Robot Middleware and its Standardization in OMG

Tetsuo Kotoku, Takashi Suehiro, and Olivier Lemaire
Natl Inst of AIST
Tsukuba, Ibaraki, Japan
t.kotoku@aist.go.jp

Masayoshi Yokomachi
NEDO
Kawasaki, Kanagawa, Japan

Makoto Mizukawa
Shibaura Inst of Tech
Minato-ku, Tokyo, Japan

Abstract—This paper introduces the basic concept of Robot Middleware and our standardization activities in OMG. For the efficient software development of robotic systems, we are planning to develop a common robot architecture based on the modularization of software. To enhance the "reusability" and "interoperability" of software modules, we are inviting your participation in standardization activities.

Keywords—RT (robot technology), middleware, standardization, OMG

I. INTRODUCTION

The robot industry in Japan has been expanded through the spread of industrial robots for manufacturer’s plants in automobile, electric appliance and other industries. These robots are based on large-scale production. The optimized robot design is the most important for the low-cost competitions. Therefore, it has been allowed to take a lot of time and to pay considerable costs for the design of mass-production robots.

In 21st century, the increase of the high elderly population is predicted. In such a high elderly dominated society, how to support the elderly in their daily lives and how to keep enough labour force in industrial and social activities will be the crucial issues to be solved. Under such circumstances, the service robots working in our daily life environments, such as hospitals, welfare facilities and households are expected to come onto the market.

However, these service robots have to meet the enormously diverse requirements of each user and have to deal with various complicated tasks. So it intrinsically becomes the manufacturing of a wide variety of products in small quantity. The conventional highly optimized design for robots will cost too much.

In order to solve these problems, the Japan Robot Association (JARA) has drawn up technology strategies for creating a “Robot Society” in the 21st Century[1]. In this report, the transformation to the Made-to-Order business model was proposed (Fig.1). Even small venture companies will be able to offer robotic components with their own unique and specialised technology. Through division of labour and specialization of the service robot industry, integration business will be born.

The custom-made robotic design based on the integration of various standardized robotic components becomes important. To realize this new architecture, the robot middleware will be one of the key technologies in future robot industry.

II. STANDARDIZATION ACTIVITY IN OMG

In the field of robotics, large numbers of research activities have been done separately to address the technological problems such as improving reliability, safety, operability and amenity for individual robotic systems. Unfortunately it is extremely difficult to share the fruits of these works. Because the most of software for robotic systems are based on proprietary design architectures invented from scratch each time[2].

There are lots of activities seeking for standards based on the commonality of robotic design and function in the various fields. Unfortunately, most of pioneering initiatives are developed independently of the others, driven by specific applications and objectives. To settle this state of chaos, we would like to contribute to the promotion of standardization in the field of robotics based on the mutual understanding between the relevant parties.

In order to increase interoperability, compatibility and reusability between the various robotic common components, the Robotics Domain Special Interest Group (Robotics-DSIG)[3] was established in February 2005 within the Object Management Group(OMG)[4]. OMG is an international software consortium responsible for establishing distributed computing specifications.

The Robotics-DSIG is looking to begin a dialog with vendors, end users, researchers, robotics organizations...
The purpose of the robotics activities in OMG is to foster the integration of robotics systems from modular components through the adoption of OMG standards. To realize this purpose, Robotics-DSIG will:

- Adapt and extend OMG technologies that apply to the specific domain of robotics systems where no current baseline specifications exist, such as MDA for Robotics. The object technology is not solely limited to software but is extended to real objects. This effort promotes the use of OMG technologies in various markets.
- Promote mutual understanding between the robotics community and the OMG community.
- Endeavor to collaborate with other organizations for standardization, such as the one for home information appliances, and make an open effort to increase interoperability in the field of robotics.
- Coordinate with the appropriate OMG subgroups and the Architecture Board, for technology areas that overlap with other OMG Task Forces, to determine where the work will be accomplished.

III. ROBOT TECHNOLOGY COMPONENTS

Generally, robot systems are integrations of a lot of robot technology functions, such as actuators, sensors, controllers, and so on, some of which include hardware and some are pure software. In order to make those robot systems, especially non-industrial custom-made robot systems, easier and more effective, it is necessary to compose robot systems as integrations of robot technology function modules.

