

Research Article

Single-Switch User Interface for Robot Arm to Help Disabled People Using RT-Middleware

Yujin Wakita,¹ Natsuki Yamanobe,¹ Kazuyuki Nagata,¹ and Eiichi Ono²

¹Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST), Ibaraki, Tsukuba 305-8568, Japan

²Department of Rehabilitation Engineering, Research Institute of National Rehabilitation Center for Persons with Disabilities, Saitama, Tokorozawa, 359-8555, Japan

Correspondence should be addressed to Yujin Wakita, wakita.y@aist.go.jp

Received 1 June 2011; Revised 16 September 2011; Accepted 30 September 2011

Academic Editor: Tetsuya Mouri

Copyright © 2011 Yujin Wakita et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We are developing a manipulator system in order to support disabled people with less muscle strength such as muscular dystrophy patients. Such a manipulator should have an easy user interface for the users to control it. But the supporting manipulator for disabled people cannot make large industry, so we should offer inexpensive manufacturing way. These type products are called “orphan products.” We report on the construction of the user interface system using RT-Middleware which is an open software platform for robot systems. Therefore other user interface components or robot components which are adapted to other symptoms can be replaced with the user interface without any change of the contents. A single switch and scanning menu panel are introduced as the input device for the manual control of the robot arm. The scanning menu panel is designed to perform various actions of the robot arm with the single switch. A manipulator simulation system was constructed to evaluate the input performance. Two muscular dystrophy patients tried our user interface to control the robot simulator and made comments. According to the comments by them, we made several improvements on the user interface. This improvements examples prepare inexpensive manufacturing way for orphan products.

1. Introduction

The use of assistive technologies by persons with disabilities to pursue self-care, educational, vocational, and recreational activities continues to increase in both quantity and quality [1]. The number of academic programs, clinical centers, schools, hospitals, and research institutes applying these assistive technologies has increased dramatically. The assistive technology device is defined as an item, piece of equipment, or product system that is used to increase, maintain, or improve functional capabilities of individuals with disabilities. As described in [1], the assistive technology system is consisting of an assistive technology device, a human operator who has a disability and an environment in which the functional activity is to be carried out.

The human activity assistive technology (HAAT) model is proposed as a framework for understanding the place of assistive technology in the lives of persons with disabilities,

guiding both clinical applications and research investigations [1]. The model has four components, the human, the activity, the assistive technology, and the context in which these three integrated factors exist. The assistive technology includes human/technology interface, environmental interface, and activity output component to contribute to functional performance. One of the important activity outputs is manipulation.

To establish the manipulation as an activity output, service robots which can support disabled people in their daily life are expected to appear in the real market. The robotic arms which can support disabled people with muscular dystrophy, spinal cord injuries, ALS, and cerebral paralysis are under development [2–6]. The manipulator should be portable for use on a bed or wheel chair and be easy for the user to control. The symptoms of the patients are varied, and the input methods to control the robot should be adapted to them individually.

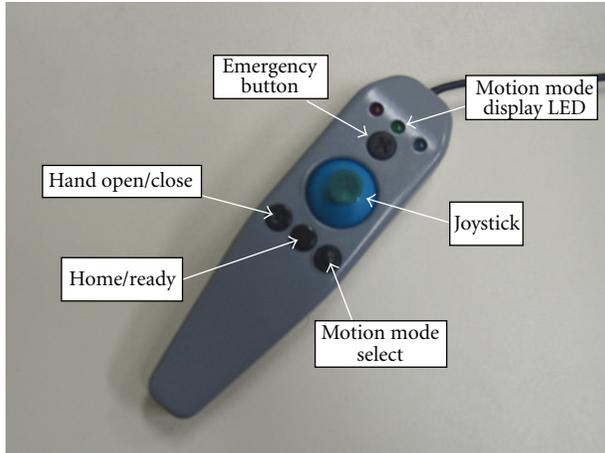


FIGURE 1: A joystick to control a manipulator.

For example, the iARM (Assistive Robotic Manipulator) [6] was developed to be attached to the side of wheel chairs or bed in order to perform its user's task. The iARM includes several user interfaces: keypad, joystick, and single-switch. The patients with muscular dystrophy can operate the manipulator with these user interfaces according to their conditions. However the information display device of the iARM is only a 7×5 matrix type indicator which is placed on the robot arm shoulder. Before using the manipulator, the single switch user must know all the presented characters shown on the indicator and also the state transitions for the control mode of the manipulator.

Now we are developing a portable manipulator system for patients with less muscle power such as muscular dystrophy patients. The authors are developing some types of user interface for the system. One of them is a single switch with a scanning menu panel. The single switch with a scanning menu panel is an important method for disabled people [1, 7].

The single switch with autoscan panel is remarkably important for the upper-limb disability people. In [1], the indirect selection interfaces have importance for the disabled people. There are some selection modes of the auto-scan menu, and the row-column scanning is most general way. We adopt this row-column scanning way.

These products for the minority users (e.g., upper-limb disabled patients) are called the "orphan products [8]." The orphan products are severely desirable for the specific users with specific symptoms but it is difficult to make industry products because the turnover is not expected. Because the symptoms of the patients are varied, the variation of the user interface should be large. The interface should be configured and reconstructed independently to each user. We will introduce our user interfaces and its improvements for the support robot. To offer the support robot to the market, we consider standard software components of the robotic system, RT-Middleware.

In this paper, we report on the development of the single-switch user interface with RT-Middleware [9]. The conditions of patients, whose behavior is restricted, are

varied. Therefore the operational input for the robot should have many types and grades. These multiple types of input method can be adopted simply through RT-Middleware because RT-Middleware is a common software platform for robot systems. We developed some types of user interface, such as a keypad, a joystick (Figure 1), and a single-switch with autoscan panel (Figure 2).

Two patients tried the single-switch user interface and made comments about it. Then we report the improvement of the single-switch user interface according to the comments. Our user interface is constructed with a single switch and input button panel on the computer console. This button panel is an autoscan type and adopts a row-column scanning way [10]. The input panel was rigid, and the user could not change its configuration as he/she likes. The first improvement enables a user to customize the input panel configuration. For the user's convenience, we added a new function to replace the buttons' location in the panel by the user him/herself only with the single switch. Furthermore, we add alternative scanning way of the button panel. Through a new scanning way, the user can put the scanned button forward by a single click of the single switch. And the user can select the button by double click of the single switch.

