

네트워크 기반 소프트웨어 로봇의 기능 확장 체계

장민수, 손주찬, 조영조(한국전자통신연구원 지능형로봇연구단)

요 약

정보 네트워크는 데이터 뿐 아니라 사물들의 연결성을 부추기고 있으며, 이에 따라 소프트웨어의 현실에 대한 영향력은 더욱 커지고 있다. 한편 로봇의 몸체(하드웨어)와 두뇌(소프트웨어)를 분리하고 네트워크를 통해 연결하는 연구 개발 시도가 늘어나고 있으며 이는 저가형의 경제성 있는 로봇을 개발하는 원동력이 되고 있기도 하다. 따라서 소프트웨어를 중심으로 한 로봇의 필요성, 역할 비중, 가능성은 그 어느 때 보다 크다. 이 글에서는 소프트웨어 로봇의 기능을 자동으로 확장할 수 있는 체계 구축의 가능성을 따져보고 이를 가능하게 하는 기반 기술로서 웹서비스와 시맨틱웹 기술 동향을 소개한다. 이들은 열린 웹 공간에서 서비스와 지식을 공유할 수 있는 표준 프로토콜과 지식 표현 체계를 제공한다. 이들 기술과 열린 네트워크 환경을 기반으로 스스로 지식과 서비스 수단을 습득하여 능력을 확장할 수 있는 소프트웨어 로봇을 구현할 수 있으리라 기대한다.

1. 현실 세계를 움직이는 소프트웨어

정보 네트워크는 일상 생활 안으로 깊숙히 들어와 우리 삶을 정보 지향적인 활동으로 채워나가고 있다. 일상 활동의 많은 부분이 각종 정보 시스템을 이용한 정보 검색과 정보 활용으로 환원되어가고 있다. 사회망(Social Network)은 웹안의 가상 현실 세계에서 활발히 구축되며 이는 현실 세계로 전이된다. 센서와 무선 통신 기술을 통해 원격지에서 일어나는 일 또는 현상을 정보의 형태로 실시간으로 감지할 수 있다. 역으로 가상 세계의 정보 상태를 변경하여 원격지에 있는 장치를 조작하고 이를 통해 현실 세계에 영향을 줄 수 있다. 이와 같은 가상 세계와 현실의 상호 연결성은 점차 더 강화되고 있다. 인터넷과 웹, 모바일 네트워크, 유비쿼터스 컴퓨팅의 등장과 급속한 보급은 가상과 현실의 상호 연결 수단 및 경로의 수를 기하급수적으로 증가시키고 있다. 결국 정보의 덩어리인 가상 세계와 현실 세계의 경계는 점점 더 흐려지고 있다. 정보 조작의 효과는 가상 세계에만 머무르지 않고 즉시 현실 세계로 전이된다. 이런 상황에서 정보 조작의 주체인 소프트웨어의 영향력은 급격히 커질 수

밖에 없다. 소프트웨어는 그 어느 때 보다 현실에 깊숙히 개입하여 현실의 문제들을 해결하고 현실 세계를 변화시키기 시작한 것이다. 사용자의 관심 주식 종목의 가격 추이를 살피며 사용자에게 매매 관련 제안을 제시 또는 직접 매매 기능을 수행하고, 사용자의 건강 상태를 지속적으로 체크하며 비상 상황 발생 시 자동으로 병원에 연락을 취하고, 사용자의 취향, 시간대, 요일 등 복잡한 컨텍스트를 고려하여 집안의 물리적 환경을 제어하는 등의 기능은 이미 소프트웨어로 할 수 있는 보편적인 일들로 여겨진다. 소프트웨어가 이러한 일을 할 수 있게 된 원동력은 현실 세계에 편재하는 정보 네트워크와 네트워크에 연결된 정보 생산 및 교환 수단들, 감지 장치들, 조작 장치들, 그리고 이들을 접근/제어할 수 있는 정보 조작 수단으로부터 나온다. 현실 세계를 움직이는 물리적 수단들이 보편화되고 이들을 네트워크를 통해 일관된 방식으로 조작할 수 있게 됨으로써 소프트웨어는 현실에 대한 보다 큰 영향력을 보유하게 된 것이다.

