능동형 서비스를 제공하는 CAMUS 시스템에 관한 설계 및 구현

정인철, 조준면, 이강우, 김형선
한국전자통신연구원
e-mail : { jic, jmjoo, kwlee, hskim}@etri.re.kr

Design and Implementation of the CAMUS system based Proactive Service

In-Cheol Jung, Jun-Myun Joo, Kang-Woo Lee and HyungSun Kim
Intelligent Robot Research Division
Electronics and Telecommunications Research Institute
161 Kajong-Dong, Yusong-Gu, Taejon, 305-350, Korea
{ jic, jmjoo, kwlee, hskim}@etri.re.kr

요 약

유비쿼터스 환경에서 사용자에게 유용한 서비스를 제공하기 위해서는 기능을 명시적으로 제공하기 보다는 사용자의 요구에 따라서 능동형으로 제공하는 기능이 필요하다. 이러한 능동형 서비스는 알아서 서비스를 수행하는 기능으로 CAMUS 시스템에서는 서비스 탐색 부분에서 사용된다. 즉 시스템의 상황에 따라서 메시지 시스템을 구현하는 경우에 자원의 형태에 따라서 영상 메세징, TTS 기능을 이용한 메세징, Text 기반 메세징 시스템으로 자동적으로 처리할 수 있다. 이를 위해 CAMUS 서버에는 SAM (Service Agent Manager) 과의 통신을 통해 환경 내에 존재하는 센서와 장치들 (Service Agent)을 관리 및 제어한다. 이러한 Service Agent Manager 는 여러 다양한 환경에 설치되어 환경 내에 위치한 다양한 센서로부터 정보를 얻고 그 정보를 CAMUS 메인 서버에 전달하는 한편 CAMUS 메인 서버로부터 실행명령을 받아 환경 내 장치를 제어하는 역할을 한다. 이러한 Service Agent Manager 는 임의의 공간 내에 설치될 수 있으며 로봇단말이나 개인 휴대단말 등에도 설치될 수 있다. 이 논문에서는 SAM (Service Agent Manager) 과 CAMUS 서버에서 원하는 서비스를 탐색하는 방법에 대한 내용을 기술하였다.

1. 서론

URC 는 네트워크 기반의 지능형 서비스 로봇의 새로운 개념으로, 최소한의 기능을 갖는 로봇이 외부 센서와 중앙 서버를 활용하여 보다 지능화된 서비스를 제공한다는 것이 매우 중요하다. 즉 사용자가 로봇에게 서비스를 요청할 때 현재의 상황을 이해하여 그 상황에 맞는 서비스를 제공하거나 사용자의 요청이 없는 경우에도 필요한 시점에 필요한 정보와 서비스를 지능적이고 능동적으로 제공할 수 있다.

CAMUS(Context-Aware Middleware for URC systems) 는 URC 개념의 구현을 위해 요구되는 기반구조 중 '소프트웨어 인프라'에 해당하는 것으로서 중앙 서버와 외부 센서 및 기기 (디바이스), 중앙 서버와 응용 서비스 간의 통신과 상호작용을 가능하게 하고 관리 제어하는 목적을 위해 개발된 상황인식 미들웨어 이다. CAMUS 는 사용자가 로봇 또는 컴퓨터에게 정보나 서비스를 요청할 때, 또는 사용자의 요청이 없더라도 로봇 또는 컴퓨터가 현재의 상황을 인식하여 그 상황에 맞는 정보와 서비스를 제공할 수 있도록 지원하는 소프트웨어 프레임워크이다.

