Development of RT-Middleware for Image Recognition Module

Akihiro IKEZOE¹, Hiroyuki NAKAMOTO¹, and Masayuki NAGASE¹

¹Systems Engineering Consultants Co., LTD., Tokyo, Japan

Abstract: We have developed “Image Recognition Device and Module” in “Development Project for a Common Basis of Next-Generation Robots” that is the consignment business of New Energy and Industrial Technology Development Organization (NEDO). The purpose of this project is to develop Image Recognition Module that is little hardware-scale, little power-consumption, useful and high-performance. The authors implemented the RT-Middleware on embedded processor DSP(Digital Signal Processor) for Image Recognition Module, for improve the development efficiency of component, and to offer the easy-to-use interfaces. In this paper, we introduce our project, concept of RT-Middleware, and mechanism of RT-Middleware for Image Recognition Module that we have developed.

Keywords: RT(Robot Technology), RT-Middleware, Image Recognition, Embedded System

1. INTRODUCTION

Needs for Next-Generation Robots that works instead of human has increased in Japan in recent years, because of lack of manpower by a falling birthrate and lack of nurse in a super-aged society. The Next-Generation Robots should be able to recognize the complex environment like the living space, for the robots coexist with human. We consider that image recognition is one of the most important functions in the environment recognition by the robot. It is necessary to process a huge image data in real-time to recognize the complex environment. To process a huge image data in real-time, the computer with an efficient processing performance like a PC is needed. Also, it is difficult to embed it in a robot because such a computer is large the hardware-scale and the power-consumption.

Therefore, our consortium (FUJITSU Ltd., The University of Tokyo, Tokyo Metropolitan University, The University of Electro-Communications, Business Design Laboratory Co., Ltd., and Systems Engineering Consultants Co., Ltd.) has developed Image Recognition Module that is little hardware-scale, little power-consumption, useful and high-performance. The authors implemented the RT-Middleware on embedded processor DSP(Digital Signal Processor) for Image Recognition Module, because to improve the development efficiency of component, and to offer the easy-to-use interfaces.

In this paper, we introduce our project, concept of RT-Middleware, and mechanism of RT-Middleware for Image Recognition Module that we have developed.

2. IMAGE RECOGNITION MODULE

2.1 Development Project for a Common Basis of Next-Generation Robots

Robots that had been developed so far are complex systems with low reusability, because functions of the image recognition and speak recognition, etc. are centralized controlled. Then, New Energy and Industrial Technology Development Organization (NEDO) [1]. executed “Development Project for a Common Basis of Next-Generation Robots” in FY2005-FY2007, for improving the robot development efficiency and promotion of the spread.

Fig. 1 shows the concept of the Next-Generation Robot that are considered about by this project. The Next-Generation Robot is composed of the combination of functions of modulated image recognition and voice recognition, etc. via the communication network. This project develops the technology of sharing and the standardization to integrate those modules. And the purpose of this project is to develop the Common Basis for the Next-Generation Robots. The project is composed of three themes: "Development of Image Recognition Device and Module", "Development of Speech Recognition Device and Module", and "Development of Motion Control Device and Module".

2.2 Development of Image Recognition Device and Module

Our consortium has developed “Image Recognition Device and Module” in “Development Project for a Common Basis of Next-Generation Robots”. The next generation robot needs the ability to recognize the situation in a complex environment. In a complex environment, the obstacle not expressible in map information such as human and furniture exists, and the
state of the lighting is not constant. Advanced image recognition ability similar to human is demanded so that the robot may act in such a complex environment. However, the hardware scale and the power consumption of the computer for advanced image recognition are generally large. Therefore, it is difficult to embed it in a small robot that supports human.

In this project, FUJITSU develops Image Recognition Device equipped with dedicated parallel processing operational circuit. SEC implements RT-Middleware on the Image Recognition Module, basic image recognition components, and development tools. FUJITSU, Univ. of Tokyo, Tokyo Metropolitan Univ., UEC and BDL perform experiments using robots with Image Recognition Module.