II. 네트워크가 부른 소프트웨어 로봇

정보 네트워크를 통해 연결성이 확대되고 현실 세계에 대한 조작 기능이 점점 더 많이 정보 네트워크에 편입됨에 따라 소프트웨어 중심의 로봇이 주요한 연구 주제로 떠오르고 있다. 이제 로봇은 바퀴나 팔을 달고 움직여야만 할 필요가 없다. 그러지 않고도 사람에게 유용한 서비스를 제공할 수 있는 길이 많이 생겼기 때문이다.

네트워크와 소프트웨어를 중심으로 한 새로운 로봇에 대한 비전은 유비쿼터스 로봇이란 이름으로 천명된 바 있다¹¹⁾. 유비쿼터스 로봇은 네트워크를 기반으로 한 여러 형태의 로봇들을 총칭

하는데, 크게 세 종류의 로봇이 존재한다. 첫째는 모바일 로봇(Mobile Robot, Mobot)이다. 모바일 로봇은 전형적인 이동체 로봇과 동일하나 네트워크를 기반으로 한 상호 작용 기능을 보유한다는 점에 있어 기존 이동 로봇과 다르다. 두 번째는 소프트웨어 로봇(Software Robot, Sobot)이다. 소프트웨어 로봇은 로봇의 특성과 기능을 보유하지만 물리적인 몸체를 갖지 않는 로봇이다. 따라서, 네트워크를 통해 물리적인 한계를 뛰어넘는 이동성을 획득한다. 세 번째는 임베디드 로봇(Embedded Robot, Embot)이다. 임베디드 로봇은 환경 안에 설치된 로봇을 말한다. 예를 들어, 지능적인 정보 처리 기능을 보유한 센서 모듈, 간단한 액추에이터들, 가전 기기 등을 임베디드 로봇으로 총칭할 수 있다. 따라서, 임베디드 로봇은 비이동 로봇(Immobile Robot)¹²⁾, 유비쿼터스 기능자¹³⁾, 지능형 건물(Intelligent Building), 디지털 가전 기기, 착용형 PC, 그 외 각종 유비쿼터스 장치 등 매우 폭넓은 영역을 아우른다. 이들 각종 유비쿼터스 로봇들은 네트워크를 통해 정보를 교환하고 상호 협력 하면서 기존의 고립형 하드웨어 중심의 로봇들이 할 수 없었던 일들을 할 수 있다.

여기서 주목할 것은 네트워크가 로봇 작동의 주요한 한 요소로 자리잡음으로써 로봇의 물리적인 작동 기능을 수행하는 몸체와 지능적인 제어 기능을 수행하는 두뇌를 분리하고자 하는 경향이 촉발되었다는 점이다. 이는 기술적인 제한 여건에 의해 자연스럽게 진행되기도 하고, 경제적인 효과를 극대화하기 위한 목적으로 적극적으로 추구되기도 한다. 대부분의 센서 모듈이나 액추에이터는 한정된 계산 자원만 가지므로 자체적으로 복잡한 계산을 수행할 수 없다. 컨텍스트 해석이나 상황 기반 서비스와 같은 복잡한 계

산은 주로 계산 능력이 풍부한 별도 모듈에서 수행하며 이 모듈들은 전문한 센서 모듈이나 액추에이터를 원격 접근 및 조작함으로써 필요한 기능을 수행한다. 자연스레 두뇌와 몸체를 분리 구현하고 네트워크를 통해 연결시키는 것이다. 한편, 로봇의 단가를 최대한 낮추어 로봇의 대중화를 실현하려는 시도들도 있다. 유비쿼터스 로봇틱 컴패니언(Ubiquitous Robotic Companion, 이하 URC)¹⁴은 이러한 측면에 있어 대표적인 사례이다. URC는 URC 클라이언트와 URC 서버로 구성된다. URC 클라이언트는 몸체를 갖는 이동 로봇으로서 자체 센서를 이용한 환경 감지, 이동 등 하위 수준의 물리적 기능을 수행한다. 예를 들면, 이동, 카메라 촬영, 음성 샘플링 등의 기능이다. URC 서버는 URC 클라이언트로부터 입수된 데이터를 접수 분석하여 판단한 후 URC 클라이언트로 제어 신호를 전송함으로써 URC 클라이언트를 제어하는 기능을 수행한다. 예를 들면, 영상 인식, 음성 인식, 지식 추론, 상황 인지, 서비스 계획 등의 기능이다. 계산 자원이 많이 필요한 고비용 작업을 서버에 배치함으로써 로봇 몸체 URC 클라이언트의 사양을 대폭 낮출 수 있으며 이는 곧 로봇의 낮은 가격으로 이어진다. 즉, 경제적으로나 기술적으로나 고성능의 네트워크가 도입되는 상황에서 주요한 로봇 모델의 하나는 몸체와 두뇌가 분리된 로봇이다.

