

# CS CAPSTONE TECHNOLOGY REVIEW

21 NOVEMBER 2017

## ROBOTIC WHEELCHAIR DATA COLLECTION AND ANALYSIS

PREPARED FOR

OREGON STATE UNIVERSITY

MATTHEW WILLIAM SHUMAN

PREPARED BY

GROUP 4

WHEELCHAIR DATA COLLECTION TEAM

AARON LEONDAR

### Abstract

This document will serve to provide the Technology Review for the Robot and Application Communication section of the Wheelchair Data Collection capstone project under Matthew Shuman. The document provides a high-level description of the technologies used in the communication between the robot and the computer application, how they compare to alternatives, and why the specific technologies we are using were chosen. This document does not discuss any of the technologies used in any part of the project outside of communication between the robot and the computer application, and the data transferred between the two.

## CONTENTS

<b>1</b>	<b>Robotics Middleware</b>	<b>2</b>
1.1	Overview and Criteria . . . . .	2
1.2	Potential Choices . . . . .	2
1.2.1	Choice One: ROS . . . . .	2
1.2.2	Choice Two: OpenRTM-aist . . . . .	2
1.2.3	Choice Three: Orocos . . . . .	2
1.3	Discussion . . . . .	2
1.4	Conclusion . . . . .	3
<b>2</b>	<b>Data Storage</b>	<b>3</b>
2.1	Overview and Criteria . . . . .	3
2.2	Potential Choices . . . . .	3
2.2.1	Choice One: ROSbags . . . . .	3
2.2.2	Choice Two: OpenRTM Ports . . . . .	3
2.2.3	Choice Three: Orocos Buffer Ports . . . . .	3
2.3	Discussion . . . . .	3
2.4	Conclusion . . . . .	3
<b>3</b>	<b>Network Communication</b>	<b>4</b>
3.1	Overview and Criteria . . . . .	4
3.2	Potential Choices . . . . .	4
3.2.1	Choice One: Wireless . . . . .	4
3.2.2	Choice Two: Wired Ethernet . . . . .	4
3.2.3	Choice Three: Wireless Bluetooth . . . . .	4
3.3	Discussion . . . . .	4
3.4	Conclusion . . . . .	4
	<b>References</b>	<b>5</b>

# 1 ROBOTICS MIDDLEWARE

## 1.1 Overview and Criteria

The middleware used in communicating with the robot is one of the most important aspects of the entire project, as it provides a means to send and receive data directly to and from the robot. More generally, the robotics middleware manages the heterogeneity of the hardware, as well as the applications, as well as assisting in the integration of new technologies, and make it easier to maintain the integrity of the robot's architecture.[1] Choosing the right middleware allows us to be able to effectively communicate with the robot. To be chosen as the Robotics Middleware that we use, it must be able to effectively communicate and send data between the robotics system on the wheelchair and the application that the tester is running. The middleware that will be chosen must also contain the ability to read sensor inputs, distance inputs, and amount of elapsed time, and be able to store them all in a single data packet, which will then be sent wirelessly to the application on the computer.

## 1.2 Potential Choices

### 1.2.1 *Choice One: ROS*

At the most basic of levels, ROS is a system that allows messages to be passed, which provides inter-process communication. It also provides a message passing system that supports recording and playback of messages. It can easily read data from a sensor, send that data to a file, and then be able to access and republish the data in that file at any point in the future. ROS also provides basic robot implementation from the get-go, from Standard Robot Message formats such as vectors, transforms, maps, and odometry, to Sensor Message formats used for cameras and lasers.[2]

### 1.2.2 *Choice Two: OpenRTM-aist*

OpenRTM exists to integrate a variety of robotic functions as a single software. To do this, it modularizes robot functional elements as software components called RT-Components. These RT-Components are combined on RT-Middleware, then are implemented into the robot. Components have data and service ports to communicate, and contain a common state machine inside of them.[3]

### 1.2.3 *Choice Three: Orocos*

Orocos is a middleware that is much more advanced and precise than either ROS or OpenRTM. The main applications of Orocos are use in real-time control machines, such as autonomous cars and visual tracking, as well as a real-time logging system that is low-overhead and usable throughout an entire system.[4]

## 1.3 Discussion

In comparing ROS to Orocos, ROS is better for quick, easy robot control, in contrast to Orocos, which is used for real-time control and not much else. ROS implements its properties in a specific central server, whereas in Orocos, every property is its own separate component.[5] ROS and OpenRTM are very similar in concept, however the main problem with OpenRTM is that documentation of it is very poor in any language outside of Japanese, which means as a result the middleware itself is not popular outside of Japan. It could still be used, but would require quite a bit more research and legwork for roughly the same thing as ROS.[6]

## 1.4 Conclusion

We chose ROS, partially due to already having a working ROS implementation that can be used on the wheelchair, and because ROS provides an easy and effective way to read sensor data, store that data to a file, then read and republish that file at a later date. ROS also has a large amount of reference material available, both on the internet and in textbooks, making it a lot more accessible to learn than OpenRTM or OrocOS.

