

## 시멘틱 웹 서비스 기반의 유비쿼터스 로봇 시스템의 설계 및 구현

하영국<sup>o</sup> 손주찬 조영조  
한국전자통신연구원 지능형로봇연구단  
{ygha<sup>o</sup>, jcsohn, youngjo}@etri.re.kr

### Design and Implementation of a Ubiquitous Robotic System based on Semantic Web Services Technology

Young-Guk Ha<sup>o</sup> Joo-Chan Sohn Young-Jo Cho  
Intelligent Robot Research Division, ETRI

#### 요 약

유비쿼터스 로봇은 최근에 소개된 새로운 형태의 네트워크 기반 로봇으로서 유비쿼터스 컴퓨팅 환경하에서 사용자에게 끊임없고 자연스러운 상황 기반의 서비스를 제공하는 소프트웨어 및 하드웨어 로봇을 통칭하는 말이다. 따라서 이와 같은 유비쿼터스 로봇 시스템은 사용자에게 서비스 제공을 위한 자율성(Autonomy), 편재성(Ubiquity), 상황 인지성(Context-awareness) 및 연속성(Seamlessness)의 요구사항을 만족해야 하며, 이를 위해 센서 네트워크, 정보가전 기기, 인터넷 정보 콘텐츠 등과 같은 다양한 유비쿼터스 컴퓨팅 자원을 기반으로 자율적인 서비스 구성 및 수행을 할 수 있는 기능을 제공해야 한다. 본 논문에서는 지능형 서비스 에이전트를 위한 차세대 웹 기술인 시멘틱 웹 서비스를 기반으로 유비쿼터스 로봇 시스템의 프레임워크를 설계하고 Evolution Robotics사의 ERSP 3.0 로봇 플랫폼을 기반으로 프로토타입 시스템을 구현한다.

#### 1. 서 론

컴퓨터 통신 및 네트워크 기술의 눈부신 발전과 더불어 인터넷을 기반으로 하는 온라인 및 네트워크 로봇[7]에 대한 연구가 활발히 진행되고 있다. 네트워크 로봇 기술은 크게 두 가지의 세부 기술로 분류할 수 있는데 사용자가 원격지에 있는 로봇에 네트워크를 통해 명령을 전송하여 제어하는 원격제어(Tele-operated) 로봇 및 로봇과 센서가 네트워크를 통해 상호 통신하며 로봇의 행동을 자율적으로 제어하는 자율(Autonomous) 로봇이 그것이다. 현재까지 월드 와이드 웹 및 분산 객체 기술을 기반으로 다양한 네트워크 로봇을 위한 연구가 진행되어 왔다. 웹 기반의 로봇 시스템은 주로 HTTP 프로토콜, CGI 및 Java 애플릿을 이용하여 원격 구동기를 제어하는 구조로서 USC의 원격 굴착 시스템 Mercury, CMU의 실내용 모바일 로봇 Xavier 및 RWU의 PumaPaint 등이 있다. 또한 CORBA 및 Java RMI 등과 같은 원격 객체 기술을 기반으로 하는 시스템으로는 NRSP(Network Robot Service Platform), DAIR(Distributed Architecture for Internet Robot) 등이 있다.

최근에는 유비쿼터스 컴퓨팅이 새로운 컴퓨팅 패러다임으로 등장하면서 유비쿼터스 컴퓨팅의 개념을 기반으로 한 네트워크 로봇인 유비쿼터스 로봇[3]이 소개된바 있다. 현재 9대 국가 신성장동력 과제 중의 하나로 개발중에 있는 URC(Ubiquitous Robotic Companion)[4] 또한 일종의 유비쿼터스 로봇으로서 언제 어디서나 나와 함께 하면서 내가 원하는 서비스를 제공하는 IT 기반의 지능형