지금까지 살펴본 바는 무엇을 시사하는가? 정보 네트워크를 기반으로 가상과 현실이 보다 더 밀접하게 통합되는 상황에 있어 소프트웨어를 중심으로 한 로봇의 중요성은 막중하다는 점이다. 따라서, 소프트웨어 중심의 로봇, 즉 소프트웨어 로봇의 정체성과 가능성에 대한 활발한 연구와 개발 활동이 필요하다.

열려있는 네트워크 환경과 각종 기술 표준들

은 로봇의 확장성을 한 차원 높일 수 있는 기반 환경을 제공한다. 다음 장부터는 소프트웨어 로봇이 자체 기능을 스스로 확장할 수 있도록 만들기 위해 필요한 기술 및 환경 요소를 따져 보고 관련 기술 동향을 살펴보고자 한다. 네트워크 기반 소프트웨어 로봇이 지닌 많은 가능성 중 하나를 살펴봄으로써 소프트웨어 중심의 로봇 연구의 중요성과 가치를 재고해 볼 수 있었으면 한다.

III. 네트워크 기반 소프트웨어 로봇 (Network based Software Robot, 이하 NSR)의 기능 확장

로봇의 기능을 확장한다는 것은 로봇이 이전에 수행할 수 없었던 기능을 이제 수행할 수 있도록 만드는 것을 뜻한다. 일반적으로 새로운 기능을 추가하기 위해서는 로봇에 탑재된 소프트웨어를 갱신하여 재탑재하거나 새로운 소프트웨어 컴포넌트를 추가해야 한다. Golog¹⁵과 같은 논리 프로그래밍 언어를 이용하여 로봇의 제어 논리를 구현한 경우라면, 선언적으로 지식을 추가함으로써 보다 손쉽게 로봇의 기능을 확장할 수 있다. 그러나, 이러한 방식의 기능 확장은 로봇 개발자나 관리자에 의해 완전히 장악된 체계에 의지하므로 기능 확장 통로는 개발/관리자에 대해 닫혀 있다고 볼 수 있다. 즉, 로봇이 사전 약속된 절차 또는 자율적인 판단을 통해 자동으로 기능을 확장할 수 없다는 뜻이다.

NSR은 네트워크에 참여하여 각종 원격 장치나 웹을 통해 정보를 수집할 수 있으므로 이전의 고립된 로봇에 비해 상대적으로 더 많은 기능 확장 가능성을 지니고 있다. 즉, 로봇 개발/관리자에 대해 닫혀있지 않은 기능 확장 체계 열린 기능 확장 체계를 확보할 수 있게 되었다. 그러면,

열린 기능 확장을 가능하게 하는 기술 요소는 무엇인가?

첫째, 네트워크이다. 앞서 살펴 본 바와 같이 네트워크는 소프트웨어 로봇에게 독자적인 가치를 부여한다. 네트워크는 NSR이 새로운 기능을 습득하기 위해 타 NSR, 정보 제공자, 장치 등과 접속하는 매개체이므로 가장 근본적인 기술 요소이다.

둘째, 표준 프로토콜이다. NSR이 네트워크를 통해 제 3의 개체와 상호작용하기 위해서는 다양한 프로토콜이 필요하다. 센서의 값을 읽어오거나 액추에이터 제어 신호를 전달하는 등의 단순한 제어 기능, 서비스 제공에 필요한 가전 제어 서비스를 찾거나 사용자 질의에 대한 답을 제공하기 위한 정보 제공자를 찾는 등의 탐색 기능, 다른 NSR들과 정보를 주고 받기 위한 정보 교환 기능 등 다양한 네트워크 기반 기능을 수행하기 위함이다. 프로토콜이 표준화되면 NSR은 사전에 설정해 놓지 않은 개체들과 실패 없이 통신할 수 있게 된다. 즉, NSR의 상호 작용 공간은 네트워크 내에서 열려있게 된다. 이는 전술한 바와 같은 열린 기능 확장을 위한 필요 조건이다.

