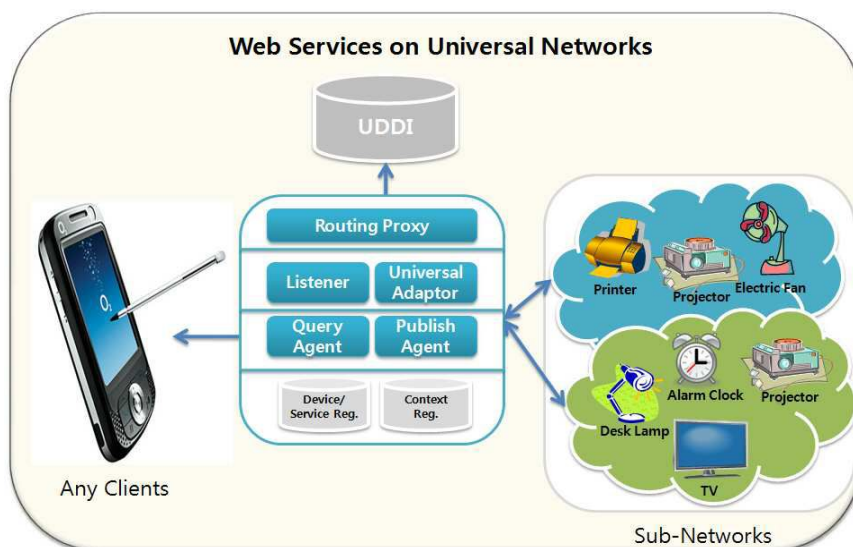
[0058] 도 5는 본 발명에서 유비쿼터스 네트워크 및 서비스의 오류에 의한 퇴장과정을 나타낸 흐름도.

[0059] 도 6은 본 발명에서 신규하지 않은 유비쿼터스 네트워크 및 서비스의 등장과정을 나타낸 흐름도.

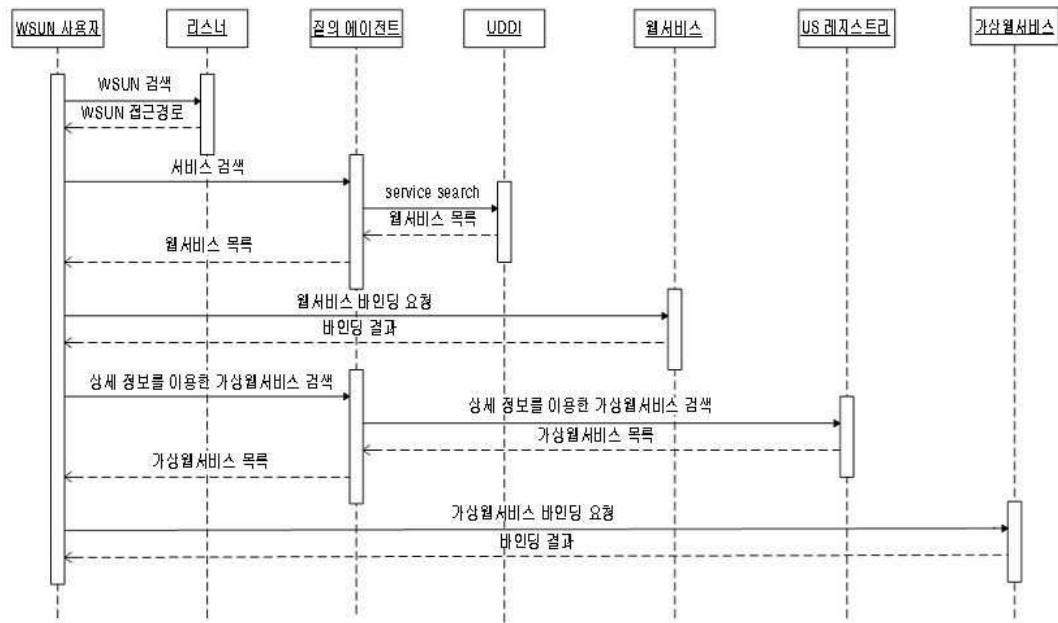
[0060] 도 7은 본 발명에서 신규하지 않은 유비쿼터스 네트워크 및 서비스의 등장과정을 나타낸 흐름도.