서비스 로봇을 의미한다. 이와 같은 URC의 비전을 실현하기 위해서 로봇 시스템은 사용자에게 서비스 제공을 위한 자율성, 편재성, 상황 인지성 및 연속성의 요구 사항을 만족하는 해야 하며, 이를 위해 유비쿼터스 센서 네트워크(USN), 정보가전 기기, 인터넷 정보 콘텐츠 등과 같은 다양한 유비쿼터스 컴퓨팅 자원을 기반으로 자율적인 서비스 구성 및 수행을 할 수 있는 기능을 제공해야 한다. 그러나 앞서 기술하였던 기존의 인터넷 로봇 시스템들에 있어서는 자율적인 서비스 제공과 상황 인지를 위한 기체가 인식가능한 지식의 부재와 편재성과 서비스 연속성의 제공을 어렵게 하는 시스템의 구조 및 네트워크 보안의 문제점 등으로 인하여 유비쿼터스 로봇을 위한 요구 사항을 충족시키기가 매우 어렵다.

본 논문에서는 지능형 서비스 에이전트를 위한 차세대 웹 기술인 시멘틱 웹 서비스를 기반으로 URC의 실현을 위한 유비쿼터스 로봇 시스템 프레임워크를 설계하고 Evolution Robotics사의 ERSP 3.0 로봇 플랫폼을 기반으로 프로토타입 시스템을 구현하도록 한다.

#### 2. 시멘틱 웹 서비스 기술

주로 인간의 판독을 위한 문자 혹은 영상 데이터의 저장소로 사용되던 웹은 현재 정보 검색 서비스, 웹 카메라/센싱 서비스 및 전자상거래 서비스 등 다양한 정보 서비스의 제공자로 진화하고 있다. 이러한 진화는 W3C가 제정한 분산 객체 기술인 웹 서비스[8]가 등장하면서 더욱 가속화 되고 있다. 웹 서비스는 CORBA나 RMI와 같은 기

존의 분산 객체 기술에서 사용하는 통신 프로토콜인 IIOP(Internet Inter ORB Protocol) 대신 HTTP, XML 및 SOAP(Simple Object Access Protocol) 등과 같은 국제 표준 웹 기술을 사용함으로써 다양한 유비쿼터스 장치를 위한 통신 프레임워크로서 적합한 특징을 제공한다.

이와 더불어 지능형 에이전트가 자율적으로 서비스를 탐색, 구성 및 실행하기 위해서는 기계가 이해할 수 있는 서비스의 의미를 표현하는 수단이 필요하다. 시멘틱 웹 서비스[9]는 이러한 목적을 위해 개발된 기술로서 웹 상의 서비스에 대한 의미를 OWL-S[5] 온톨로지를 기반으로 기술하도록 하여 서비스 에이전트로 하여금 지능적, 자율적인 서비스 제공을 실현할 수 있도록 한다. 그림 1은 시멘틱 웹 서비스를 통한 에이전트 및 서비스 객체간의 기본적인 통신 프로토콜을 나타낸다.

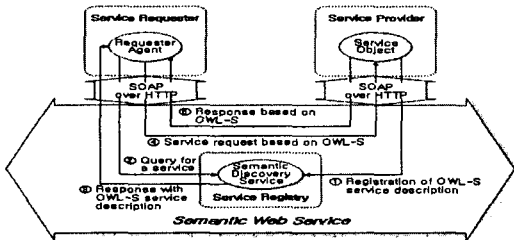


그림 1 시멘틱 웹 서비스 프로토콜

시멘틱 웹 서비스 기술에서 서비스의 의미를 표현하는 상위 온톨로지인 OWL-S는 크게 다음과 같은 세가지의 클래스의 시멘틱 마크업으로 구성된다.

1) **서비스 프로파일**: 서비스 프로파일은 기술된 서비스가 에이전트의 요구를 만족할 수 있는 서비스 인지를 매칭하기 위한 정보를 제공하는 온톨로지로서 서비스의 입력(Input), 출력(Output), 사전조건(Precondition) 및 효과(Effect)에 대한 기술을 포함하고 있다.

2) **서비스 모델**: 서비스 모델은 서비스의 내부 프로세스 모델을 기술하는 온톨로지로서 서비스 프로파일과 함께 보다 상세한 서비스의 탐색을 가능하게 한다. 또한 특정 복합 서비스(Composite service)를 구성하기 위해 단위 서비스(Atomic service)를 통한 프로세스의 구성 및 서비스 실행 모니터링을 위한 정보를 제공한다.