O 환경으로부터의 정보획득 및 환경 내 기기 지원
  - 환경 내 센서로 부터의 정보를 획득하고 가공할 수 있는 프레임워크를 제공하며, 환경 내 장치를 동적으로 탐색하고 탐색된 장치를 제어할 수 있는 수단을 제공

O 사용자 성향 및 선호도 관리 지원
  - 사용자에 대한 성향 및 선호도에 대한 정보를 학습할 수 있는 소프트웨어 엔진을 제공

O 상황정보 관리 및 접근 지원

- 로봇 또는 소프트웨어 로봇에서 환경과 사용자에 대한 상황정보에 접근할 수 있는 수단을 제공

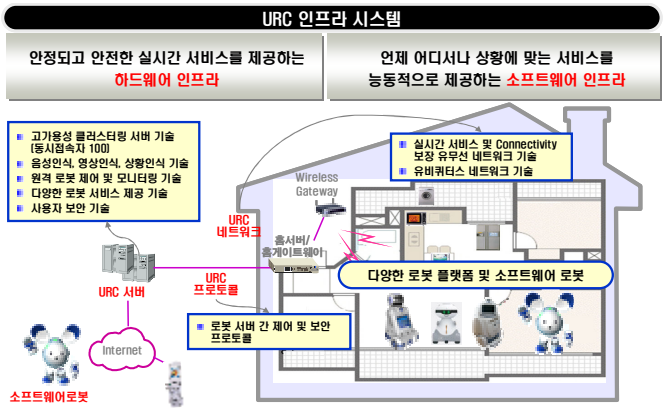

Figure 1. URC 인프라 시스템 구성도

O 상황 기반 응용 개발 지원
- 상황 정보 기반의 응용 프로그램을 개발 할 수 있도록 지원하며, 개발된 상황기반 응용이 구동할 수 있는 소프트웨어 엔진 기능을 제공

O 응용 시스템 연동 지원
- 음성인식, 영상인식, 행위 제어 등 로봇 또는 소프트웨어 로봇이 갖는 기본적인 기능으로서 서버 측에서 제공되어야 할 기본 기능

2. CAMUS System Overview

CAMUS 시스템은 물리적으로 탑재되는 위치에 따라서 CAMUS 메인 서버와 Service Agent Manager 로 구분된다.

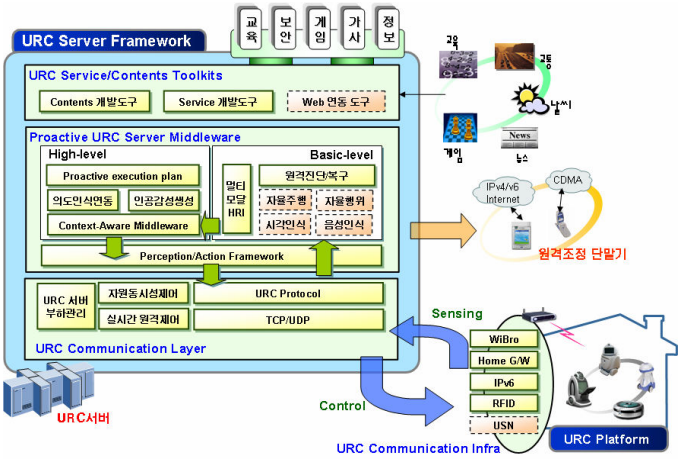

Figure 2. CAMUS 시스템 구성도

O 로봇 서버간 통신
로봇과 서버간의 실시간 통신 안정성을 보장하는 유무선 네트워크 실시간 모니터링 시스템을 통하여 QoS 점검 및 통신 다중화 방법을 채용한 통신단절 대응기술 개발

O 능동형 URC 서버 프레임워크 기술
- 유비쿼터스 네트워크 환경에서 상황을 인식하여 언제 어디서나 능동형 서비스를 제공할 수 있는 분산 컴퓨팅 구조를 기반으로 함.
- 분산 환경에서 가용한 센서를 활용하여 다양한 상황정보를 수집하고, 상황지식 병합 및 관리를 위한 기술 구현
- 기존의 시각인식, 음성인식의 기술수준을 감안하여 사용자 행동모델 컨텍스트 정보를 이용하여 사용자의 행위와 주변 환경 변화에 따라 사용자의 의도를 파악하여 상황인식 기반의 적응적인 로봇 행위 제어 및 서비스가 가능

O URC 를 위한 인공감성 시스템
기존의 인공감성 기술은 복잡한 구조의 감성모델 및 대규모 사용자 접속지원 등 URC 플랫폼의 정보처리 자원의 제약을 극복하기 위해 URC 서버와 연계하여 분산처리 구조로 개발

3. CAMUS Service Agent Manager

3.1 Service Agent

Service Agent 는 CAMUS 와의 통신을 통해 물리적인 공간상에 위치한 센서나 장치의 기능을 수행할 수 있도록 만들어진 소프트웨어 모듈로써 Service Agent Manager 에서 수행된다. Service Agent 는 CAMUS 와 연동되는 외부 장치 및 시스템에 대한 인터페이스를 구현한 소프트웨어 모듈이라고 할 수 있다.