Fig.2 is a typical figure of the robot system which is formed from modules. For example, the servo of a robot arm consists of functional elements such as

- A sensor module which outputs the angle of a joint,
- A controller module which takes the angle information output from the sensor module and decides what the input should be to an actuator module,
- An actuator module which operates the actuator hardware based on its input.

The servo module integrates and manages the behaviors of those elements. In addition, the robot arm system which consists of these function modules can, itself, be considered a module that provides arm functions.

As for the hardware dependent layer, developers of robot systems use various kinds of platforms, computer hardware, operating systems, computer languages, communication methods and so on. Those are dependent on implementations and it is difficult to standardize those implementations.

To achieve the “reusability” and “interoperability” of robot modules, it is important to standardize in the distributed object layer. We call the robot modules in this level “Robot Technology Components (RTCs)”.

RTCs are software modules which support the functionality of such robot technology elements as sensors, controllers and motors. They have following characteristics (Fig.3):

- Each RTC has its own functional task. For example, the sensor RTC periodically accesses the hardware measurement device, gets data, does data processing, and sends the data to the controller if necessary. The controller RTC sends the actuator RTC the operation input that it has decided upon, based on the received data.

The functionality might have been achieved through a combination of hardware and software mechanisms, or it might have been achieved purely through the software. RTCs could be passive or active components.

- They exchange data with each other. For instance, the controller RTC receives data from the sensor RTC,
and decides the value it will provide to the motor RTC operation.

Fig. 4(a) shows an example of typical data flow between RTCs. A filter is often added to the output data of the sensor for the characteristic improvement as shown in Fig. 4(b). It is important that, to achieve interoperability and reusability, the RTCs should be developed such that other RTCs do not have to be changed.

**Fig. 4. Interoperability of data flow between RTCs**

- They exchange commands with each other. For instance, the controller RTC changes the operation parameter and changes the operation mode according to requests from other modules.
- It is necessary to be able to compose a new RTC from a combination of RTCs. This supports hierarchical construction of robot systems. Moreover, this gives the degree of freedom of the grain size selection to the RTC developer. And, this facilitates wrapping an existing technology as an RTC.

For instance, a certain developer might construct the arm component as a synthesis of a sensor, a controller, and a motor as shown in Fig. 2. Or, a conventional arm vendor might wrap an existing arm system as an arm component without changing the existing arm system. Interfaces for each component RTC must be provided as well as an interface for the combined RTC. For example, an arm and a hand may be made as separate RTCs (Fig. 5). It is also possible to combine an arm and a hand in a single RTC. Additionally, an “arm and hand” RTC may be made by combining an arm RTC and a hand RTC. No matter how created, a member of each category of RTC must be treated the same as every other member of the category.

**Fig. 5. Interoperability in Composition of RTCs**

RTCs are modules of robot technology functions which may include hardware devices. OMG already has the specification of Super Distributed Objects (SDO)[5] for modules that may contain hardware devices. SDO is defined by a key interface called SDO interface, other interfaces related to functions which SDO provides and the resource data model which expresses various attributes of the SDO including these interfaces. The user can access all functions that SDO offers by acquiring the resource data through the SDO interface.

An RTC can be considered as a kind of SDO. Advantages to using SDOs as a basis of the RTC specification:

- SDO is the specification of autonomous modules containing hardware devices,
- The attribute and the offered function could be accessed through the resource data obtained from the SDO interface.
- Above all, the SDO specification is an existing OMG standard.

Disadvantages to using SDOs as the basis of the RTC specification:

- The SDO specification does not yet guarantee reusability and interoperability. The robotics domain must first mutually agree on common usage patterns of the SDO specification.
- The resource data of a SDO contains a lot of information about the SDO. But there is no specification of which data are essential for reusability and interoperability and which are not. For example, “vendor” of the SDO might not be important for reusability and interoperability.

To use SDO in robotics domain, extension and modification of the specification are needed.

**IV. RTCs Request for Proposal**

After the collaboration of Robotics-DSIG, SDO-DSIG, and Middleware and Related Services Platform Task Force (MARS-PTF), OMG issued the “Robot Technology Components (RTCs) Request for Proposal” (RFP)[6] in September 2005. This RFP solicits proposals that provide a platform independent model (PIM) and at least one CORBA-specific model of RTCs as an extension to the specification of Super Distributed Objects (SDOs). In the proposals, the submitters have to show at least one example of RTCs for a specific application robot system (manipulator control, mobile robot system, robotic space, and so on) using the specified model and interface.

