Communication protocol for the URC robot and server

Hyun-Sik Shim, Jongchang Lee, Tae-Hoon Kim
Samsung Electronics Co. LTD.
Dong Suwon P.O. BOX 105, 416 Maetan-3dong,
Yeongtong-gu, Suwon-si, Gyeonggi-do, 442-600, Korea
{hyunsik.shim, birdy.lee, ihoon.kim}@samsung.com

Hyun Kim, and Kang Woo Lee
Intelligent Robot Research Division
ETRI
161 Kajong-Dong, Yusong-Gu, 305-350, Daejon, Korea
{hyunkim,kwlee}@etri.re.kr

Abstract — A URC (Ubiquitous Robotic Companion) is a new concept for a network-based service robot. It allows the service robot to extend its functions and services by utilizing external sensor networks and remote computing servers. It also provides the robot’s services at any time and any place. The URC requires not only the hardware infrastructure such as ubiquitous networks or sensor networks and high-performance computing servers, but also the software infrastructure which resides above the hardware infrastructure. This paper introduces application protocol for URC robot applications to reduce hardware dependency, to improve usability, and to provide a server-client communication framework.

Keywords — URC(Ubiquitous Robotic Companion), URC Server, URC Robot, Protocol.

1. Introduction

As the industrial robot market is tending to stagnate, many robot industries are undertaking efforts for opening new market opportunities, especially in the field of service robots. In order that service robots can be introduced in our daily life as if they were digital appliances and information devices, they should be able to provide various services at affordable prices. However, current service robots seem to be high cost yet provide just simple services. One of the approaches to solve the problem is a network-based robot. Recently, the Korean government has strategically promoted the development of new concept of the network-based intelligent robot, which is called as URC (Ubiquitous Robotic Companion). Its goal is for the robot industries to be a key driving force of Korean economy by reducing robot costs and improving its services.

Generally, the robot has three functional components of sensing, processing and acting. The main approach of URC is to distribute these functional components through the network, and moreover, to fully utilize external sensors and external processing servers (Fig. 1)[1]. In many cases, it may be more efficient to utilize external sensors embedded in the environment rather than to expand the sensing functions of the robot itself. It may also be more preferable to utilize the high-performance computing servers in a remote side rather than to increase the computing power of the robot’s processor. If the sensing functions could be enhanced by using external sensors, the context-awareness of the robot would be dramatically improved. At the same time, if the robot could overcome the memory and processor constraints of its own by using remote servers, overall robot functions and services would be expanded, and the robot intelligence could be also improved.

Fig. 1  The concept of the URC

The URC requires not only the hardware infrastructure such as ubiquitous networks or sensor networks and high-performance computing servers, but also the software infrastructure which resides above the hardware infrastructure. The software infrastructure plays an important role to expand the robot functions and services, improve the context-awareness in the external environment, and enhance the robot intelligence[1].

It is very important to standardize the interfaces of hardware devices and the interfaces they support. Unfortunately, there was not much effort up to a few years ago on the standardization of network robot. Recently, several research institutes and companies have launched an international standardization group in OMG where SDO-DSIG and Robotics-DSIG are working hard to standardize various topics such as robot middleware, related APIs[2][3][4], and so on.

The URC protocol is the result of the project targeting the standardization of URC infrastructure and robots. This is the application protocol for network based robots and URC clients to communicate with URC server system.
Through the interface defined by the URC protocol, it is possible to control robots, and to have excess to the variety of services provided by robots/clients through servers.

The remainder of this paper is organized as follows. Section 2 discusses the URC system, URC robots and clients. Section 3 describes the URC protocol concept, mechanism and structure and section 4 explains the URC applications. Finally, Section 5 concludes this paper with some remarks.

2. The URC System

The figure 2 shows the structure of the URC system. The URC system is composed of URC servers, URC robots and URC clients.

The URC server plays an important role within the URC infrastructure in providing functions of various technical components required by URC robots or clients. This URC server executes voice recognition/synthesis, visual recognition, verification, image storage/streaming, external contents supply, circumstantial recognition, and so on.

From the points of views from URC server, the URC robots are categorized into the following four different types of robots. Each type differs in URC server verification and connection forms as well as in the usage of the profile defined by URC protocol.

- Physical robot (Mobot)
  - It means existing robots including mobile robots
- Embedded robots (Embot)
  - It means sensor system or robots with basic functions (ex. Camera-bot, voicebot and sensorbot)
- Software robot (Sobot)
  - The software robot is a virtual robot system embodied in the software infrastructure of the URC
- System robots (Sysbot)
  - It is a system which combines Sobots or Mobots with Embots managing overall functions (ex. CAMUS and Roboid server)

An URC client in short is a client application program which promotes service provision to users by utilizing numerous functions of URC robots and servers. Remote controls applications from PDA and mobile phones belong to the above. At large, The URC robot is commonly known as URC client.

The URC massage is a basic communication unit of URC protocol and performs its necessary functions by exchanging pre-defined URC messages between URC server and URC robots/clients.