3) **서비스 그라운드**: 서비스 그라운딩은 실제 웹 서비스에 접근하기 위한 상세 정보를 제공하는 온톨로지로서 통신 프로토콜, 메시지 포맷, 서비스 포트 및 구체적인 데이터 타입 등과 같은 세부 실행 정보를 기술한다.

그림 2는 온도 센서 서비스를 위한 OWL-S 서비스 표현(서비스 프로파일 및 서비스 모델)의 예이다.

```

...
<!-- Temperature sensor service description -->
<service:Service rdf:ID="GetDataService">
  <service:presents rdf:resource="#GetDataProfile" />
  <service:describedBy rdf:resource="#GetDataProcessModel" />
  <service:supports rdf:resource="#GetDataGrounding" />
</service:Service>

<!-- Profile description -->
<sensor:Profile:TemperatureSensorService rdf:ID="GetDataProfile">
  <service:presents rdf:resource="#GetDataService" />
  <profile:has_process rdf:resource="#GetDataProcessModel" />
  <profile:serviceName>Temperature Sensor #1</profile:serviceName>
  <profile:textDescription>This service gets the data from the sensor.</profile:textDescription>
  <profile:qualityRating>
  <profile:QualityRating rdf:ID="Temperature-Sensor-1-Rating">
    <profile:qualityRating>
    <profile:ratingName>SomeRating</profile:ratingName>
    <profile:rating rdf:resource="http://etrl.re.kr/services/Concepts.owl#Excellent" />
    <profile:QualityRating>
  </profile:QualityRating>
  </profile:qualityRating>
  <profile:hasOutput rdf:resource="#Data" />
</sensor:Profile:TemperatureSensorService>

<!-- Process Model description -->
<process:ProcessModel rdf:ID="GetDataProcessModel">
  <service:describes rdf:resource="#GetDataService" />
  <process:hasProcess rdf:resource="#GetDataProcess" />
</process:ProcessModel>

<process:AtomicProcess rdf:ID="GetDataProcess">
  <process:hasInput>
  <process:input rdf:ID="SensorID">
    <process:input>
    <process:parameterType rdf:resource="http://www.w3.org/2001/XMLSchema#int" />
  </process:input>
  <process:hasOutput>
  <process:UnconditionalOutput rdf:ID="Data">
    <process:parameterType rdf:resource="http://www.w3.org/2001/XMLSchema#float" />
  </process:UnconditionalOutput>
  <process:hasOutput>
  <process:UnconditionalOutput rdf:ID="Time">
    <process:parameterType rdf:resource="http://www.w3.org/2001/XMLSchema#dateTime" />
  </process:UnconditionalOutput>
  </process:hasOutput>
</process:AtomicProcess>
...

```

그림 2 OWL-S 온도 센서 서비스 표현

하여 그 의미가 기술되어 서비스 레지스트리에 등록된다. 우선 서비스 에이전트는 사용자 인터페이스를 통하여 서비스 요청을 입력 받아 서비스 구성 엔진을 호출하게 된다. 서비스 구성 엔진은 지식베이스로부터 OWL 추론 엔진[1]을 통해 요청된 서비스 프로세스를 구성하기 위하여 서비스 템플릿을 검색하게 된다. 서비스 템플릿은 OWL을 기반으로 정의된 시멘틱 템플릿 기술 언어인 STDL(Service Template Description Language)을 통해서 기술되는 HTN(Hierarchical Task Network)[2] 기반의 서비스 플래닝을 위한 Decomposition 스키마로서 에이전트에게 서비스 프로세스 생성을 위한 지식을 제공한다. 서비스 구성 엔진은 프로세스를 구성하는 과정에서 서비스 탐색 모듈을 통해 서비스 레지스트리로부터 템플릿에 기술된 단위 서비스의 의미와 동일한 실제 웹 서비스 인스턴스를 검색하게 되고 이를 기반으로 실행가능한 서비스 프로세스를 생성하게 된다. 최종적으로 생성된 서비스 프로세스는 서비스 실행 모듈을 통하여 수행되며 이 과정에서 탐색된 실제 서비스들이 웹 서비스 프로토콜(SOAP)을 통해서 호출된다.