The precise evaluations of the robot system are reported on [11]. In this paper, we discuss the basic evaluation of the system and the improvements' examples of the user interface to offer our "orphan products" to the real users' daily life.

In the following section, we discuss RT-Middleware and the OpenRTM-aist implementation of it. In Section 3, we introduce the user interface system constructed with RT-Middleware. In Section 4, we describe the operation and evaluation by the patients for the user interface. In Section 5, we describe the improvement of the user interface according to the evaluation by the patients for the user interface and finally conclude this paper.

2. RT-Middleware

RT-Middleware is developed by AIST as an open robot software platform. "RT" means "Robot Technology," which is applied not only to the industrial field but also to the nonindustrial field such as human daily life support systems. RT-Middleware is a software platform for RT systems. RT-Middleware aims at establishing a common platform based on distributed object technology that supports the construction of various networked robotic systems by the integration of various network-enabled robotic elements called RT-Components. If the communication protocol between RT-Components is unified, any other RT-Components can be replaced with them. Modularization of RT elements and RT-Middleware has been studied and developed at AIST, which promotes the application of RT in various fields. RT-Middleware may raise the efficiency of robotic study and development, expand the RT target field, and make a new robot market [12].

2.1. RT-Component Architecture. An RT-Component is the basic functional unit of RT-Middleware-based systems.

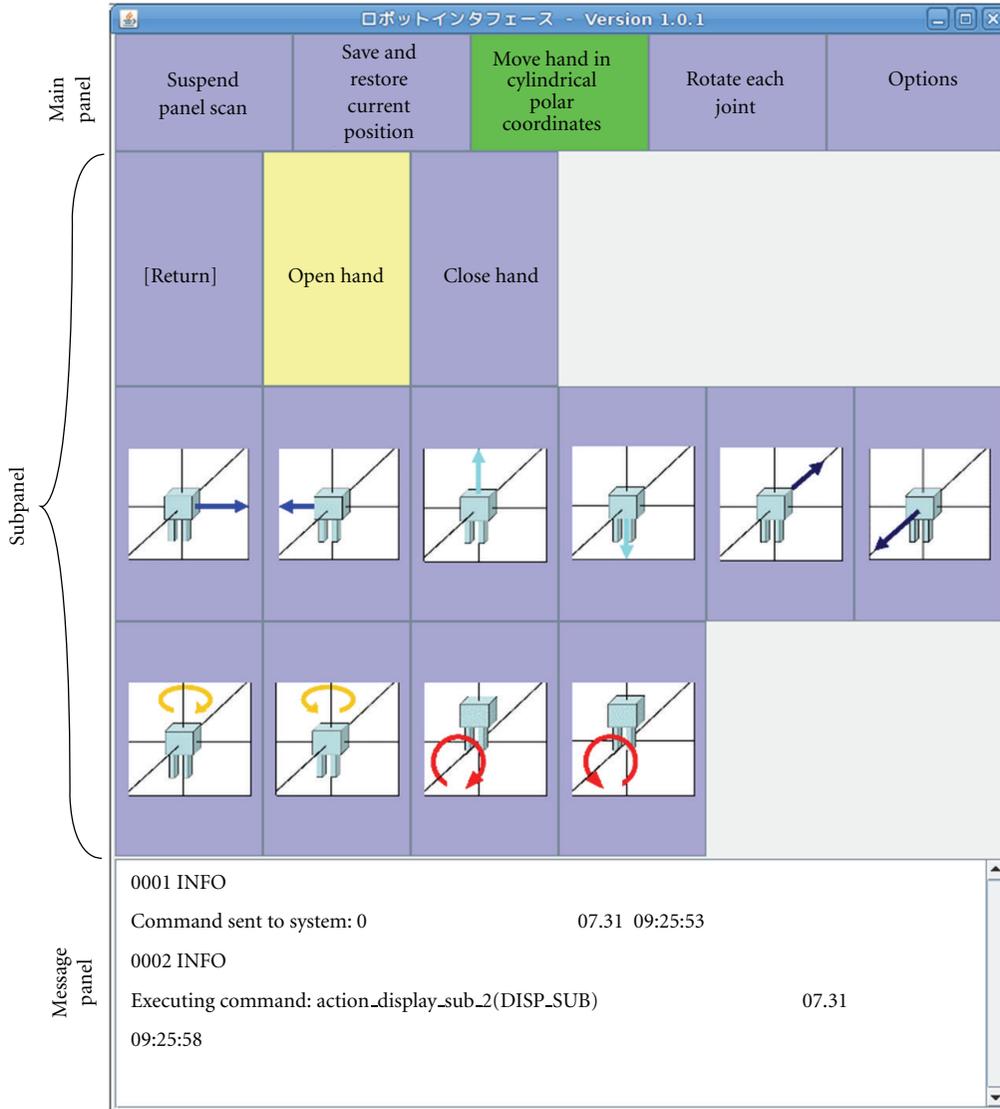


FIGURE 2: An autoscan panel for a single-switch to control a manipulator.

Figure 3 shows the architecture block diagram of an RT-Component.

For platform independencies, CORBA (common object request broker architecture) is adopted as the distributed object middleware and is used for modeling RT-Components. An RT-Component consists of the following objects and interfaces:

- (i) component object,
- (ii) activity,
- (iii) inPort as input port object,
- (iv) outPort as output port object,
- (v) servicePort as service interface,
- (vi) configuration interfaces.

The general distributed object model can be described as some interfaces that contain operations with parameters

and a return value, for example, a torque command and an encoder value. On the other hand, the RT-Component model has a component object as the main body, activity as the main process unit and input ports (InPort), and output ports (OutPort) as data stream ports.

2.2. OpenRTM-aist. “OpenRTM-aist” is developed and distributed by AIST as an implementation of the RT-Middleware interface specification and RT-Component object model [13, 14]. “OpenRTM-aist” consists of an RT-Component development frame work, manager, and a set of tools.