Fig.2 shows the hardware configuration of Image Recognition Module. Table 1 shows the hardware specification of Image Recognition Module.

![Fig. 2 Hardware Configuration.](image)

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSI</td>
<td>200MHz</td>
</tr>
<tr>
<td>MPU</td>
<td>TMS320DM642 600MHz</td>
</tr>
<tr>
<td>Flash Memory</td>
<td>4MB</td>
</tr>
<tr>
<td>SDRAM</td>
<td>32MB</td>
</tr>
<tr>
<td>DDR-SDRAM</td>
<td>16MB</td>
</tr>
<tr>
<td>Network Interface</td>
<td>10/100Base-T LAN</td>
</tr>
<tr>
<td>Camera Interface</td>
<td>CameraLink</td>
</tr>
<tr>
<td>Power</td>
<td>14.5[W] (max)</td>
</tr>
<tr>
<td>Size</td>
<td>≤ 150[mm] x 100[mm]</td>
</tr>
<tr>
<td>Weight</td>
<td>≤ 250g</td>
</tr>
</tbody>
</table>

3. RT-MIDDLEWARE

3.1 Summary of RT-Middleware

RT-Middleware[2] is a development framework to make basic robot element (RT: Robot Technology) a component. RT-Middleware developed by National Institute of Advanced Industrial Science and Technology (AIST) [3]. In the RT-Middleware, functional element of the robot is segmented into the unit that is called RT-Component. The various network-based robotic-system can be constructed by combining these components.

3.2 OpenRTM-aist

The guideline of RT-Middleware’s development technique is MDA(Model Driven Architecture)[4] that OMG(Object Management Group)[5] advocates. MDA is architecture to develop the system mainly using modeling. MDA consists of two models named PIM(Platform Independent Model) and PSM(Platform Specific Model). PIM is a model that independent on platforms of OS and the development languages, etc. PIM is chiefly modeled by UML(Unified Modeling Language)[6]. On the other hand, PSM is a model that depends on the platform. In development by MDA, first of all, the system developer designs the specification of the model based on PIM. PSM and implementation are generated from PSM with the tool automatically.

![Fig. 3 RT-Component’s PIM, PSM and Implementation.](image)

RTM Specification conceals the difference of the platform. Thus, the function and the interface model of the implemented RT-Middleware are compatible. What should be considered to make the RT-Middleware between different platforms cooperate is only a difference of a physical interface and the protocol. Then, the cooperation between RT-Middleware of different platform becomes possible by using the bridge that interface is converted mutually. In addition, if the bridge is involved by RT-Middleware, a seamless connection becomes possible between different platforms.

3.3 Advantage of RT-Middleware

“Development Project for a Common Basis of Next-Generation Robots” develop the technology of sharing and the standardization to integrate those modules. And the purpose of the project is to develop the Common Basis for the Next-Generation Robots. Then, a performance target of this project is that the developed module has the interface as the RT component.

We will be able to obtain the following advantages by implementing RT-Middleware. First, offering a general interface facilitates the development of the module that recycles easily. In this project, it is assumed to develop the robot by making developed Image Recognition
Module, Speech Recognition Module, and Motion Control Module cooperate.

Second, the following advantages exist for the robotic application developer: 1) The specification of APIs need not be investigated. 2) It comes to be able to develop in various processors, OSs, and the programming languages. 3) Cooperating with other RT-Components becomes possible easily. As a result, we consider that the development efficiency of the robotic application improves.

Third, the following advantages exist for Image Recognition RT-Component developer: 1) The developer only has to concentrate on the logic of image recognition. 2) The interface for the robotic application developer need not be considered. As a result, we consider that the development efficiency of Image Recognition Component improves.

**4. RT-MIDDLEWARE FOR IMAGE RECOGNITION MODULE**

4.1 Examination

We examined the implementation method of RT-Middleware on Image Recognition Module.

At present, two kinds of approaches are tried about the implementation method of RT-Middleware with interoperability in OpenRTM-ai.