셋째, 표준 지식 표현 언어이다. 지식 표현 언어는 선언적으로 개체의 특성과 기능을 기술할 수 있는 기반 체계를 제공한다. 지식 표현 언어로 기술한 지식은 그 언어의 의미론을 구현한 추론 시스템을 통해 처리된다. 지식 표현은 선언적이므로 사람에게 직관적이고, 스크립트 언어처럼 인터프리터인 추론 엔진을 통해 처리되므로 보다 손쉽게, 실행 중이라도 변경이 가능하다는 점에서 NSR의 기능 확장을 구현하기에 적합한

도구이다. 지식 표현 언어를 표준화하고 이 언어의 구문론 및 의미론을 지원하는 추론 엔진을 사용한다면, NSR은 이 언어로 기술된 다양한 지식을 입수하여 처리할 수 있게 된다. 표준 지식 표현 언어를 이용하여 NSR에게 새로운 상황 해석 방법, 서비스 방법 등을 제공할 수 있는 기반이 마련되는 셈이다.

넷째, 표준 기호 체계이다. 표준 기호 체계는 미지의 개체들 간에 상호 호환성 있는 의미 체계를 공유할 수 있게 한다. NSR이 A라는 효과를 지닌 서비스를 찾고 있고 어떤 미지의 서비스 S가 A라는 효과를 지녔다는 정보가 공개된 경우, 이 두 A가 상호 동일한 의미를 지녀야 서비스 탐색이 의미 있는 결과를 낼 수 있다. 또, NSR이 S를 실행하여 받은 결과값이 상호 공유되는 의미를 지닌 기호로 표현된 데이터가 아니라면 S를 실행한 의미가 상실될 수도 있다. 이와 같이 NSR이 미지의 어떤 개체와 표준 프로토콜을 통해 성공적으로 통신을 할 수 있게 되었다손 치더라도 NSR과 그 개체가 주고 받는 데이터의 의미가 상호간에 동일하지 않다면 NSR은 스스로 지식을 습득하거나 서비스를 수행하는데 실패할 것이다.

다섯째, 표준 지식 표현 체계이다. 표준 기호 체계는 개개 기호의 의미를 공유할 수 있는 기반을 제공하나, 그 기호들을 이용하여 기술한 지식 구조체를 공유할 수 있도록 만들어 주지는 못한다. 여기서 지식 구조체란 특정 영역(Domain)의 지식을 표현하는데 사용하는 스키마(Schema)를 가리킨다. 스키마는 영역 별로, 그리고 지식의 용도 별로 각각 정의되어야 한다. 공유된 스키마를 기반으로 저작된 공개 지식은 그 스키마를 이해

하는 NSR에 의해 습득 및 활용될 수 있다.

상기한 요소들을 기반으로 가능한 NSR의 열린 기능 확장의 대표적인 예는 다음과 같다.

(1) 서비스 탐색 및 실행 : NSR은 센서 정보 레지스트리에 접근하여 미지 센서의 출력 값의 의미, 설치 위치, 작동 시간, 접근 방법을 알아낼 수 있다. NSR은 이를 기반으로 이전에 접근할 수 없었던 센서 장치에 접근하여, 이전에 이해할 수 없었던 센서 값을 읽고 해석하여 상황 지식 구축에 활용할 수 있다.

(2) 지식 확장 : NSR은 사용자의 명령에 포함된 단어 중 자체 보유한 지식베이스에 포함되어 있지 않은 단어가 존재하는 경우 웹에 공개된 단어 사전 서비스에 접근하여 단어의 의미를 가리키는 기호를 검색/입수하여 명령 처리에 활용할 수 있다. 명령 처리를 위한 관련 서비스 지식 또한 웹에 공개된 지식베이스를 입수하여 활용 가능하다. 이때, 지식베이스를 기술한 언어 및 어휘는 전술한 바와 같이 표준화되어 있어야 한다.