Because RT-Middleware aims to improve the reuse of software, our single-switch interface component and robotic simulator component can be replaced with the joystick component and real manipulator control component, respectively.

Furthermore, when many types of user interfaces are developed to correspond to the patient’s symptoms, these

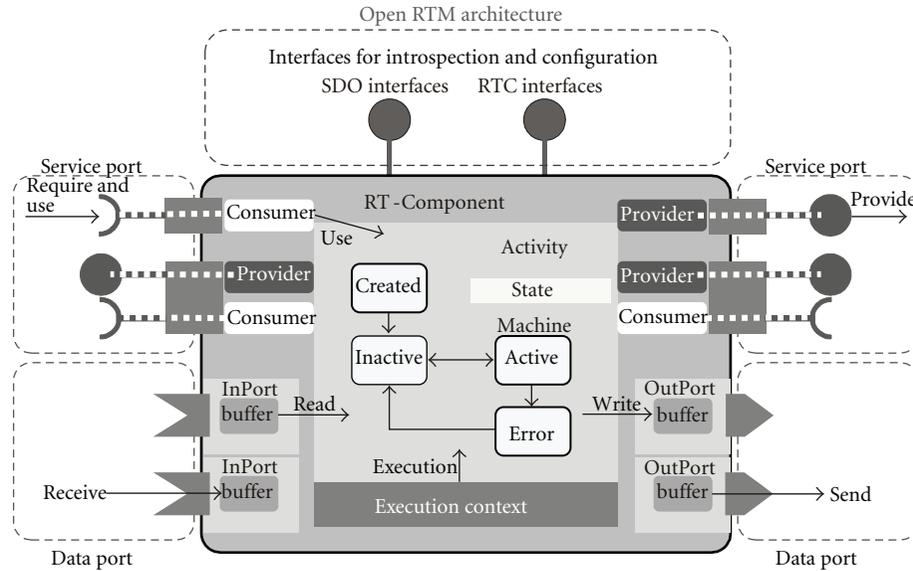


FIGURE 3: Architecture of RT-Component. RT-Component has a component object, command interface, an activity, InPorts, OutPorts, and service ports.

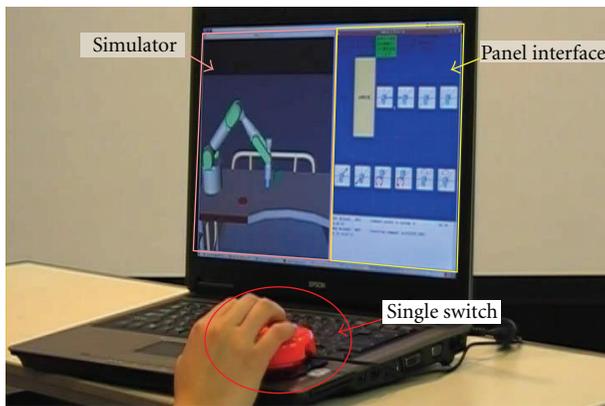


FIGURE 4: Overview of user interface and robot components with a single-switch.

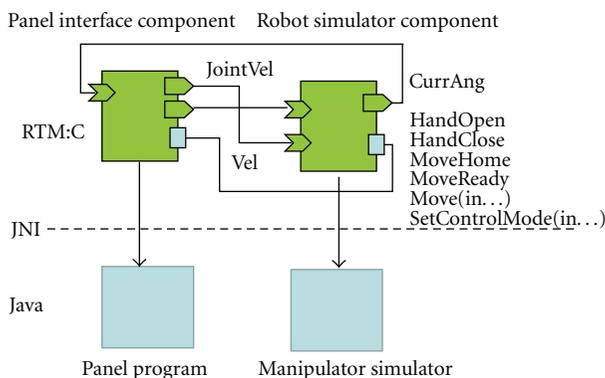


FIGURE 5: Connection of components.

new components can be connected to the original components without any change in them and used, if the communication protocol of the components is the same.

3. User Interface System

3.1. An Example of Single-Switch Interface. Single-switch user interfaces are used by disabled people to control computers [8]. There are several commercial software packages [15, 16] for the single-switch. For example, “Den-no-shin” is one application. In den-no-shin, the computer shows a panel of buttons on the console. The buttons’ color changes in order (scanning), and the user can select the highlighted button to push the single-switch. This software enables the disabled people to input literals, to speak the input sentences, to start Windows applications, to control the mouse cursor with the cross-cursor on the console, and to click the mouse cursor.

However there are few single-switch user interface systems to control a robot. Control of a robotic wheelchair by single-switch scanning has been reported in [16]. But the user can only give commands to change the direction of the wheelchair. We developed a single-switch scanning user interface component to control a robot arm.

3.2. An Outline of the System. User interfaces for disabled people have many variations because their symptoms are varied and the input methods available to them are restricted. Even if the user interfaces are varied, the communication protocol of the user interface and the robot components can be decided in advance. Under the protocol, many kinds of user interfaces and robot components can be connected uniformly as RT-Components [9].

The overview of the user interface and the robot simulator on the computer console with the single-switch is shown

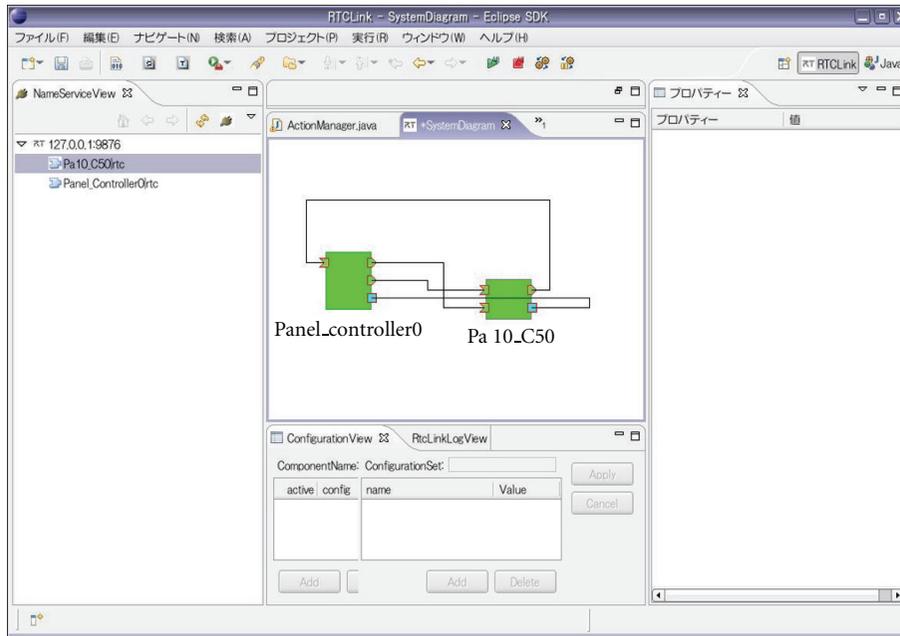


FIGURE 6: Components on rtc-link.