The figure 3 illustrates the composition of URC server and URC robot/client in detail. The URC protocol is not the only means of the service between robots and servers within the URC infrastructure platform. At present, the commonly used services are based on the structure utilizing the existing standardized protocols. For instance, the service for voice communication and image streaming accommodates the existing protocols such as RTP/RTCP.

The protocol that the component named ‘URC Main’ adapts, is URC protocol within URC server. In order for other server system (ex. CAMUS) in the URC server group to use main function, URC protocol has to be utilized. [1], [5]

3. The URC Protocol

As shown in the figure 4, URC client/server communication protocol is the application protocol of TCP protocol.

The data exchanged between client/server is at the upper end of TCP, which is called URC message that consists of header and payload containing real data.

The URC infrastructure is comprised of URC servers which provides functions such as recognition and intelligence to various type of URC robots, and URC client is able to supply users various services by using these functions supported by URC infrastructure.
3.1 URC Client/Server communication protocol model

URC infrastructure is comprised of URC servers which provides functions such as cognitive techniques and intelligence to various types of URC robot. The URC client is able to supply users various services by using these functions supported by URC infrastructure.

The functions in the URC infrastructure have the units known as URC profile and the functions of this URC profile are requested through URC client/server communication protocol.

URC robots living in the ubiquitous environment exist within this surroundings and transmit user’s voice commands and other sensed information to URC server. They also receive requests from the server and proactively carry out all forms of services such as physical services using mobility, supply information via robots voice and so on. To make these services available, several types of communication models have to be shared between URC robot and URC server.

Case 1: transmission sensed information from URC robots to URC server
URC Robot $\rightarrow$ URC Server, Ex) voice command detection, motion detection etc.

Case 2: sending messages from URC server to URC robot
URC Server $\rightarrow$ URC Robot, Ex) robot movement command, speech guidance services etc.

Case 3: sending messages from URC client to URC server
URC Client $\leftrightarrow$ URC Server, Ex) request for speech synthesis etc.

In the URC environment, the form of messages including the requests of services is as follows:

- **Event notification**
  URC robots in the ubiquitous surroundings must be able to asynchronously notify URC servers the events such as user’s voice commands and motion detection. Then URC server understands the situation and the intention of the user by processing these asynchronous events, and hence is able to provide service in demand.

- **Request/Response**
  This is the request and the response of URC client to URC server in order to utilize various services provided by URC server.

  - Robot Control/Acknowledgement
    This is the activation request of and the response from URC server to URC robots for utilizing the functions of URC robots/clients. As the URC robots in the URC infrastructure connect to the network in the form of a private IP address where lies inside of AP (Access Point), an additional environmental setup is required if connectionless protocol is in use.

3.2 URC Client/Server communication protocol mechanism

The following is the function mechanism of URC client/server communication protocol.

- URC framing
- URC encoding
- URC authentication
- URC robot ACK (acknowledgement)
- HB (Heartbeat) Mechanism

3.2.1 URC Client/Server communication protocol Framing

The URC client/server communication protocol carries out communication based on messages in the TCP platform and the units of these messages are called URC messages. Framing of URC message adapts an binary format for the efficiency and numerous data contained in the URC messages borrows the data format of UDR (URC Protocol Data Representation)(See Table 1). Depending on its use, there are four ramifications of URC messages and they are namely URC Request, URC Response, URC Heartbeat and URC Event.

3.2.2 Encoding

URC messages are encoded with “little-endian” format and its Korean encoding uses “KSC-5601”.

3.2.3 Authentication

Every URC robot and URC client pass through authentication process to be identified into users and robots and then grant themselves of the necessary rights. Pre-registered ROBOT ID identifies URC robots and URC clients authenticate themselves with user ID and password.

3.2.4 Event Notification / Acknowledge

When the events such as voice commands and movements occur in the URC environment, URC robots must be able to asynchronously notify URC server. Using such events, URC server perceives user’s intention and situation, and then produce adequate services.
Furthermore, as most jobs done by robots are relatively time-consuming, server must be acknowledged with the start and the end of the job in the form of events in that the synchronizations of URC robots match with other functions. The URC server then matches the synchronizations of the jobs through ACK.

As each function of a robot can be defined by each component, this component performs its jobs in a state machine and should notify the event where the state is in transition.

3.2.5 HB (Heartbeat) Mechanism

URC robots simultaneously start their connection and keep this condition as long as the URC robot connects with URC server. Therefore, in case of disconnection caused by abnormal network environments, it is important to swiftly grasp the situation and to address the right direction. In order to monitor the connection between URC server and URC robots on a consistent basis, URC protocol defines HB (Heartbeat) protocol amongst URC robot/client and URC server to manage abnormal situations.

3.2.6 URC client/server Data Representation (UDR)

The table below describes the data used in UDR (URC client/server protocol Data Representation).