이와 같이 NSR은 감지할 수 있는 상황의 종류를 자동으로 확장할 수도 있고 제공할 수 있는 서비스의 수를 늘려나갈 수도 있다.

IV. 웹서비스와 시맨틱웹

앞서 설명한 바와 같은 NSR의 열린 기능 확장을 실현하기 위한 주요 기술로 웹서비스와 시맨틱웹을 들 수 있다.

1. 웹서비스(Web Service)

웹서비스를 이용하면 센서, 액추에이터, 정보 서비스 등 각종 이질적 서비스 제공자들을 동일한 구조의 외부 인터페이스로 감쌀 수 있다. NSR은 동일한 접근 방식으로 이들 서비스 제공자들에 접근할 수 있게 된다. 웹서비스 관련 표준은 서비스 기술(Service Description), 서비스 발견(Service Discovery), 서비스 실행(Service Execution)을 위한 언어 및 프로토콜로 구성되는데, 주요 표준은 SOAP(Simple Object Access Protocol), WSDL(Web Service Description Language)^[6], UDDI(Universal Description, Discovery and Integration) 등이다.

2. 시맨틱웹(Semantic Web)

시맨틱웹은 웹에 편재한 데이터에 구조와 의미를 부여할 수 있는 표준화된 수단을 제공한다. W3C를 중심으로 전개된 표준화 작업의 결과로 데이터의 의미와 구조를 표현할 수 있는 표준 언어인 RDF(Resource Description Language)와 OWL(Web Ontology Language)이 권고안으로 제정되었다. RDF와 OWL은 NSR의 열린 기능 확장 실현의 근간인 표준 지식 표현 언어에 해당한다. RDF와 OWL은 NSR의 표준 기호 체계와 표준 지식 표현 체계를 구축하는데 필수적인 도구이다. 앞으로 보다 많은 서비스와 웹 데이터가 RDF와 OWL로 기술될 것으로 예상되므로 NSR의 기능 확장 및 활용의 폭은 더욱 확대될 것이다.

3. 시맨틱 웹서비스(Semantic Web Service)

WSDL은 웹서비스의 구조적인 면만을 기술할 수 있다. 즉, WSDL로 웹서비스 A가 입력값으로 정수 두 개를 받고 출력값으로 실수 하나를 내준다는 정보를 기술할 수는 있지만, A가 나눗셈을 수행한다는 정보를 기술할 수는 없다. NSR이 서비스의 효과와 입출력값의 의미를 이해하여 서비스의 용도를 파악하려면 서비스의 구조 정보 뿐만 아니라 서비스의 의미를 알 수 있어야 한다. 서비스에 대한 설명 및 서비스의 입출력 데이터를 OWL과 같은 시맨틱웹의 데이터 표현 언어로 기술함으로써 이러한 요구 사항을 만족시킨 새로운 형태의 웹서비스를 시맨틱 웹서비스라 부른다.

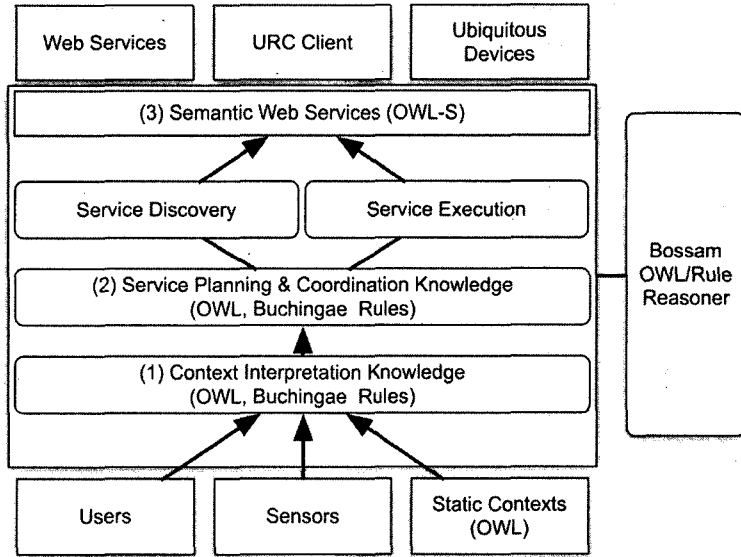
서비스의 의미를 기술하는 지식 체계로서 대표적인 것은 OWL S(OWL Service)이다. OWL S는 서비스의 입력(Input), 출력(Output), 전건(Pre condition), 효과(Effect)를 OWL로 기술할 수 있는 지식 체계(Schema)를 제공한다. OWL의 표현력 상 한계로 인해 OWL을 기반으로 한 OWL S는 서비스의 전건 및 후건(Post condition)을 논리식으로 기술할 수 없다. 이로 인해 OWL S로 기술한 서비스의 의미가 부정확하게 되는 문제가 발생한다. 이를 개선할 목적으로 전건이나 후건에 KIF(Knowledge Interchange Format) 구문이나 여러 개의 항을 가지는 술어(Predicate) 및 함수(Function)를 삽입할 수 있도록 확장된 OWL S 1.1, OWL S 1.2가 정의되었다. 보다 최근에는 OWL 언어를 탈피하여 일차술어논리(First Order Logic), 논리 프로그래밍(Logic Programming), 생성 규칙(Production Rules), 트랜잭션 논리(Transaction