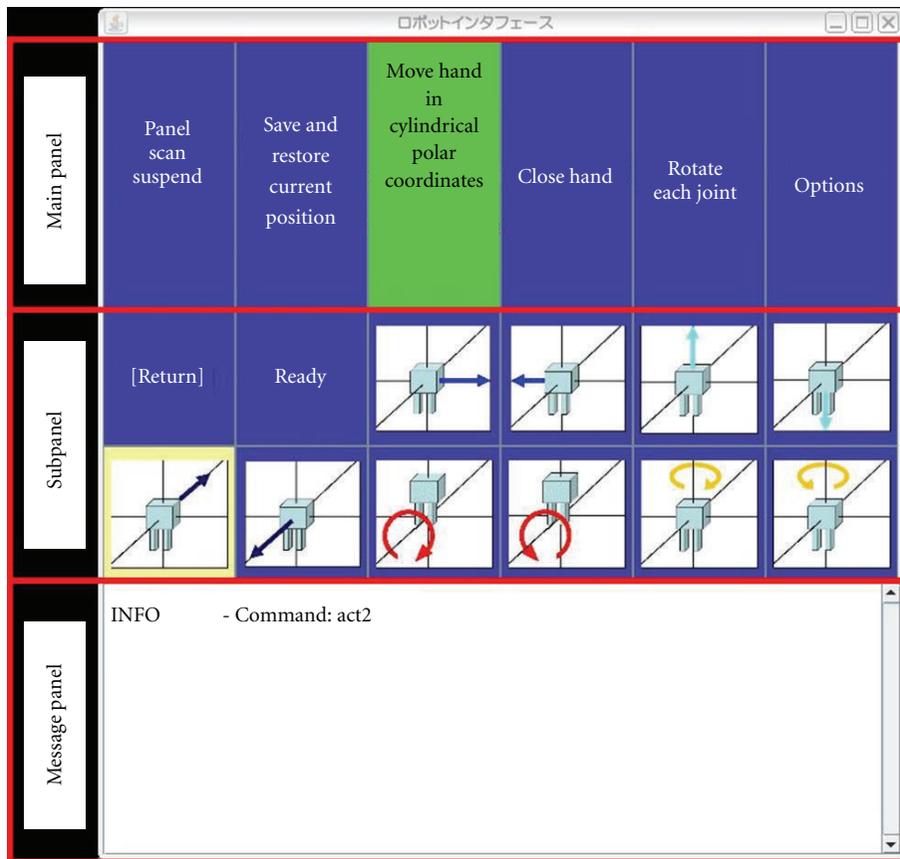


FIGURE 7: Input panel for single-switch scanning and 5 DOF arm.

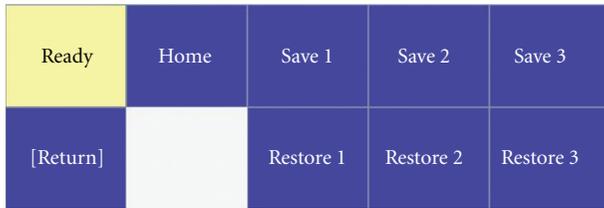


FIGURE 8: Subpanel for save and restore functions.

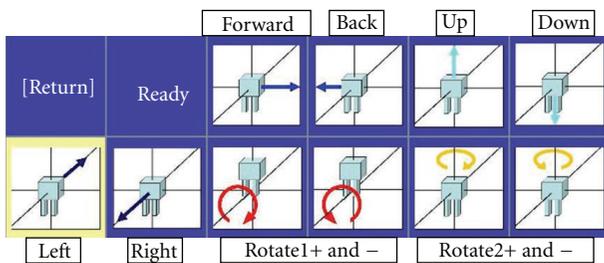


FIGURE 9: Subpanel for move command buttons.

in Figure 4. The user interface consists of a single-switch and scanning menu panel interface. A manipulator simulator controlled by the user interface is shown on the left of the panel interface.

The panel interface and the robot simulator are created from the interface component and the simulator component, respectively.

3.3. Robot Component. The communication protocol of the user interface and robot components depends on the input/output data of the robot component. The robot component should have the following functions:

- (1) manual control of robot motion,
- (2) move to home position,
- (3) move to ready position,
- (4) hand open/close,
- (5) servo on/off.

The robot arm has two control modes. One of them is a coordinate control mode. In this mode, the coordinates of the robot hand can be controlled. Another control mode is a joint mode. Joints of the robot arm can be controlled.

The home position is the closing configuration of the robot arm. The ready position is the initial position of the robot arm to start to move.

The robot hand can open and close to grasp an object.

The robot arm can suspend and restart with the servo on/off function.

In this paper, the robot arm has 5 or 6 degrees of freedom. The coordinate commands are given as cylindrical polar and Cartesian coordinates.

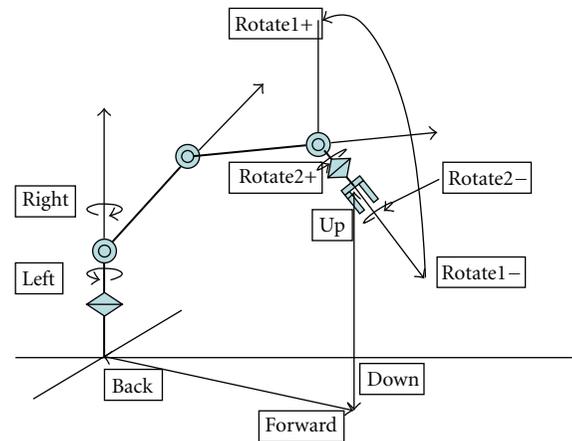


FIGURE 10: Directions of hand in the simulation.