<table>
<thead>
<tr>
<th>TYPE</th>
<th>Corresponding to C++</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>short</td>
<td>short</td>
<td>2-byte integer</td>
</tr>
<tr>
<td>integer</td>
<td>int</td>
<td>4-byte integer</td>
</tr>
<tr>
<td>float</td>
<td>float</td>
<td>4-byte single-precision floating point</td>
</tr>
<tr>
<td>double</td>
<td>double</td>
<td>8-byte double-precision floating point</td>
</tr>
<tr>
<td>string[N]</td>
<td>char[N]</td>
<td>Fixed-length string data</td>
</tr>
<tr>
<td>string&lt;M&gt;</td>
<td>char[]</td>
<td>Variable-length char array</td>
</tr>
<tr>
<td>opaque[N]</td>
<td>char[N]</td>
<td>Fixed-length opaque data (binary data)</td>
</tr>
<tr>
<td>opaque&lt;M&gt;</td>
<td>char[]</td>
<td>Variable-length opaque data</td>
</tr>
<tr>
<td>T&lt;M&gt;</td>
<td>T[]</td>
<td>Variable-length array of type T</td>
</tr>
<tr>
<td>structure</td>
<td>structure</td>
<td>structure</td>
</tr>
</tbody>
</table>

Table 1 UDR Definition

3.2.7 URC Messages

URC messages are composed of URC common message header and URC message body. URC common message header contains basic information for communicating with URC server, and URC message body includes the information on the actual functions to be performed. The formats of messages supported by URC protocol is as follows, and they all share the same URC common message header.

- URC Heartbeat Message
- URC Request Message
- URC Response Message
- URC Event Notification Message

The URC common Message header shared among them has the following format.

The URC heartbeat message is periodically exchanged between URC robot and URC server to tackle abnormal situations found both in wireless and wired connection. The URC protocol can identify abnormalities through heartbeat message and reset the connection. The URC server transmits heartbeat messages on a regular basis and robots/clients reply to the server with heartbeat messages as a result.

The URC request message has the following format.

The URC response message has the following format.
3.2.8 URC Profile

The URC profile is the unit function of URC robot and URC server in the URC infrastructure. A profile consists of functions supplied by the profile and events. By using the components techniques of the profile, URC service plug-in or URC clients satisfy the services in demands.

At large, URC profile ramifies to the profiles supported in the URC server, common profiles of URC robots and specific profiles for robot’s unique functions.

- URC server profile

URC server profiles provide the interface for numerous functions such as URC server’s recognition function (such as speech or image recognition). The below is the representative profiles.

- System, Authentication profile
- Remote, Event profile
- ASR, TTS profile
- Face Recognition, Motion Detection profile
- Localization, Contents profile
- SMS, Real time recording profile
- Mpeg4 Play, VOD Play-list profile
- Reservation Recording profile

- URC Common Robot Profiles

URC common robot profiles support the interface of functions that all URC robots should provide.

- System, Move profile
- Navigation Profile
- EPD(End Point Detection), ASR profile
- Sound, Vision profile
- Motion, Sensor profile

- URC Robot Specific Profiles

The URC robot specific profile provides the interface of its characteristic functions. These characteristic profiles of URC robots can later be reregistered as the profiles of URC common robot profile.

- Text Display, Possible Output Notification profile
- Action-Emotion, Service Notification profile

Figure12 is an example explaining message flow of clients (PDA, mobile phones, PC) remotely controlling and monitoring the services that utilizes the defined URC profile.

4. URC system application

From October to December 2005, intelligent robot services are experimented on 64 households in the Seoul/Kyung-gi provisions using URC infrastructure and URC robot platforms developed since 2004. The three types of robot systems using URC protocol as well as URC server have been installed and carrying out services using BcN Network [6].

API using URC protocol is embodied in Linux, Windows, Win CE, JAVA platform and there are 9 mobile robots, a SW robot and a Embedded robot which are all
applied and tested. At present, there are two URC protocol standardization proposals, which are submitted to TTA (Telecommunication Technology Association of Korea) standardization workgroup PG413 (Intelligent Service Robot Project Group), as standardization candidates. [7].

![Fig. 13 URC experimental field services](image)

5. Conclusion

In a conventional robotic system, all computer processing is done on a robot. It causes the limitation of robot’s functions and services, and at the same time increases robot’s cost. Recently, the network connectivity is no longer an expensive add-on. Rather it is a basic feature of any computing system including robots. We proposed the new concept of the network-based intelligent robot, which is called as URC. In the URC system, the computer processing of the robot is distributed over the network. The sensors in the external environment are also utilized to improve the contextual situation.

In this paper, URC protocol has been described. This protocol has been used for URC experimental field services, together with several different types of robot.

In the future, we plan to improve it based on URC experimental field services results, and expand the profiles for increasing number of new services.

ACKNOWLEDGMENT

This research has been supported by Korean Ministry of Information and Communication (Grant No. A1100-0502-0028), and is part of the URC Project.

REFERENCES

[3] [http://www.robotservices.org](http://www.robotservices.org)
[4] [http://www.orin.jp/e/index.html](http://www.orin.jp/e/index.html)
[6] [http://www.urclife.or.kr](http://www.urclife.or.kr)