

# CAMUS 시스템에서의 효율적인 Service Agent Manager 관리방법에 관한 연구

정인철, 서영호, 이강우, 김현  
한국전자통신연구원

e-mail : { jic, yhsuh, kwlee, hyunkim }@etri.re.kr

## Efficient Management of the Service Agent Manager in CAMUS

In-Cheol Jung, Young-Ho Suh, Kang-Woo Lee and Hyun Kim  
Intelligent Robot Research Division  
Electronics and Telecommunications Research Institute  
161 Kajong-Dong, Yusong-Gu, Taejon, 305-350, Korea  
{ jic, yhsuh, kwlee, hyunkim }@etri.re.kr

### 요 약

CAMUS 시스템은 물리적으로 탑재되는 위치에 따라서 CAMUS 메인 서버와 Service Agent Manager 로 구분된다. Service Agent Manager 는 한 환경 내에 존재하는 센서와 장치들 (Service Agent)을 관리 및 제어하는 프로그램이다. 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.

### 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.

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

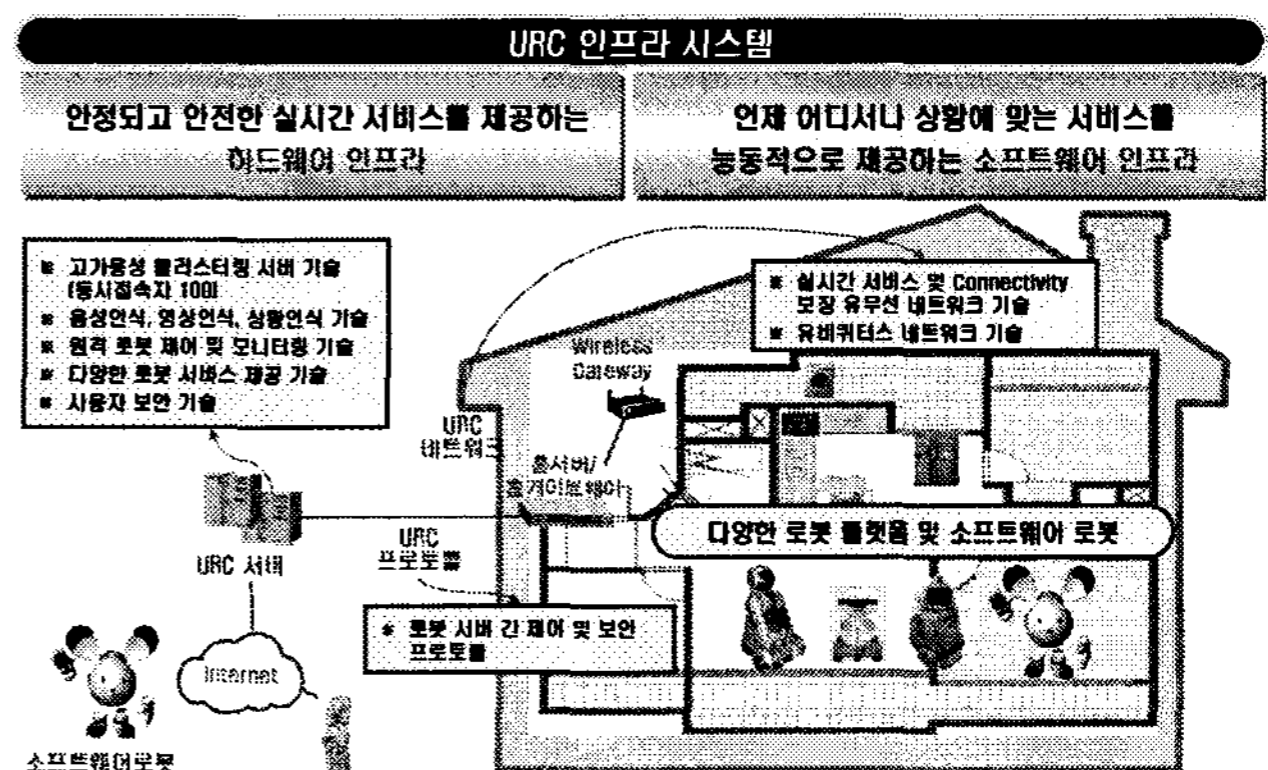


Figure 1. URC 인프라 시스템 구성도

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

- 상황정보 관리 및 접근 지원
  - 로봇 또는 소프트웨어 로봇에서 환경과 사용자에 대한 상황정보에 접근할 수 있는 수단을 제공
- 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
- 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.