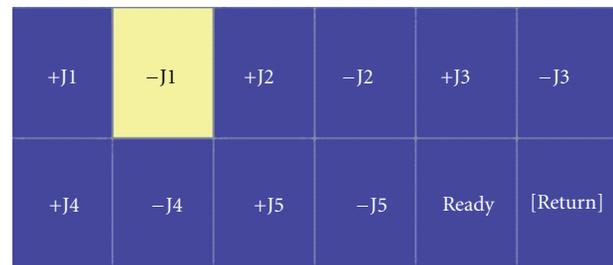


FIGURE 11: Subpanel for joint command buttons.

3.4. Communication Protocol. To realize the above mentioned functions in the robot component, we decide the communication protocol between the robot component and the user interface component as follows. The data of the output ports of the user interface component are as follows:

- (1) the reference velocity of the robot hand,
- (2) the reference joint angles.

The robot component receives the above data and moves along with them. The data of the input port of the user interface component is as follows:

- (1) the current joint angles of the robot arm.

The user interface component can record the configuration of the robot with the data.

The service port is used for asynchronous commands. The service port commands from the user interface component to the robot component are as follows:

- (1) servo on-off,
- (2) set of the control mode,
- (3) move to target joint angles,
- (4) move to home position,
- (5) move to ready position
- (6) hand open-close.

A precise diagram of component connections is shown in Figure 5.

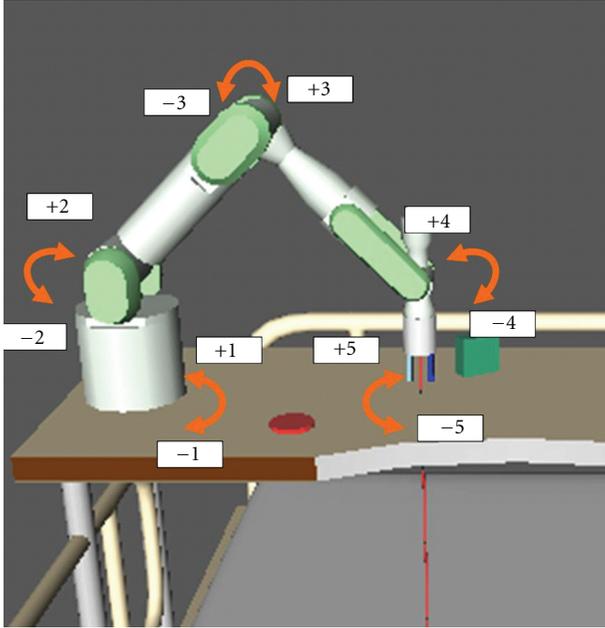


FIGURE 12: Rotations of joints in the simulation.

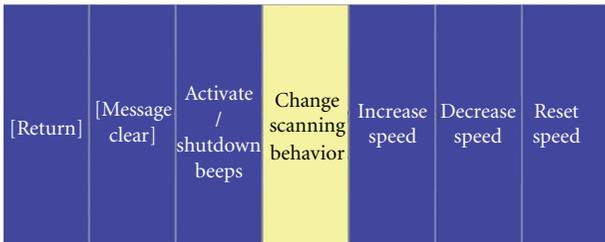


FIGURE 13: Subpanel for option buttons.

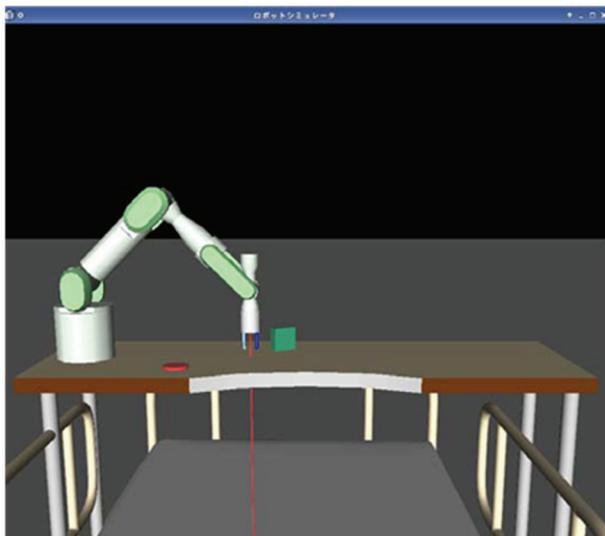


FIGURE 14: Moving robot arm in simulation.



(a)



(b)

FIGURE 15: Tester’s single switches.

RT-Components are connected with “rtc-link.” The rtc-link is a GUI tool that manages connections of InPort/OutPort and Service Ports between RT-Components and performs activation/deactivation of an RT-Component. Rtc-link is a powerful tool which can be used when development and debugging of RT-Components is performed. Moreover, it can also be used in verification and experimentation with a robot system, performing the low level integration of components. The rtc-link is shown in Figure 6.

Each component is described and executed in Linux, and both components can be executed on Windows with VMware.

3.5. Panel Interface. Our operational user interface includes a button panel shown on the computer monitor. The buttons on the panel change their color in order (scanning in order). When the target button is highlighted, the user can select it with single-switch pushing and control the robot arm (Figure 7).

The panel includes three parts. They are a main panel, a subpanel, and a message panel. The user can command the robot by selecting the button on a main panel and on a sub-panel. The message panel shows the executed actions result and the errors.

The main panel has six buttons. The first button is to start and suspend panel scanning. The second button is to save and restore the robot hand’s current position. The third button is the move command using the robot hand coordinates.



FIGURE 16: Option panel.

The fourth button is open-close of the finger of the robot arm. The fifth button is move command of the robot's joints' angle. The sixth button corresponds to options.

(1) *Scan/Suspend Button*. When this button is selected, scanning of the panel is suspended. If the single-switch is pressed again, scanning restarts.