**A. Scope of Proposal Sought**

This RFP seeks proposals that specify RTCs as a framework, based on the SDO specification, for the modularization of robot technology functions in the distributed object layer. It is necessary to consider the following in the specification of RTCs:

1. The proposed RTC specification should provide a framework for the modularization of robot technology functionality in an interoperable and flexible manner.
(2) The RTC specification must be general enough to allow a variety of robot systems to be easily constructed.
(3) The RTC specification must provide for reusability and interoperability. Modules implemented by one vendor should be able to be replaced with modules from other vendors.
(4) The functions provided by RTCs and environment in which RTCs are used may vary according to each robot system. Therefore, the grain size of RTCs should be able to freely chosen by developers. Additionally, a large RTC may be composed of smaller RTCs.
(5) Efficiency, timeliness and small footprint are important aspects of robot technology. However, the degree of each strongly depends on implementation and the target environment. Since such concerns are not platform independent, this RFP does not mention them, but rather seeks proposals in which interoperability and reusability are the most important features.
(6) As described earlier, RTCs can be considered applications or specializations of SDOs within the robotics domain. In specializing SDOs, RTC specification should consider the following:
   - The possible specification of a common service interface for all RTCs. Such an interface might handle on/off functions, for example.
   - RTCs shall communicate with other RTCs directly, and cooperate to realize their tasks. RTCs should provide both the command and data communication interfaces.
   - Interfaces for each component RTC must be provided as well as an interface for the combined RTC. In the previous example (Fig.5, interfaces should be provided for arm functions, for hand functions and for “arm with hand” functions. The base states and the state transitions of the life cycle of all RTCs need to be defined. Moreover, RTCs should be composable, i.e., the externally visible states of “arm with hand” are the same as “composed arm with hand.”
   - The resource data of a SDO is designed to contain a lot of information about the SDO. But most of items are optional and usages of them are not defined. We would like to specify some items of the data model appropriate to express attributes of RTCs well.
   - A discovery method of RTCs is important in real usage, though it is not included in the specification of SDO given the fact that it is often an implementation or application-dependent issue. However, it may be beneficial to include some generic discovery method.
(7) Real-time support Real-time features are sometimes very important in robot programming. They strongly depend on implementation and platform choices, so that the developer of each RTC can choose the appropriate platform and write a real-time program. While it is very difficult to support “hard” real-time operation between RTCs, ideas for platform-independent “soft” real-time or pseudo-real-time support are encouraged. Ideas for alternate ways to support real-time features, for example the adding of timestamps to data, are also welcome.

B. Timetable

The rough timetable for the standardization process for this RFP is given below.

<table>
<thead>
<tr>
<th>Days</th>
<th>Event or Activity</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LOI to submit to RFP due</td>
<td>December 15, 2005</td>
</tr>
<tr>
<td>129</td>
<td>Initial Submissions due</td>
<td>January 23, 2006</td>
</tr>
<tr>
<td>150</td>
<td>Initial submission presentations</td>
<td>February 13, 2006</td>
</tr>
<tr>
<td>262</td>
<td>Revised Submissions due</td>
<td>June 5, 2006</td>
</tr>
<tr>
<td>283</td>
<td>Revised submission presentations</td>
<td>June 26, 2006</td>
</tr>
<tr>
<td>287</td>
<td>TC votes to recommend specifications</td>
<td>June 30, 2006</td>
</tr>
<tr>
<td>360</td>
<td>BOD votes to adopt specifications</td>
<td>September, 2006</td>
</tr>
</tbody>
</table>

V. Conclusion

The robot middleware technology will help create new markets. Even small venture companies will be able to offer RTCs with their own unique and specialised technology. Through division of labour and specialization of the service robot industry, integration business will be born. Robot middleware technology is the key to reform the structure of the robot industry. We have already started working toward the standardization of robot middleware. This innovative technology should make possible the development of robots that can respond to the diverse needs of everyday life.

Acknowledgment

We greatly appreciate the OMG staffs and senior members for their tremendous help and kind assistance for us.

References