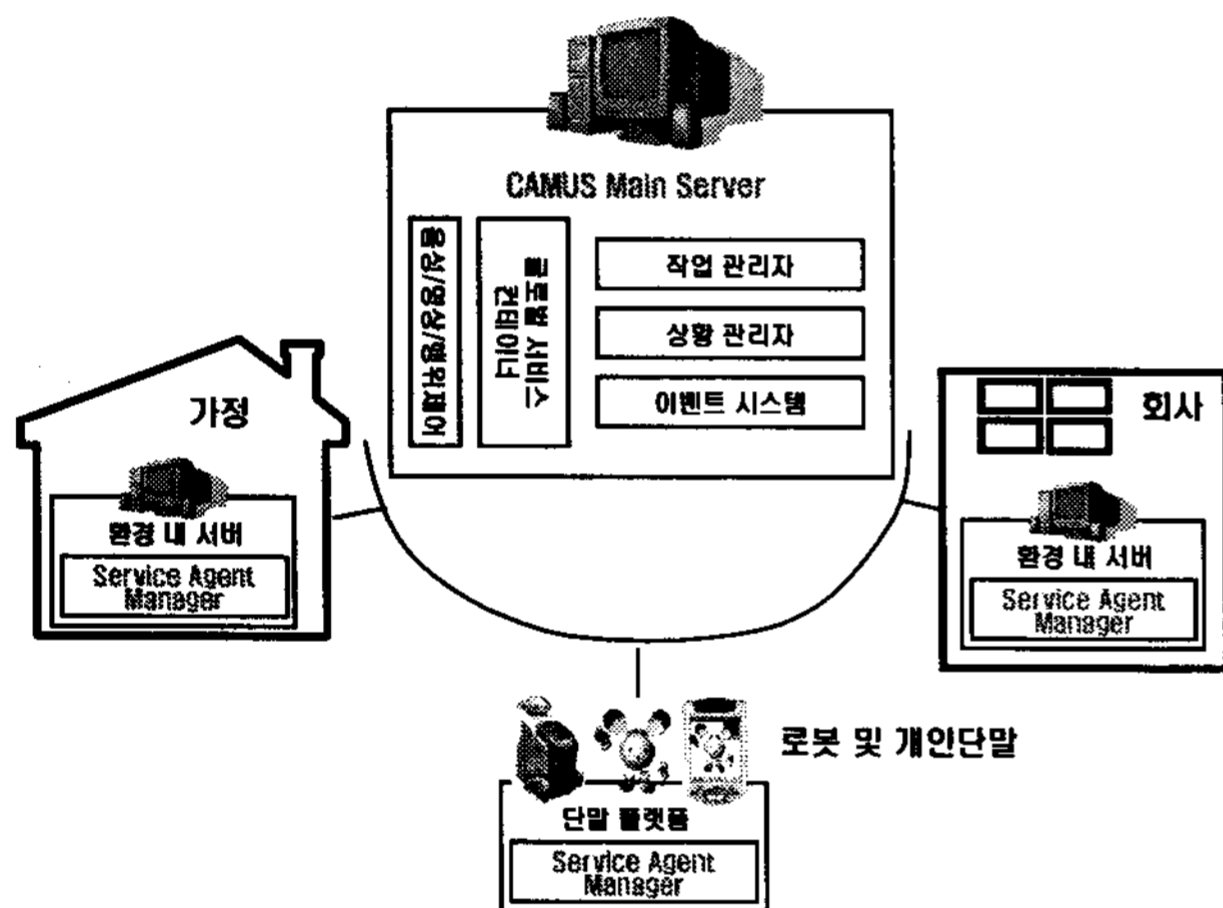


Figure 2. CAMUS 시스템 구성도

Service Agent Manager 는 한 환경 내에 존재하는 센서와 장치들 (Service Agent)을 관리 및 제어하는 프로그램이다. 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 환경내 장치를 제어하는 역할을 한다.