(2) *Save/Restore Button*. This button opens a subpanel to save the current arm position and restore the position. The subpanel is shown in Figure 8. The subpanel contains buttons to move to ready position and home position.

(3) *Coordinate Move Command Button*. This button opens a subpanel for the command button of the robot's hand coordinates (Figure 9). In the beginning of our project, the target manipulator was taken to have 5 degrees of freedom.

There are 10 commands. The user can select a button among these 10 buttons to indicate the direction of the robot hand's motion.

The selected buttons control the robot hand to move in the directions in Figure 10. The directions and rotations of the robot hand motions are the same in the appearance of the simulation and the buttons' characters in the panel.

These buttons are toggle buttons. When the single-switch is pressed and the button is selected, the robot starts to move. Then the single-switch is pressed again, and the robot stops.

(4) *Hand Open/Close Button*. When this button is selected, the hand closes if the hand already is open. When this button is selected again, the hand opens.

(5) *Joint Command Button*. This button opens a subpanel of the joints' angle command of the robot arm (Figure 11).



FIGURE 17: Button swap in panel.

The user can command 5 joints' rotations in plus and minus directions. In Figure 12, the rotations of all the joints of the manipulator are shown.

These buttons are toggle buttons. When the single-switch is pressed and the button is selected, the robot starts to move. Then the single switch is pressed again, and the robot stops.

(6) *Options Button.* This button opens a subpanel for options to change the panel settings (Figure 13). Buttons to clear the literals in message panel, to set the on/off of the beep sound of panel scanning, to set the scanning order when the scanning restarts, and to change the scanning speed of the buttons are shown in this panel.

All the button actions can be recorded in a log file with their executed time.

Our panel user interface is written in Java. The C++ program calls the panel user interface program through JNI as

shown in Figure 4. Our system is developed with OpenRTM-aist-0.4.0.

3.6. *Robot Arm Simulator.* We constructed a robot simulator to evaluate the single-switch scanning user interface (Figure 14). The robot arm moves in the simulator window according to the commands from the operation of the single-switch scanning interface.

We developed a simple task in the simulator. The task is to let the robot grasp the green box and place it on the red circle. The initial position of the green box and the red circle are alterable with a configuration file for the simulator.

In our simulator, the contact between the robot hand and the grasping object or the environment is not calculated. When the hand closes near enough to the target object, the simulator decides that the robot has grasped the object and

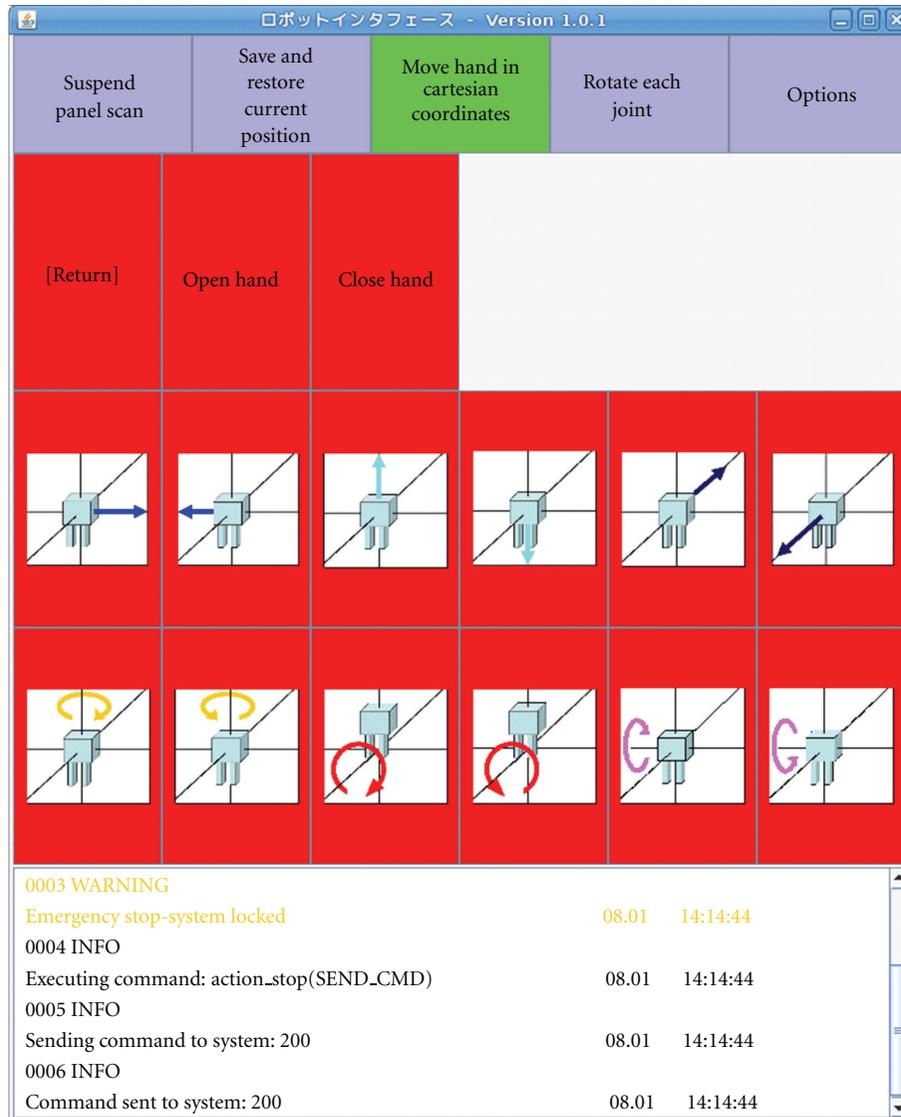


FIGURE 18: Input panel at emergency stop.

draws the object between the fingers of the robot as the grasping is completed.

The image of the simulator is drawn on a plane. Therefore the user cannot understand the robot hand position precisely in the depth direction. A red line like a laser pointer is drawn from the center of the hand on the vertical line of the hand to show the relative position of the hand in the task environment.

The simulator is written in Java similar to the user interface panel and the C++ program calls it through JNI.