Service Agent 는 서비스 에이젼트 인터페이스를 통해 서비스 에이젼트가 제공하는 속성 및 동작을 외부 시스템에 노출한다. 이 들 인터페이스에 대한 구현은 실제 센서나 장치 측에서 이루어지고 실제 CAMUS Task 등의 외부 시스템에서의 접근은 CAMUS 통신 프레임워크에 의해 이루어 진다. 즉 서비스 에이젼트 인터페이스는 외부 시스템 과의 약정을 선언하는 것이고, 서비스 에이젼트의 구현은 이 인터페이스에 대한 실제 의미를 기술하여 동작하게 하는 것이며, 서비스 에이젼트에 대한 접근은 CAMUS 통신 프레임워크로 이루어진다.

예를 들면 사용자 위치를 인식하기 위한 Service Agent 가 있고, 이 에이젼트 인터페이스가 사용자 id 와 location 정보를 넘겨주는 함수를 갖는다고 하자, 이 경우에 실제 사용자 위치 인식을 위한 센서는 다양하게 존재할 수 있다. 만약 환경에 RFID 센서가 있다고 하면 이 RFID 센서로부터 제공하는 Device Driver 를 이용하여 위의 인터페이스를 실제 프로그램으로 구현하게 된다. 이와 같이 구현된 프로그램이 결국 사용자의 위치 정보를 제공하는 서비스 에이젼트가 된다.

한편, 영상센서가 있다고 하자 이 영상센서로부터 사용자의 위치정보를 얻고자 한다면 마찬가지로 위치센서 인터페이스를 영상센서로 구현한 프로그램 모듈

이 위치 정보를 제공하는 서비스 에이전트가 된다. CAMUS 타스크 쪽에서는 실제 환경 내 물리적 센서가 RFID 센서인지 영상센서 인지를 신경 쓰지 않고 사용자의 위치 인식 서비스 에이전트 인터페이스를 호출함으로써 원하는 작업을 하게 된다.

Service Agent 는 Service Agent Manager 상에서 수행되며, CAMUS Task 에 의해 호출된다.

3.2 Service Agent Manager

Service Agent Manager는 환경에 설치되어 환경 내 다양한 센서로부터 정보를 얻고 그 정보를 CAMUS Main Server로 부터의 실행명령을 받아 환경 내 장치를 제어하는 역할을 한다.

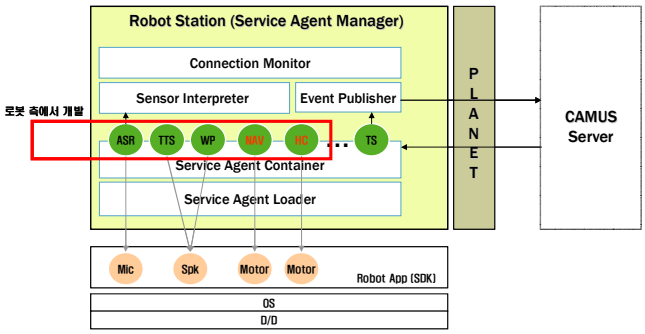

Figure 4. Service Agent Manager 구성

Service Agent Manager 는 다양한 서비스 Agent 들이 탑재할 수 있는 Service Agent Container 를 제공한다. 서비스 Agent 들은 Service Agent Loader 에 의해 Service Agent Container 에 탑재되며 이렇게 탑재된 서비스 Agent 들은 CAMUS 통신 프레임워크를 통해 CAMUS Main Sever 와 통신하게 된다.

서비스 Agent 들로 부터의 센싱되는 정보는 Sensor Interpreter 를 거치면서 정제(Filtering)되거나 합성(Aggregation)된다. 이러한 과정을 통하여 최종 정보는 Event Publisher 를 통해 CAMUS Main Server 에 Event 를 발생시킨다.