이러한 Service Agent Manager 는 임의의 공간 내에 설치될 수 있으며 로봇단말이나 개인 휴대단말 등에도 설치될 수 있다. 개인 휴대 단말에 설치되는 Service Agent Manager 를 특별히 Sobot Station 이라고 부른다.

CAMUS Main Server 는 여러 Service Agent Manager 로부터 전달되는 정보를 수집하여 Context 정보를 운용하고, 그 접근을 지원하며, 상황인식 응용 개발에 필요한 제반 기능을 제공하는 프레임워크이다. CAMUS Main Server 에는 환경 Context 와 사용자의 Context 에 대한 모든 정보를 관리하고 있으며, Context 변환에 따

른 이벤트를 응용 시스템에게 전달하여 그 Context 에 적합한 행위를 취할 수 있도록 한다. 또한 음성인식, 영상인식, 행위제어 등 로봇 또는 소프트웨어 로봇의 기본적인 서비스와 연동할 수 있는 서비스 컨테이너를 제공한다.

## 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 에이전트 인터페이스를 통해 서비스 에이전트가 제공하는 속성 및 동작을 외부 시스템에 노출한다. 이 둘 인터페이스에 대한 구현은 실제 센서나 장치 측에서 이루어지고 실제 CAMUS Task 등의 외부 시스템에서의 접근은 CAMUS 통신 프레임워크에 의해 이루어 진다. 즉 서비스 에이전트 인터페이스는 외부 시스템 과의 약정을 선언하는 것이고, 서비스 에이전트의 구현은 이 인터페이스에 대한 실제 의미를 기술하여 동작하게 하는 것이며, 서비스 에이전트에 대한 접근은 CAMUS 통신 프레임워크로 이루어진다.

예를 들면 사용자 위치를 인식하기 위한 Service Agent 가 있고, 이 에이전트 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 에이전트가 된다.

한편, 영상센서가 있다고 하자 이 영상센서로부터 사용자의 위치정보를 얻고자 한다면 마찬가지로 위치 센서 인터페이스를 영상센서로 구현한 프로그램 모듈이 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.

#### ○ Service Agent 의 구성요소

- 속성 (Attribute)

서비스 Agent 의 속성은 속성명(name)과 그 속성값(value)의 쌍으로 표현된다. CAMUS 시스템에서는 서비스 에이전트의 속성값을 읽어서 서비스 Agent 에 해당하는 센서 또는 장치의 상태 값을 확인하거나 그 값을 변경함으로써 서비스 에이전트에 해당하는 물리적인 장치를 제어할 수 있다.

- 동작 (Operation)

서비스 Agent 의 동작은 서비스 에이전트에 해당하는 연산처리, 또는 물리적, 논리적 장치의 제어를 위한 함수 호출에 해당한다. 이러한 동작 호출에 대한 결과는 환경에 영향을 미친다.

- 이벤트 (Event)

서비스 Agent 는 특정 상황 발생을 CAMUS Main Server 로 통보한다. 따라서 서비스 Agent 는 그 특성에 따라 발생하는 다양한 이벤트를 갖는다. 예를 들면 음성센서의 경우에 사용자의 음성명령이 들어올 때 SpeechReceived 이벤트를 발생시키며 RFID 센서의 경우에 사용자가 임의의 위치에 들어올 때 UserEntered 라는 이벤트를 발생시킨다.

다음은 음성센서 및 조명장치에 대한 예를 각각 보여 준다.

```
public interface SpeechSensor {
    // properties
    public String getSpeechList();
    public void setSpeechList(String speechList);

    // operations
    public void activate(String speechList) throws
    SpeechException;
    public void deactivate() throws SpeechException;
    public void reactivate() throws SpeechException;
    public boolean isActivated();
    public RecognizedSpeech recognize(byte[] stream)
    throws SpeechException;
}

public interface Light {
    // properties
    public boolean getPower();
    public void setPower(boolean power);

    // operations
    public void turnOn() throws
    LightException;
    public void turnOff() throws LightException;
    public boolean isTurnedOn();
}
```

}

Figure 3. CAMUS 센서 동작의 프로그램 예제

위의 SpeechSensor와 같이 센서에 대한 서비스 에이전트 인터페이스의 경우에 일반적으로 해당 센서의 동작을 시작/중지 시키는 동작(Operation)을 갖게 되며, 센서가 동작을 시작하게 되면 이벤트가 발생할 때 마다 Event Publisher를 통해 CAMUS Main Server로 전달하게 된다.