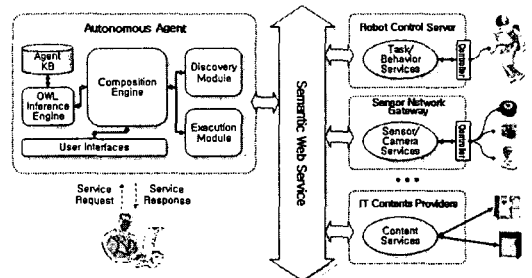


그림 3 유비쿼터스 로봇 시스템 프레임워크

### 3. 유비쿼터스 로봇 시스템의 설계

그림 3은 본 논문에서 설계하는 유비쿼터스 로봇 시스템의 프레임워크이다. 그림에서 보는 바와 같이 전체적인 구조는 소프트웨어 로봇으로서 서비스 에이전트 및 하드웨어 로봇제어 서버, 센서 네트워크 게이트웨이, IT 콘텐츠 제공자 등의 유비쿼터스 서비스들이 시멘틱 웹 서비스를 기반으로 상호 연결되어 있다. 각각의 서비스들은 웹 서비스 표준을 기반으로 구현되며, 또한 OWL-S를 이용

상기의 과정을 기반으로 하여 박물관 안내 서비스 시나리오를 설명하면 다음과 같다. 우선 박물관에 도착한 사용자는 박물관 홈페이지로부터 자신의 PDA상에 다운로드된 안내 서비스 소프트웨어 로봇(에이전트)에게 관람을 원하는 장소를 입력한다. 안내 소프트웨어 로봇은 자신의 지식베이스로부터 안내 서비스를 위한 템플릿을 검색하고 이를 기반으로 안내 서비스 프로세스를 구성한다. 안내 서비스 템플릿은 크게 세단계의 하위 템플릿으로 구성되는데, 로봇을 관람 위치로 이동시키는 서비스, 관람 대상의 정보를 검색하는 서비스 및 검색된 정보를 사용자에게 출력해주는 서비스 템플릿으로 구성된다. 다시 관람 대상 정보 검색 템플릿은 대상을 인식하는 서비스 및 인식된 대상의 정보를 검색하는 하위 서비스 템플릿으로 구성된다. 다음 단계에서 각각의 서비스 템플릿들은 서비스 탐색을 통해 검색된 실제 웹 서비스 인스턴스들로 매핑되어 실행된다. 그림 4는 상기 시나리오상의 박물관 안내 서비스 템플릿, OWL-S 서비스 모델 및 매핑된 실제 웹 서비스(MoveTo, GetImage, ObjRecognition, Speak 및 GetInformation)간의 상호 구성 관계를 보여준다.

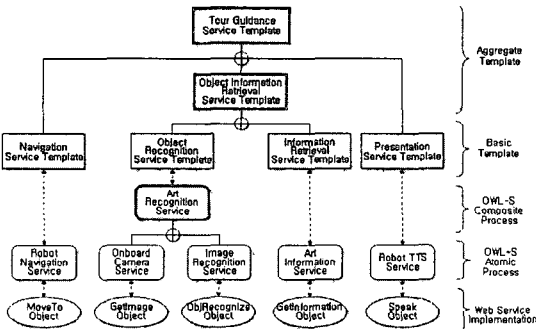


그림 4 서비스 템플릿 및 웹 서비스의 구성

유비쿼터스 로봇 프레임워크 상에서 서비스 에이전트들은 하나의 서비스 목표를 수행하기 위하여 서로 협력을 수행할 수도 있다. 이는 서비스 에이전트 스스로가 하나의 서비스 객체로서 자신의 프로파일을 서비스 레지스트리에 등록하고 협업을 원하는 다른 에이전트들로 하여금 서비스 탐색을 통해서 자신을 발견하게 함으로써 가능하다. 그림 5는 두개의 서비스 에이전트간의 서비스 레지스트리를 통한 협력 과정을 보여준다.