The first approach is method that implements original RT-Middleware by using CORBA. The method of using JCG CORBA[10] has been proposed[11] so far. If Open-Fusion e*ORB SDR C Edition for DSP[12] (CORBA implementation for DSP) is used, it seems that this technique can be applied in the Image Recognition Module.

On the other hand, CORBA cannot be implemented on the environment such as hardware resource is poor, or without the Ethernet interface. Method of using Proxy Component was proposed[13], [14], to apply the RT middleware in such the environment. It is a method of using the Proxy Component that converts interface between an original protocol RT-Middleware and OpenRTM-ai. RT Component has interoperability to OpenRTM-ai by using this technique. This technique is examined to apply to the project of "Development of Motion Control Device and Module"[15], [16].

We considered which technique had to be applied to the Image Recognition Module. First, the method of using CORBA is costly for implementation although has the independence as the module.

The method of using the Proxy Component decreases the independence as the module. However, middleware can be constructed with the best protocol in each platform. The future, we consider that the independence as the module improves, because the module like Proxy Component comes to be connected to middleware.

Then, we decided it in the Image Recognition Module to the use of not CORBA but Proxy Component. And, we implement RT-Middleware based on RTM Specification for DSP by using the TCP/IP communication with an original message protocol.

Next, we examined the demand performance to this RT-Middleware. It is necessary to secure real-time, because this RT-Middleware is used for a high-speed image recognition. In addition, the memory footprint of the RT-Middleware should be small, because to use a lot of memories for the image recognition.

4.2 System configuration

Fig.4 shows the configuration of RT-Middleware for the image recognition module.

This system is composed of an original protocol RT-Middleware for DSP, and Proxy Component to offer the CORBA interface.

It explains the details as follows.

4.3 RT-Middleware for DSP

We explain the RT-Middleware for DSP.

In this RT-Middleware, RT Component runs as a real-time task. The start timing of the RT Component can be specified by the resolution of 1msec, because it has wrapped the scheduler function of DSP. This RT Component can use the function of the Image Recognition Device through the device driver.

The communication between RT Components in DSP used the message queue. In the communication, the communication delay is less to use the network.

This system is using the TCP/IP communication with an original message specification for the interface with the outside. And, it offers the command interface and the data interface to the Proxy Component.

Fig.5 shows a detailed object composition of this RT-Middleware. This RT-Middleware is composed by RtmServer, RtcManager, RtcDispatcher, RtcActivity, InPort, and OutPort etc.

RtmServer is the object that manages connection between RT Components for DSP and Proxy Components. RtmServer waits for the connection request from the Proxy Component. When the connection request is received, it establishes the connection with the corresponding RT Component. When the corresponding RT Component doesn’t start, it send starting request to RtcManager.
RtcManager is the object that manages the RT Component. It has the function of "Generation of RT Component", "Search of RT Component", "Management of the awake timing of RT Component", and "Getting the object reference".

RtcDispatcher is the object that manages the connection with Proxy Component. It receives the message from the Proxy Component, and dispatches the message to the corresponding object (RtcActivity and InPort). RtcActivity is the activity of the RT Component, and it runs according to own state and inputted data. In this project, RtcActivity accesses the Image Recognition Device.

InPort and OutPort are the I/O ports to send and receive data between RT Components. RtcActivity can communicate data with other RT Components in DSP and RT Components connected on the network by using this InPort/OutPort.

### 4.4 Proxy Component

The Proxy Component is RT Component constructed on OpenRTM-aist. It mutually converts the interface between RT Components constructed on OpenRTM-aist and RT Component by implemented other protocols. Even in the RT-Middleware of the platform where CORBA cannot be implemented, RT Component has interoperability to OpenRTM-aist by using Proxy Component.

Fig. 6 shows the relation to RT Component constructed on DSP and Proxy Component.

Proxy Component receives the message from other RT Components constructed on OpenRTM-aist via the CORBA communication. The received message is converted according to an original message specification, and it is sent to RT Component on Image Recognition Module. It receives the data output from OutPort and the command reply from RT Component on Image Recognition Module. The received message is converted according to the CORBA specification, and it is sent to RT Component on OpenRTM-aist.