4. Operation and Evaluation by Users

We cooperated with the Shimoshizu national hospital, Japan. The hospital has a special ward for muscular dystrophy patients. Two patients made trials of our user interface and

simulator for 30 minutes. Before the trials, we investigated about human ergonomic experiment and applied for the ethics committee in the AIST. Both of them usually used single-switch scanning user interface to control computers so they are accustomed to the single-switch scanning user interface. Their own single switches were connected to the user interface computer through a USB connection (Figure 15).

First, examiners simply explained about the user interface and the simulator and demonstrated the task. Then each subject separately performed the task of taking the box and placing it on the target position in the simulator. During the task execution and after the task, the examiners asked subjects about general impressions, contents of buttons, scanning speed of buttons, safety issues, and their desire to control robot arms with this user interface. Their comments are shown in Table 1.

TABLE 1: Comments by subjects.

	Subject A	Subject B
Evaluations of easiness of operations	At first I was afraid that it may be difficult to control the robot. I feel easier to use it.	When the new robot will appear with this interface, I want to use them very much. This robot will be much better than to call a helper.
Evaluations and desire about the panel	Configuration of the button on the panel seems good, but the contents of the button are difficult to understand.	I want to configure the scanning speed. The scanning can be continued anytime.
	Scanning of the button should start at the beginning of the list when the scanning restarts.	The number of the buttons on the panel seems to be necessary and sufficient. If I can select one of the many types of the panels, it seems preferable.
	The scanning should be stopped unless the user wants to use this interface.	Beep sound at scanning of button is preferable.
Request about safety	I want the emergency stopping button to push the single switch for a long time or in the panel	It is better for me if the single switch involves emergency stopping function. If the system can find the freezing of the computer and let the robot stop, I can feel safer.

Both subjects stated that this user interface is mainly good to use and they want to control a real robot with this user interface by all means. They also commented on safety functions for the user interface.

In this evaluation, only the move command of hand subpanel is used by the examiner and subjects to simplify the evaluation. Subpanels described in the previous section have been refined using the patients' comments [17].

This evaluation is basic to improve our user interface. Precise human ergonomic evaluations with real manipulator "RAPUDA" are reported in [11]. The examinees responses are generally good. The evaluations by real users are the Peg-board task in [11]. The test and evaluation in their daily life should be performed for the next step.

5. Improvements of User Interface

5.1. Request from Patients of Muscle Dystrophy. As stated in the above section, we let 2 patients of the muscle dystrophy test our user interface. One of the patients gave us a following comment as "If I can select one of the many types of the panels, it seems preferable." (Table 1).

The previous input panel is rigid as shown in Figure 7. The panel configuration is described in the initial configuration file.

The programmer of the system can rewrite it and reconfigure the panel configuration. But the user of the user interface is not supposed to reconfigure the panel by him/herself alone.

Reconfigurable function of the input panel can be more convenient than other types of the input panel for the user.

Another patient told us "If scanning of buttons does not continue automatically and is commanded by myself, it seems preferable." (Table 1).

Both of the patients expressed the necessity of the emergency stop function. Besides, patients' comments are

as follows, "Scanning of the button should start at the beginning of the list when the scanning restarts" and "The scanning should be stopped unless the user wants to use this user interface." (Table 1).

5.2. Button Replacement Function. Therefore we made new function so the user can replace the location of the buttons in the input panel freely and made the replace button in the option panel (Figure 16).

When the user selects the button, the user interface will enter to the button replace mode. The next selected button with the single switch click and the following selected button are replaced in the panel (Figure 17). This panel is utilized for the manipulator with the 6 degrees of freedom motion.

To escape from the replace mode, the long click of the single switch is utilized.

The result of the replacement of the buttons by this function can be stored in the initial configuration file of the input panel.

This function allows the user to change freely the location of the button which he/she use usually.

5.3. Alternative Mode for Scanning and Selection of Buttons. We made alternative scanning mode for the user to control the button scanning according to his/her will. The user can command one button scanning with single click and can select the button with double click. The alternative scanning mode button is prepared in the option panel (Figure 16).

The user can let the buttons scan by him/herself with this function. The options which the user can select become rich.

5.4. Emergency Stop with Long Click. The emergency stop function is prepared with a long click of the single switch.

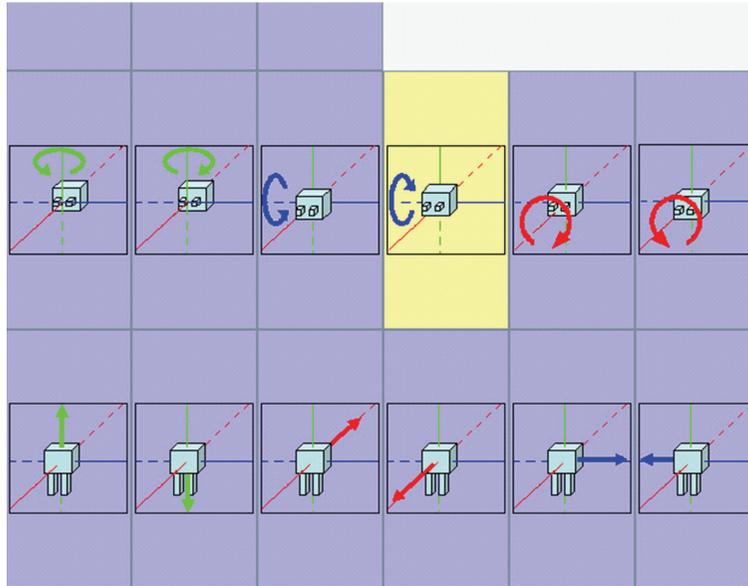


FIGURE 19: Scanning button replacement according to the previous input.

The long click makes the color of all the panel red and sends a stop command to the robot component (Figure 18).

The long click is used to escape from the replace mode too. During the replace mode, the robot does not move anyway. Therefore the user will not be confused [15].

The long click time can be changed. In this system, 1 second is indicated. The time should be decided according to the robot speed and autoscan speed.

5.5. Changing Position of Scanning Restart. The restart position of the scanning can be set at the first in the button list or the current button. The toggle button of this option is set in the options panel (Figure 16).

5.6. Suspend Button of Scanning. The suspend panel scan button is set on the top of the main panel. This button was added because of the patients' request (Figure 7).