Logic) 등 풍부한 표현력을 수용한 새로운 서비스 기술 언어들이 정의되고 있기도 하다. 미국의 NIST를 중심으로 만들어진 SWSL(Semantic Web Service Language), 유럽의 주요 시맨틱웹 연구 단체인 DERI를 중심으로 제정된 WSML(Web Service Modeling Language)이 대표적이다. SWSL과 WSML로 양극화된 시맨틱 웹서비스 표준 활동을 아우르는 단체로서 SWSI(Semantic Web Services Initiative)⁹⁾가 설립되어 활동 중이기도 하다.

V. NSR 기능 확장 체계 구현 사례 : URC용 지능형 서비스 플랫폼

URC용 지능형 서비스 플랫폼(URC Service Platform, 이하 URCSP)은 URC 클라이언트를 비롯한 각종 유비쿼터스 서비스들을 통해 상황을 감지하여 이해하고 사용자에게 상황에 맞는 적절한 서비스를 제공하는 서비스 플랫폼으로서 웹서비스, 시맨틱웹, 지식처리 기술을 기반으로 하고 있으며, 초보적 수준의 열린 기능 확장 체계를 갖추고 있다. <그림 1>은 URCSP를 구성하는 지식베이스를 중심으로 한 URCSP의 구조이다.

그림에서 보듯 URCSP의 주요 기능은 지식 표현 언어인 OWL과 Buchingae로 기술되었으며, 이는 Bossam 추론 엔진을 통해 처리된다¹⁰⁾. URCSP의 구조에서 열린 기능 확장이 가능한 부분은 크게 세 부분으로, 각각 (1) 사용자 명령 및 상황을 해석하는 지식, (2) 서비스 계획 및 제어 지식, 그리고 (3) 서비스 기술 지식이다. 그러나, (1)과 (2)에 대해서는 자동화된 기능 확장 체계를 구현하지 않은 상황이다. (3)은 각종 서비스를 OWL S로 기술해 놓은 지식으로서 시맨틱



〈그림 1〉 URCSP의 구조

서비스 레지스트리에 저장된다. 누구든 새로운 서비스를 제공하고 싶으면 서비스의 기능 및 특성을 OWL S를 기반으로 작성한 후 레지스트리에 등록하면 된다. 레지스트리에 등록된 서비스는 URCSP의 서비스 탐색(Discovery) 대상이 된다. 즉, URCSP는 미지의 서비스를 찾아 이용할 수 있으므로, 열린 기능 확장이 가능하다고 할 수 있는 것이다. 그러나, 이 경우에도 서비스를 기술한 어휘와 (2), (1)의 호환성이 보장되어야만 기능 확장이 성공한다. 근본적으로는 (1)과 (2)를 동적으로 확장할 수 있어야만 (3)의 확장이 본격적으로 의미를 갖게 되는 것이다.