CAMUS Main Server 로 부터의 서비스 Agent 호출은 CAMUS 통신 프레임워크를 통해 직접 호출될 수 있으며 Service Agent Invocator 를 통해 호출될 수 있다.

4. CAMUS

4.1 CAMUS 서버 구조

CAMUS Server 는 여러 Service Agent Manager 로부터 전달되는 정보를 수집하고 운용하며, 응용에서의 접근을 지원하고, 서버 기반 로봇 응용 개발에 필요한 제반 기능을 제공하는 프레임워크이다. CAMUS Server 에서는 환경 정보, 사용자 정보 및 Task 정보를 관리하고 있으며, 상황 변화에 따라 발생한 이벤트를 응용시스템으로 전달하여 그 상황에 적합한 행위를 취할 수 있도록 한다. 또한 음성인식, 영상인식, 행위제어 등 로봇 또는 소프트웨어 로봇의 기본적인 서비스와 연동할 수 있는 서비스 컨테이너를 제공한다. 그러나 CAMUS Server 자체에는 음성인식, 영상인식 등의 기본 서비스를 포함하고 있지는 않다. Figure5 는 CAMUS Server 의 개념적 구조도 이다.

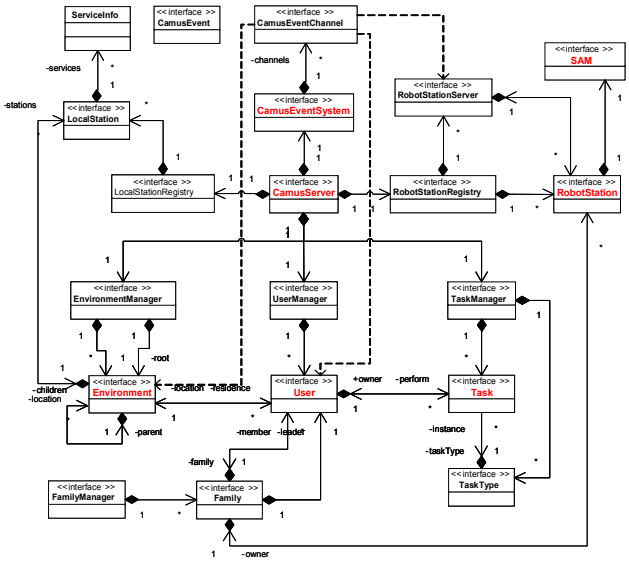

Figure 5. CAMUS Server 구조

CAMUS 에서는 실제 물리 공간을 추상화한 가상 공간을 *환경(Environment)*으로 정의한다. 환경은 공간영역 (Place)의 포함 관계에 따라 계층적인 구조를 갖게 되며, 각 영역 내에는 다양한 센서나 기기 등의 전산자원을 갖는다.
실제 물리공간 내 존재하는 각종 센서 나 기기 등의 물리자원은 가상공간에서 그 기능을 수행할 수 있도록 전산자원으로 만들어져야 한다. 앞 장에서 언급한 바와 같이 물리공간 내 다양한 자원의 물리적 기능을 가상 공간 (CAMUS 환경)에서 수행할 수 있는 컴퓨터 프로그램 모듈을 Service Agent 라고 정의한다.
물리 공간에서의 *사용자(User)*는 실제 물리객체와의 상호작용을 통해 일을 하며, CAMUS 에서 제공하는 서비스의 주체가 된다. 실세계의 사람은 CAMUS 에서 고유의 식별자, 프로파일 정보, 선호사항 정보 등을 갖는 가상공간 상의 사용자로 매핑된다.
Task 는 사용자의 요청에 의해서 또는 사용자 요청이 없더라도 환경이나 사용자의 상황에 따라 수행해야 할 일련의 행위의 집합으로서, CAMUS 에서의 작업 수행 단위이다.
Context 는 특정한 상황에 맞게 Task 를 수행함에 있어서 필요한 제반 정보 및 지식을 의미한다. 이러한 Context 는 *환경 관리자(Environment Manager), 사용자 관리자(User Manager), 작업 관리자(Task Manager), 로봇 관리자 (Robot Station Registry)* 등에서 관리된다.

