論文 / 著書情報
Article / Book Information

論題	
Title	Basic Study on Element Administration in Robotics Middleware -A first approach using Role Definitions-
著者	Ceron Lopez Arturo E., 福島 E. 文彦
Author	Arturo E. Ceron Lopez, Edwardo F. Fukushima
掲載誌/書名	,　, No. 12-3,
Journal/Book name	Proceedings of the 2012 JSME Conference on Robotics and Mechatronics,　, No. 12-3,
発行日 / Issue date	2012, 5
URL	http://www.jsme.or.jp/publish/transact/index.html
権利情報 / Copyright	本著作物の著作権は日本機械学会に帰属します。
Note	このファイルは著者（最終）版です。 This file is author (final) version.

Basic Study on Element Administration in Robotics Middleware
-A first approach using Role Definitions-

Arturo E. CERON LOPEZ and Edwardo F. FUKUSHIMA

Tokyo Institute of Technology, aceron@robotics.mes.titech.ac.jp, fukusima@mes.titech.ac.jp

This article proposes a novel method for the administration of elements in robotics middleware according to their given Role Definition, which contains information about the element type, the functions that the element can perform, as well as the inputs and outputs related to it. By looking at the available elements and their classification based on their Role Definition, the elements can be managed. In order to test this method, an implementation that we named as Interface for Administration by Roles (IAR) is realized; it becomes the intermediary between element connections. The IAR can be used in the OpenRTM-aist platform as an RT-Component.

Key Words: *Service Robot, Robotics Middleware, Role Definition, Administration*

1. Introduction

Standards for service robots have been discussed over the last years. The current standard approach is to use middleware for robotics.

By definition, a middleware consists of a set of services that allow many processes to run on one or more machines and to interact with each other. Various middleware for robotics have been developed in the recent years, from these available middleware, *Robotics Technology Middleware* (RTM) [1], *Robot Operating System* (ROS) [5] and *Yet Another Robot Platform* (YARP) [2] are popular for use in service robots.

Some robotics middleware platforms such as RTM, are based on CORBA (Common Object Request Broker Architecture) [4] standard defined by OMG, software elements are regarded as RT-Components (RTCs), which generally are programs designed to perform a certain task (e.g. read joystick, process images, and produce speech). RTM standardizes the way in which the RTCs should behave and communicate between each other, data-types are also defined, among other things.

The Universal Plug and Play (UPnP) [6] architecture with some additional features has also been suggested to be used as robotics middleware.

2. Objective

The RTM standard helps to make the development of robot software easier and in some cases more feasible. Developers can create independent specialized programs in the form of RTCs and let them interact with each other, while all run at the same time as different processes.

However, RTM standard does not provide a definition for the role of elements, leaving the following question: Which role does each element play in the operation or action done by the service robot?

The lack of a definition of role for the elements can lead to a poor or wrong administration of the available elements, translating into less pleasant programming experience for the final user, since some background knowledge about the way the element operates is needed.

The objective of this study is to propose a novel method for the administration of elements in robotics middleware using their given roles; we call it Administration by Roles. As for the method implementation, we named it Interface for Administration by Roles (IAR). This article takes RTM for discussion.

3. Administration by Roles

Our proposed method first classifies the elements according to their specified Role Definition, which is assumed to be already given (by the programmer, vendor, etc…) and can contain description information about the element type, the kind of basic actions or functions that the element can perform, the kind of data needed as input and generated as output; then according to this information, element connections, statuses and other features are managed; this is implemented in the IAR.

For building a service robot system, classical (Fig. 1) and proposed (Fig. 2) approaches can be compared.

The classical approach has generally 4 main parts: input, function, output and feedback; each one represented with elements having input and output ports. In robotics middleware, these elements are connected directly to each other, delivering data from output ports to input ports; the elements decide when and how to send data. Here, the user should decide the connections between ports and then connect manually each output port to the correct input port.

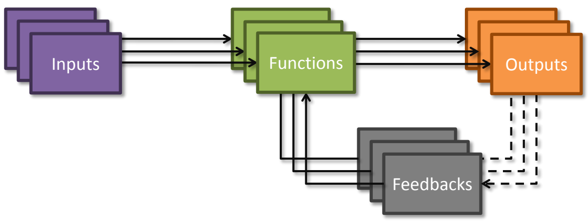

Fig. 1 Classical approach for service robot systems.

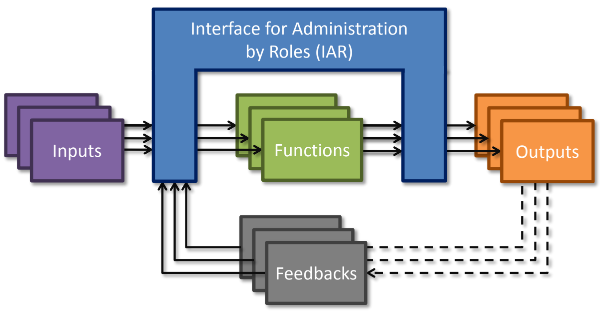

Fig. 2 Proposed approach for service robot systems.