5.7. Scanning Button Replacement according to the Previous Input. The order of the panel buttons are exchanged automatically according to the user's previous operation. The order is exchanged in 2 steps. At first the parallel motion low and rotation motion low are exchanged. In the low, previous command set (+ and -) are located on the top of the low. Figure 19 shows the replaced low and position of button sets in the scanning panel (Figure 19).

With this improvement, the user can easily repeat a desirable command's + and -. 2 trained operators have performed pick and place task with this user interface. The average total time of the task is decreased 6%.

6. Conclusion

The supporting robot for disabled people cannot make large industry, so we should offer inexpensive manufacturing way.

These type products are called as "orphan products." In order to support disabled people especially with less muscle strength such as muscular dystrophy patients, we developed a single-switch scanning user interface to control a manipulator system. The input menu panel is designed to perform various actions of the robot arm with the single switch and to be easy for users to understand. All the components of the user interface and the robot simulator are constructed with RT-Middleware. Therefore other new user interface components can be reconnected to our system without any change in them and utilized if the communication protocol of the components is the same.

Patients with muscular dystrophy tested the user interface and made evaluations. Both patients stated that this user interface is mainly good to use, and they want to control a real robot with this user interface by all means.

According to the testers' comments, we improved a single-switch scanning user interface to control a manipulator system. The first improvement enables a user to customize the input panel configuration. We added the function to replace the buttons' location in the panel by the user him/herself only with the single switch.

Furthermore, we have added an alternative way of scanning the button panel. Through the new scanning, the user can put the scanned button forward by a single click of the single switch. And the user can select the button by double clicking of the single switch.

Besides, we added an emergency stop function, an option for changing position at scanning restart and a scanning suspend function.

Finally, we made an optional improvement that the order of the panel buttons are exchanged automatically according to the user's previous operation.

These new input functions which involve user configuration can make our user interface system more user friendly.

In this project, we will synthesize the user interface and the real manipulator for the result. Moreover, this simulator can be a training kit to control a service robot for use in daily life.

Acknowledgments

The authors appreciate the cooperation of the patients and staff of the Shimoshizu National Hospital, Japan in this research. Also we would like to thank the staff of the AIST project for the personal robot. This research was carried out as one of UCROA (User Centered Robot Open Architecture) project at AIST [12].

References

- [1] A. Cook and J. Polgar, *Assistive Technologies, Principles and Practice*, Mosby Elsevier, Baltimore, Md, USA, 3rd edition, 2008.
- [2] J. Sijs, F. Liefhebber, and G. W. R. B. E. Römer, “Combined position & force control for a robotic manipulator,” in *Proceedings of the 10th IEEE International Conference on Rehabilitation Robotics (ICORR '07)*, pp. 106–111, Noordwijk, The Netherlands, June 2007.
- [3] M. J. Johnson, G. A. Di Lauro, M. C. Carrozza, E. Guglielmelli, and P. Dario, “GIVING-A-HAND: early development of a small, countertop, mobile robot kitchen assistant for elderly and medium- to high-disabled persons,” in *Proceedings of the 8th International Conference on Rehabilitation Robotics (ICORR '03)*, pp. 120–123, April, 2003.
- [4] M. Nokhtari, B. Abdulzarak, M. A. Fki, R. Rodriguez, and B. Grandjean, “Integration of rehabilitation robotics in the context of smart homes: application to assistive robotics,” in *Proceedings of the 8th International Conference on Rehabilitation Robotics (ICORR '03)*, pp. 5–8, April, 2003.
- [5] G. Verburg, M. Milner, S. Naumann, J. Bishop, and O. Sas, “An evaluation of the MANUS wheelchair-mounted manipulator,” in *Proceedings of the RESNA International '92*, pp. 602–604, 1992.
- [6] <http://www.exactdynamics.com/>.
- [7] J. Angelo, “Factors affecting the use of a single switch with assistive technology devices,” *Journal of Rehabilitation Research and Development*, vol. 37, no. 5, pp. 591–598, 2000.
- [8] <http://www.rehab.go.jp/ri/event/html/english.html>.
- [9] K. Nagata, Y. Wakita, N. Yamanobe, and N. Ando, “Development of assistive robotic arm system with Robot Technology Middleware (RTM),” in *Proceedings 19th IEEE International Workshop on Robot and Human Interactive Communication (Ro-man '10)*, pp. 737–742, 2010.
- [10] Y. Wakita, N. Yamanobe, K. Nagata, N. Ando, and M. Clerc, “Development of user interface with single switch scanning for robot arm to help disabled people using RT-middleware,” in *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV '08)*, pp. 1515–1520, Hanoi, Vietnam, December 2008.
- [11] N. Yamanobe, Y. Wakita, K. Nagata, M. Clerc, T. Kinose, and E. Ono, “Robotic arm operation training system for persons with disabled upper limbs,” in *Proceedings 19th IEEE International Workshop on Robot and Human Interactive Communication (Ro-man '10)*, pp. 749–754, 2010.
- [12] AIST 2005/2006 Annual Report.
- [13] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, and W. -K. Yoon, “RT-Middleware: distributed component middleware for RT (Robot Technology),” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, pp. 3555–3560, Edmonton, Canada, 2005.
- [14] <http://www.openrtm.org/openrtm/en>.
- [15] K. Ozawa, “Barrier free and accessibility of information processing equipment,” *The Journal of the Institute of Electronics, Information, and Communication Engineers*, vol. 81, no. 2, pp. 166–169, 1998 (Japanese).
- [16] H. A. Yanco and J. Gips, “Driver performance using single switch scanning with a powered wheelchair: robotic assisted control versus traditional control,” in *Proceedings of the Annual Conference of the Rehabilitation Engineering and Assistive Technology Society of North America*, pp. 298–300, RESNA Press, Minneapolis, Minnesota, June 1998.
- [17] Y. Wakita, N. Yamanobe, K. Nagata, and M. Clerc, “Customize function of single switch user interface for robot arm to help a daily life,” in *Proceedings of the 10th IEEE International Conference on Robotics and Biomimetics (ROBIO '08)*, pp. 945–950, Bangkok, Thailand, 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