4.2 CAMUS 서버기능

CAMUS Server 는 Event System, Knowledge Manager, Environment Manager, Robot Manager, User Manager 및 Task Manager 등의 컴포넌트로 이루어져 있다. 각각의 컴포넌트들은 Event System 을 통해 메시지를 주고 받게 되며, Service Agent Manager 로부터의 메시지도 이벤트로서 전달된다. 따라서 각각의 컴포넌트의 기능에 대해 기술한다.

O Event System
Event System 은 CAMUS 시스템의 각 컴포넌트 간 메시지 교환을 담당한다. 특히, 분산된 여러 환경에 설치된 Service Agent Manager 들로부터 중앙 서버(CAMUS Server)로 전달되는 이벤트를 중앙 서버 내에서 실행되는 상황 관리자와 작업 관리자에게 배포하여 관심을 가지고 있는 개별 작업(CAMUS Task)이 상황 변화를 인지할 수 있도록 하며, 또한 상황 모델을 업데이트하도록 한다

O Knowledge Manager
CAMUS 지식 관리 시스템은 URC 응용 실행에 필요한 지식 및 정보를 저장/관리하고 URC 응용이 통일된 방식으로 지식 베이스에 접근하여 필요한 지식과 정보를 쉽게 이용할 수 있도록 지원하는 시스템이다. CAMUS 지식 관리 시스템은 기술논리 (Description Logic) 기반의 지식 모델 즉, 온톨로지를 기반한 지식 표현 및 관리를 전제한다.

O Environment Manager
CAMUS Server 에서 환경은 사용자가 위치한 공간을 의미한다. 환경은 트리 형태의 계층구조를 가지며, 각각의 환경은 유일한 식별자 LUID(Locally Unique Identifier)를 갖는다. 예를 들어 아파트 단지 내 한 가구를 대상으로 환경을 모델링 한다면, 다음과 같다. 이 때, LUID 는 시스템 전역에 거쳐 유일한 식별자가 아닌 상위 환경에 직접 포함된 환경 내에서 유일하다. 따라서 실제 전역적으로 유일한 식별자는 루트 노드로부터 시작하여 지정된 환경까지의 LUID 를 구분자(/)를 통해 문자열로 연결한 값이 된다.

O Robot Station Manager
CAMUS 시스템과 연동되는 로봇은 서버 상에 각각에 대응되는 가상 로봇 형태로 등록되며 관리된다. 로봇에 탑재된 Service Agent Manager 인 Robot Station 은 그에 대응하는 서버 측 Robot Station 을 갖는다. 또한 각 로봇은 유형 별 또는 지역 별 등 관리 목적에 따라 구분된 서버 측의 Robot Station Server 에 등록되며 관리된다. 로봇 관리자는 로봇을 등록하고 등록된 로봇 정보를 관리하는 모듈이다.

O User Manager
사용자 관리자는 CAMUS Server 에 등록된 사용자 정보를 관리한다. 사용자를 추가, 삭제할 수 있으며, 사용자 식별자, 사용자 이름, 사용자의 Tag 식별자 등을 통해 사용자 검색을 할 수 있다.

O Task Manager
작업 관리자는 개별 CAMUS Task 를 기동시키고 수행중인 Task 프로세스를 관리 또는 제어하는 역할을 한다.
CAMUS 모델에서는 하나의 물리적인 환경 내에서도 여러 Task 가 실행되는 것을 가정한다. 예를 들어, 한 가정 환경에서 온도 조절 Task 와 정보 컨텐츠 제공 Task 가 동시에 수행될 수 있다. 더 나아가 동일한 Task 가 여러 개 수행될 수도 있다. CAMUS 모델에서는 서비스의 개인화를 위해 각 Task 에는 소유자가 있게 되고 소유자의 요구나 선호에 따라 동일한 Task 일지라도 다른 서비스를 지능적이고 능동적으로 제공해야 하기 때문이다.