VI. 결 론

정보 네트워크를 기반으로 한 새로운 환경 하에서 로봇의 몸체와 두뇌는 분리되는 경향을 띠고 있다. 고도의 지능적 계산을 담당하는 두뇌가

몸체로부터 분리됨으로써 두뇌에 해당하는 소프트웨어 로봇은 보다 더 큰 중요성을 띠게 되었다. 네트워크가 제공하는 연결성과 개방성의 장점을 최대한 활용하여 소프트웨어 로봇의 기능을 고도화할 수 있는 한 기법으로서 웹서비스와 시맨틱웹을 기반으로 한 열린 기능 확장 체계를 소개했다. URCSP는 초보적인 수준의 열린 기능 확장 능력을 보유하고 있는 네트워크 기반 소프트웨어 로봇으로서 향후 기능 확장을 통해 보다 높은 수준의 기능 확장 능력을 보유하도록 만들 계획이다. 표준화된 지식 처리 기술을 기반으로 고도화된 기능 확장 능력을 구현한다면 소프트웨어 로봇도 사람처럼 세계와 상호작용하며 지식과 기술의 지평을 넓혀 갈 수 있지 않을까?

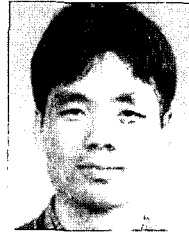
=====참고 문헌=====

[1] Jong Whan Kim, Yong Duk Kim, and Kang Hee

Lee, "The Third Generation of Robotics: Ubiquitous Robot," Proceedings of the 2nd International Conference on Autonomous Robots and Agents, December 13 15, 2004 Palmerston North, New Zealand.

- [2] B. C. Williams and P. P. Nayak, "Immobile Robots: AI in the New Millennium," AI Magazine, pp.17 34, Fall 1996.
- [3] Bong Keun Kim et al., "Design of Ubiquitous Functions for Networked Robots in the Informative Spaces," Proceedings of the 2nd International Conference on Ubiquitous Robots and Ambient Intelligence, November 2005, South Korea.
- [4] Sang Rok, Oh, "IT Based Intelligent Service Robot," Proc. First NSF PI Workshop on Robotics and Computer Vision (RCV'03) Invited talk, Las Vegas, Oct. 2003.
- [5] Levesque, H., Reiter, R., Lesperance, Y., Lin, F., and Scherl, R., Golog: A logic programming language for dynamic domains, Journal of Logic Programming, 1997.
- [6] David Booth, Canyon Kevin Liu, "Web Services Description Language (WSDL) Version 2.0 Part 0 : Primer," W3C Candidate Recommendation 6 January 2006.
- [7] Frank Manola, Eric Miller, "RDF Primer," W3C Recommendation 10 February 2004.
- [8] Mike Dean, Guus Schreiber, "OWL Web Ontology Language Reference, W3C Recommendation, 10 February 2004.
- [9] Semantic Web Services Initiative (<http://www.swsi.org/>)
- [10] Minsu Jang, Joo chan Sohn, "Bossam : an extended rule engine for the web", Proceedings of RuleML 2004 (LNCS Vol. 3323), Nov. 8, 2004.

저자소개



장민수

1994년 서강대학교 전산학 석사 졸업
 1994년-1999년 현대전자산업(주) 통신연구소 재직
 1999년-2003년 ETRI 전자거래연구부 선임연구원
 2003년-현 재 ETRI 지능형로봇연구단 선임연구원
 주관심분야 지능/추론 시스템, 시맨틱웹, 인공 생명, 디자인 패턴



손주찬

1990년 한국외국어대학교 경영정보학 석사 졸업
 1991년-현 재 한국전자통신연구원 재직
 현 재 ETRI 지능로봇연구단 지식및추론연구팀장
 주관심분야 로봇 서비스 아키텍처, 감성형 로봇



조영조

1989년-1998년 KIST 선임연구원 (분산제어시스템설계)
 1993년-1994년 일본통신청 기계기술연구소 로봇공학부 초빙연구원 (가상현실기반 텔레로봇 제어)
 1997년 Univ. of Massachusetts at Amherst 초빙연구원 (로봇 제어구조 설계)
 1998년-2001년 KIST 책임연구원 (로봇지능제어)
 2001년-2004년 아이콘트롤스 기술연구소장/상무 (홈 게이트웨이 및 빌딩용 DDC 개발)
 2004년-현 재 ETRI 지능형로봇단장