#### 5. EXPERIMENTAL

We evaluated function and performance of the developed RT middleware. Fig.7 shows the system configuration for evaluation.

This system sends image data from the USB camera to the RT Component on DSP via Proxy Component. The RT Component on DSP executes an easy image data processing. These Components executes grayscale conversion and right-left reversing. The processing result is output to the monitor via Proxy Component.

#### 5.1 Function Evaluation

Table 2 shows the result of function comparing OpenRTM-aist with RT-Middleware for DSP. RT-Middleware for DSP conforms to RTM Specification, and it is possible to connect with OpenRTM-aist in the seamless.

(1) **Command Interface**

In this RT middleware, the command on RTM Specification was able to be sent to RT Component on DSP via Proxy Component. We confirmed RT Component that received the command changed own state, and executed processing according to own state.
### Table 2 Comparison between OpenRTM-aist and Our RT-Middleware

<table>
<thead>
<tr>
<th>Function</th>
<th>OpenRTM-aist</th>
<th>RT-Middleware for DSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command interface</td>
<td>RTM Specification and original extension</td>
<td>RTM Specification</td>
</tr>
<tr>
<td>Data type</td>
<td>Primitive type with timestamp</td>
<td>Primitive type with timestamp</td>
</tr>
<tr>
<td></td>
<td>Primitive type sequence with timestamp</td>
<td>Primitive type sequence with timestamp</td>
</tr>
<tr>
<td></td>
<td>User’s definition type</td>
<td>User’s definition type</td>
</tr>
<tr>
<td>Dynamic connection between</td>
<td>Enable</td>
<td>Enable</td>
</tr>
<tr>
<td>components</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Composite component</td>
<td>Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>Function of RtcManager</td>
<td>Generate, Register, Search, Load Components</td>
<td>Generate, Register, Search, Awake Components</td>
</tr>
<tr>
<td>RT-Component generator</td>
<td>Available</td>
<td>Not Available</td>
</tr>
</tbody>
</table>

(2) Data Interface
We confirmed an active communication and a passive communication were possible, when communicating between components. In the communication between components, usable data was primitive types, sequence of primitive types, and user definition types.

5.2 Performance Evaluation

(1) Realtime Performance
We measured the start delay time of RT Component by using the profiler tool. The delay time was always shorter than five microseconds. This result shows that this RT-Middleware demonstrates real-time by DSP.

(2) Processing Performance
We measured the delay time when data was sent and received between RT Components by using the profiler tool. Fig.8 shows the result. This figure shows that communication time increases in proportion to size of sending and receiving data. This shows that RT Component cannot has assurance of real-time, when the data size exceeds tens KB.

![Fig. 8 Processing Performance.](image)

On the other hand, the memory size that RT-Middleware dynamically allocates is proportional to data size and number of I/O ports. For example, the component with each of one I/O Port where the size of 200kB data is sent and received dynamically allocates the size of 2MB memory.

6. CONCLUSION

This paper introduced RT-Middleware for Image Recognition Module. We evaluated the performance of our RT-Middleware. The result of our evaluation clearly shows the following. First, RT-Component on RT-Middleware that we has developed can connect to other components seamlessly via the network. Second, our RT-Middleware has interoperability with OpenRTM-aist, by it has a minimum function as RT-Middleware and our Proxy Component model. Third, our RT-Middleware has enough real-time on embedded processor, if sent and received huge data like the image, performance of RT-Middleware might degrade.

We will develop the image recognition components and the tools to support developer, and improve RT-Middleware in the future.

(1) Improve RT-Middleware
The demerit of our RT-Middleware was to have sent and received huge image data. Then, we will improve the performance, and add the function to specialize in the image recognition processing.

(2) Develop Image Recognition Components
Image Recognition Module will be embed for various robots. We will develop the image recognition component that can be generally used for those robots.

(3) Develop Tools to Support Developer
This RT-Middleware doesn’t have the development support tools. Then, tool that generates the template source code of the component automatically and tool that can do graphical programming of image recognition will be developed in the future.

REFERENCES