5. 서비스 시나리오

학습 시나리오
1. 사용자가 기상을 한다. 2. URC 로봇은 사용자에게 다가가서 아침인사를 한다. 3. 사용자가 “경제 뉴스 알려줘” 요청하고 URC 로봇을 통해 경제 뉴스를 듣는다.
능동적인 서비스 추천 시나리오
1. 사용자가 평소보다 늦게 기상한다. 2. URC 로봇은 사용자에게 다가가면서 “늦었습니다. 서두르세요. 뉴스 알려드릴까요 아니면 교통정보 알려드릴까요?” 라고 물어본다. 3. 사용자가 “뉴스”라고 대답하면 로봇은 경제 뉴스 컨텐츠를 읽어준다.

O 사용자는 URC 로봇이 제공 가능한 서비스를 음성명령이나 리모콘을 통해 요청하고 URC 로봇은 네트워크에 연결된 컨텐츠 서버로부터 사용자가 요청한 서비스 정보를 얻어와서 사용자에게 제공한다.

O 지능화된 URC 로봇은 사용자가 사용한 서비스 패턴, 센서, 그리고 디바이스에 의해 발생되는 상태 값을 이용하여 상황정보를 학습한 후에 사용자에게 개인화된 선호 서비스 및 상황에 맞는 서비스를 능동적으로 제공한다.

O 학습된 어떤 상황이 발생하면 사용자 패턴 학습 시스템은 학습된 정보를 참조하여 URC 로봇을 통해 해당 사용자에게 필요한 서비스를 능동적으로 추천한다.

O 사용자는 추천된 서비스에 대한 피드백을 주고 이는 다시 사용자 패턴 학습 시스템 반영되어 학습 정

보를 갱신하는데 사용된다.

6. 결론

유비쿼터스 환경에서 능동형 서비스는 알아서 서비스를 수행하는 기능으로 CAMUS 시스템에서는 서비스 탐색 부분에서 사용된다. 즉 시스템의 상황에 따라서 메시지 시스템을 구현하는 경우에 자원의 형태에 따라서 영상 메세징, TTS 기능을 이용한 메세징, Text 기반 메세징 시스템으로 자동적으로 처리할 수 있다. 이를 위해 CAMUS 서버에는 SAM (Service Agent Manager) 과의 통신을 통해 환경 내에 존재하는 센서와 장치들 (Service Agent)을 관리 및 제어한다. 이러한 Service Agent Manager 는 여러 다양한 환경에 설치되어 환경 내에 위치한 다양한 센서로부터 정보를 얻고 그 정보를 CAMUS 메인 서버에 전달하는 한편 CAMUS 메인 서버로부터 실행명령을 받아 환경 내 장치를 제어하는 역할을 한다. 이러한 Service Agent Manager 는 임의의 공간 내에 설치될 수 있으며 로봇 단말이나 개인 휴대단말 등에도 설치될 수 있다.

참고문헌

[1] H. Kim, Y.-J. Cho and S.-R. Oh, " CAMUS: A Middleware Supporting Context-aware Services for Network-based Robots", IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO) 2005.

[2] OSGi Alliance, OSGi Service Platform The OSGi Alliance Release 3, IOS Press.

[3] C. Lee, D. Nordstedt, and S. Helal, "Enabling Smart Spaces with OSGi," Pervasive Computing,IEEE, vol. 2, pp. 89-94, 2003.

[4] http://www.caucho.com/hessian/

[5] H. Elzabadani, A. Helal, B. Abdulrazak and E. Jansen, "Self Sensing Spaces: Smart Plugs for Smart Environments ". ICOST 2005, to be held in Montreal, Canada, July 2005.

[6] Zigor Salvador, Alberto Lafuente, and Mikel Larrea, “Jini as a platform for ubiquitous computing”, Proceedings of the Simposio sobre Computación Ubicua e Inteligencia Ambiental, UCAmI *2005*, Granada, Spain, Sep 2005.

[7] Zigor Salvador, Raúl Jimeno, Alberto Lafuente, Mikel Larrea, and Julio Abascal, “Architectures for ubiquitous environments”, Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob *2005*, Montreal, Canada, Aug 2005.