### 도면

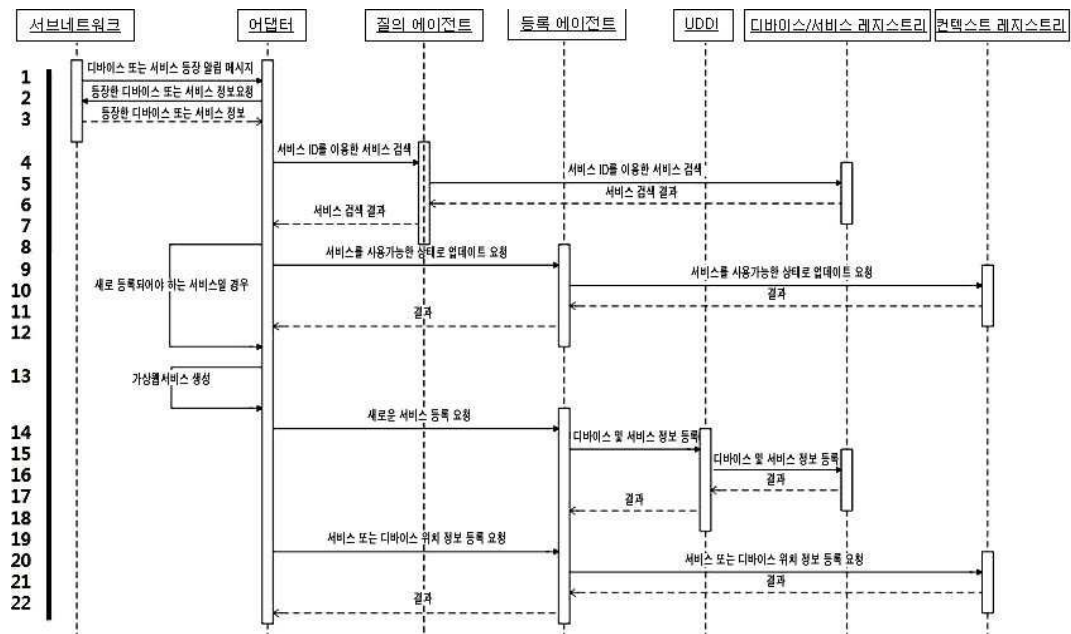
#### 도면1



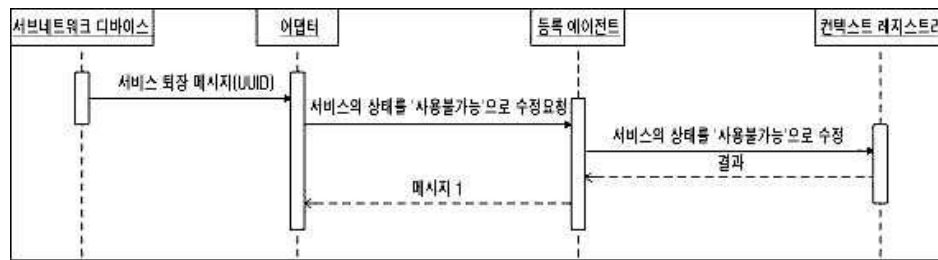
도면2



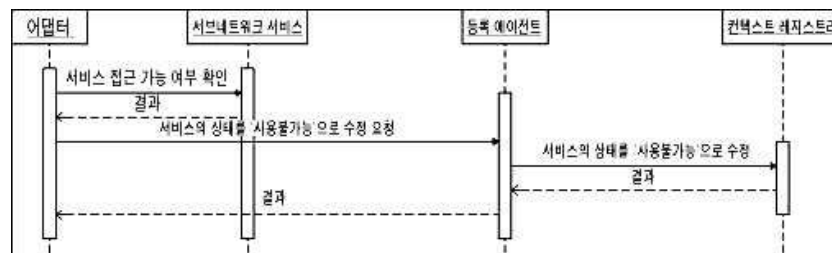
도면3



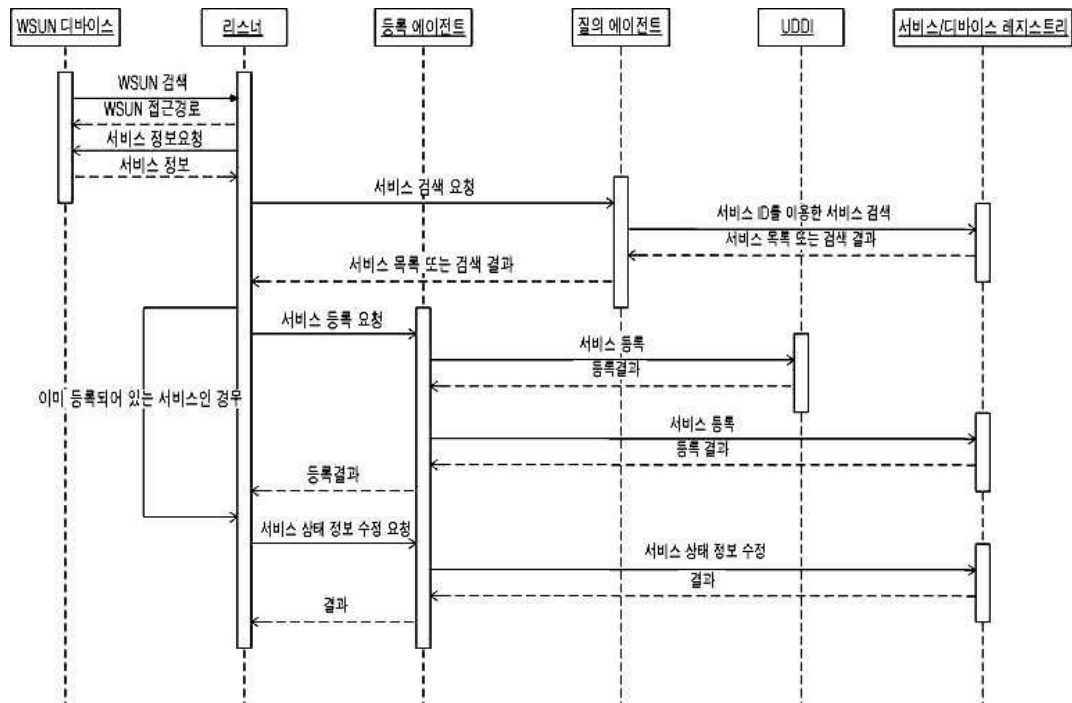
도면4



도면5



도면6



도면7