각 서비스 Agent에서 발생하는 이벤트 타입에 대한 정보는 인터페이스에 명시되지 않는데 이는 이벤트의 경우에 전형적인 Subscribe/Notification 방식으로 시스템 내부에 전달되기 때문이다.

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.

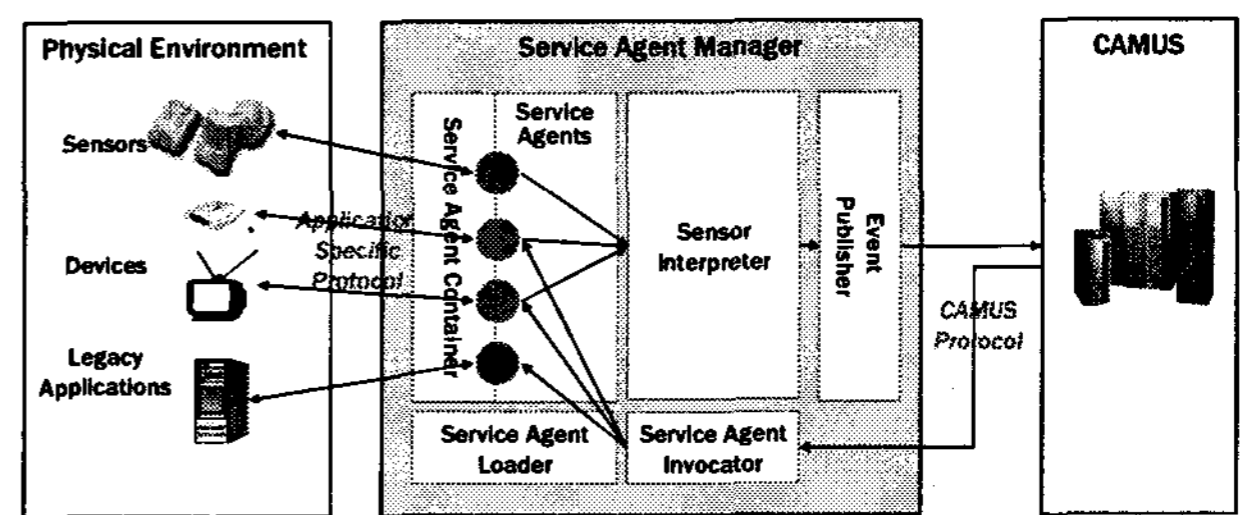


Figure 4. Service Agent Manager 구성

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 Server 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. 결론

이 논문에서는 CAMUS 시스템에서의 효율적인 SAM (Service Agent Manager) 관리방법에 대하여 기술하였다.

이러한 서비스 Agent 들이 실행되고 외부로부터 호출될 수 있게 하기 위해서는 Service Agent Manager에 설치된 서비스 Agent 들이 실제 자바 객체로 인스턴트와 되어 Service Agent Container에 탑재되고, Service Agent Registry에 등록되어야 한다.

현재 CAMUS 에서는 3 가지 방식의 Service Agent Loader ( XML, UpnP, OSGi) 을 지원한다.

CAMUS 에서는 환경 변화에 따라 Service Agent 를 원격에서 동적으로 관리할 수 있도록 OSGi 프레임워크 기반 Service Agent Loader 를 제공한다. 따라서 도메인 관리자는 환경이 변화되어도 Service Agent Manager 가 재기동 (Re-boot)하지 않고, bundle repository 로부터 동적으로 원하는 Service Agent 를 설치하거나 설치된 Service Agent 를 로딩, 갱신, 제거 등을 할 수 있게 한다.

#### 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. Elzabani, 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.