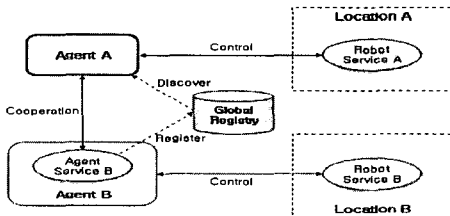


그림 5 서비스 에이전트간의 협력

#### 4. 유비쿼터스 로봇 시스템 프로토타입

그림 6은 유비쿼터스 로봇 시스템 프레임워크를 기반으로 구현한 프로토타입 시스템의 구성도이다. 본 시스템

에서는 로봇 제어 객체의 구현을 위해 Evolution Robotics사의 ERSF 3.0 로봇 플랫폼 및 Scorpion 로봇을 이용하였으며, 정보 검색 서비스 및 로봇 웹 서비스의 구현을 위하여 Apache AXIS 웹 서비스 툴킷을 사용하였다. 또한 서비스 에이전트 시스템에서는 Java기반의 사용자 인터페이스를 통해서 서비스 요청을 입력 받고, 이를 수행하기 위하여 서비스 구성 엔진을 통해 OWL-S 프로세스 모델로부터 실행가능한 형태인 BPEL4WS[6] 프로세스를 생성하고 IBM의 BPWS4J 엔진을 이용하여 실행하게 하였다. 마지막으로 본 프로토타입 시스템의 시험을 위해 ERSF의 영상인식 라이브러리를 이용하여 학습시킨 대상을 기반으로 앞서 설명한 박물관 관람 시나리오를 시연하였다.

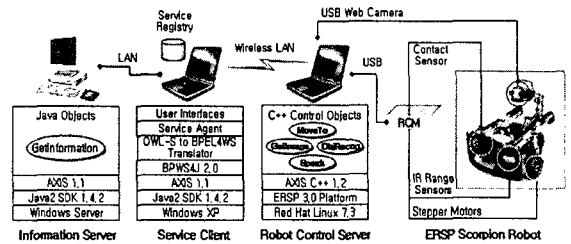


그림 6 유비쿼터스 로봇 시스템 프로토타입

#### 5. 결론

본 논문에서는 시맨틱 웹 서비스 기술을 기반으로 유비쿼터스 로봇인 URC를 위한 서비스 프레임워크를 설계하고 프로토타입 시스템을 구현하였다. 유비쿼터스 로봇은 유비쿼터스 컴퓨팅 환경에서의 지능형 서비스 로봇으로서 국가 신성장동력으로서 크게 기여할 것으로 기대되는 기술중 하나이다. 앞으로의 연구개발 방향은 본 서비스 플랫폼을 Ad Hoc 센서 네트워크 및 홈 네트워크 환경 등과 연동하여 보다 다양한 서비스의 개발이 가능하도록 확장해 나갈 계획이다.

#### 참고 문헌

- [1] 허영국, 손주찬, 함호상, " 어휘의 공리화를 이용한 Web Ontology 추론 시스템의 설계 및 구현," 2003 한국정보과학회 추계학술발표회, 건국대학교, 2003.10.25.
- [2] S. Kambhampati, et.al., " Hybrid Planning for Partially Hierarchical Domains," AAAI98, Wisconsin, Jul.1998.
- [3] J. H. Kim, " Introduction to Ubiquitous Robotics," ICARA2004, New Zealand, Dec.13.2004.
- [4] S. R. Oh, " IT Based Intelligent Service Robot," RCV03 Invited talk, Las Vegas, Oct.26.2003.
- [5] DARPA, " OWL-S: Semantic Markup for Web Services," <http://www.daml.org/services/owl-s/1.0/owl-s.html>
- [6] IBM, " Business Process Execution Language for Web Services Version 1.1," <http://www.ibm.com/developerworks/library/ws-bpel/>
- [7] IEEE, TC on Networked Robotics, <http://www.ieor.berkeley.edu/~goldberg/tc/>
- [8] W3C, Web Services, <http://www.w3c.org/2002/ws/>
- [9] W3C, SIG on Semantic Web Services, <http://www.w3c.org/2002/ws/swsig/>