In the proposed approach, the 4 main parts can be the same as in the classical approach. However, connections between ports are not to be directly made between elements, they are to be connected through the IAR, thus elements with output ports should deliver and report data to the IAR, then data is to be delivered by the IAR to other elements that possess input ports.

Then the IAR, when set by the user, can receive all the output data from all the elements, this data is processed, formatted and organized by the IAR, then it is sent to the elements that may need it. To do this, the IAR manages the features of the elements, like performing the task of connecting their ports automatically and dynamically.

4. Implementation Example

An example of the IAR for building a service robot system is shown in Fig. 3, where it can be seen that the IAR is located between the communications of each element, encapsulating the function elements.

The basic internal structure of the IAR is composed of a set of databases for the data coming from the elements and the data to be sent to other elements inside the system, an algorithm which helps to process the data, and algorithms to classify the elements and to manage their features; it could be regarded as a virtual user since it can perform similar administrative operations as a real user.

However, the IAR is not limited to the mentioned tasks and could do as much as the libraries in which it is based allow (e.g. modification of element configuration parameters, element resetting on error, etc…).

It is to be noted that the definition of the 4 main parts presented here for a service robot system are an application example and does not limit the number of parts or stages involved in the system. However, the Role Definition can be an aid to define the number of parts or stages involved in the system.

4.1 Implementation on RT-Middleware

To test the IAR, an omnidirectional platform is used (Fig. 4); the test implementation (based on CORBA and RTC libraries) was made in the form of RTC for use in the OpenRTM-aist platform (Fig. 5); the IAR becomes the intermediary between element connections.

5. Discussion and Conclusion

As presented in this article, a method of administration using role definitions can help to solve the element administration issue in robotics middleware.

As an example, by looking at the available elements and their classification based on the Role Definition, the elements can be connected automatically, where the IAR is the intermediary between element connections. A question remains about how much the IAR will delay the data transmission between elements. However, preliminary results show that the introduced time delay does not affect element's normal operation.

We think that when the final user is not familiarized with robot technology, the IAR can improve user's experience by focusing more on the desired application.

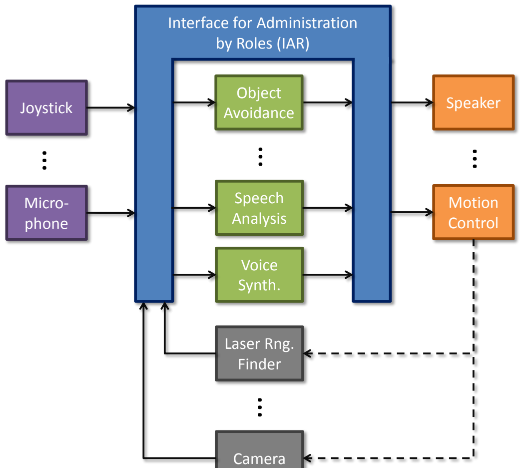

Fig. 3 Example of IAR for service robot systems.

Fig. 4 Omnidirectional platform for testing.

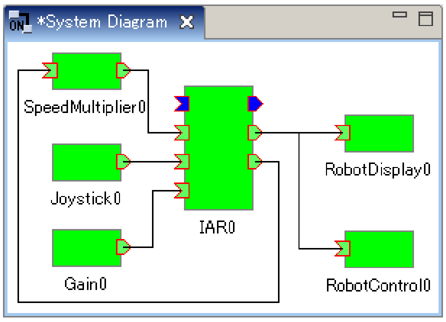

Fig. 5 Implementation of the IAR as RTC.

As future work different scenarios are to be studied with various kinds of inputs, functions and outputs; also response time issues are to be analyzed.

Acknowledgement

The first author acknowledges support from CONACyT through a scholarship for graduate studies at TokyoTech.

References

[1] Ando, N.; Suehiro, T.; Kitagaki, K.; Kotoku, T. & Woo-Keun Yoon; "RT-middleware: distributed component middleware for RT (robot technology)", 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.3933-3938, 2-6 Aug. 2005.

[2] Metta, G.; Fitzpatrick, P. & Natale, L.; "YARP: Yet Another Robot Platform", International Journal on Advanced Robotics Systems, vol. 3 (1), pp. 43-48, 2006.

[3] Mizukawa, M.; "Robot technology (RT) trend and standardization", 2005 IEEE Workshop on Advanced Robotics and its Social Impacts, pp. 249- 253, 12-15 June 2005.

[4] Object Management Group, http://www.omg.org

[5] Quigley, M. et al.; "ROS: an open-source Robot Operating System", ICRA Workshop on Open Source Software, 2009.

[6] Sang Chul Ahn et al.; "Requirements to UPnP for Robot Middleware", 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.4716-4721, 9-15 Oct. 2006.