## 2 DATA STORAGE

### 2.1 Overview and Criteria

Another big part of this project is to be able to read a lot of different data and store it to a place where it can be accessed in the future. The data storage method must be compatible with ROS since ROS is the middleware that has been decided upon. It also must be able to read sensor data such as odometry and touch sensors, and store that data along with time elapsed data in a way so that it can be accessed and republished later.

### 2.2 Potential Choices

#### 2.2.1 Choice One: ROSbags

The ROSbag, or more simply, just the bag, is a file format that is used within ROS to store, process, analyze and visualize messages. The bags are able to store several different types of messages into one single bag. Each individual bag is contained within its own file, and can be accessed and republished at any time.[7]

#### 2.2.2 Choice Two: OpenRTM Ports

OpenRTM uses buffers to send data through ports, and then that data is stored within the buffers. It can utilize basic read and write functions. Additional information regarding data storage using OpenRTM is limited or unavailable.[8]

#### 2.2.3 Choice Three: OrocOS Buffer Ports

OrocOS uses buffers to send data through ports in real-time. It can utilize basic read and write functions. Additional information on data storage using OrocOS is extremely limited.[9]

### 2.3 Discussion

OpenRTM Ports and OrocOS Buffer Ports accomplish roughly the same things, in contrast to bags, which accomplish everything that the other two can do, but also can read and store sensor data, which is essential for the project. Also, information regarding sensor reading and storing sensory data is limited or nonexistent for both OpenRTM and OrocOS, in contrast to ROS which has an abundance of information for sensor data storage in the form of ROSbags.

### 2.4 Conclusion

We chose ROSbags, mainly due to them being a part of ROS, but also because they are able to store touch sensor data, odometry sensor data, and time-elapsed data all in one "bag". These "bags" can also be re-accessed at a later date. ROS also is advantageous over the other two because there is already a ROS environment set up on the Permobil wheelchair

### 3 NETWORK COMMUNICATION

#### 3.1 Overview and Criteria

In order to send and receive data between the robot and computer application, a network connection must be established between the two. To be a connection that would be considered for use in this project, the network must be reliable and be able to support sending data between the Robot and the application multiple times. The network should also be accessible while the robot is running trials.

#### 3.2 Potential Choices

##### 3.2.1 *Choice One: Wireless*

Wi-Fi is the leading method of connecting an enabled device to the internet using a wireless connection, usually a wireless network adapter or wireless router. Using a wireless router would allow the computer application to communicate with the network on the robot from a great distance, and could allow the tester to not have to move while the robot moves around during its trials.

##### 3.2.2 *Choice Two: Wired Ethernet*

The complete opposite of a Wi-Fi connection is a wired network connection, most commonly used with a ethernet cable plugged between the robot and the network router. Using a wired ethernet connection would allow for a faster communication speed between the robot and the application. Using a wired connection would also mean that any kind of wireless interference would be negligible.

##### 3.2.3 *Choice Three: Wireless Bluetooth*

Bluetooth is an open standard specification for a radio frequency. It specializes in exchanging data over short distances using short-wavelength radio waves. Bluetooth can be used with a mobile device to communicate with a bluetooth-enabled device mounted to the robot that is able to receive data from the mobile device and can transmit data back to the mobile device. However, using bluetooth for anything more powerful than a mobile phone would not be suitable.[10]

#### 3.3 Discussion

In contrast to a wired ethernet connection, a wireless router makes the hassle of dragging around a wire non-existent. Additionally, dragging around an ethernet wire would become problematic, especially if the robot would have to traverse the same paths it had already gone through, as it could end up running over wires and possibly getting them tangled. Another glaring problem with choosing an ethernet solution would be if at any point the ethernet cord became unplugged from either the robot or the computer housing the application, the entire network becomes inaccessible. In comparing Wi-Fi to Bluetooth for wireless communication options, Bluetooth is usually used to communicate over short distances between Bluetooth enabled devices, but doesn't provide any kind of Internet access.[11]

#### 3.4 Conclusion

We chose to use a Wireless router, mostly because there was already a wireless router installed on the wheelchair that is ready to use. Though another advantage of a wireless router is that it isn't restricted by wires while the robot is moving around, so there is no risk of running over wires or getting them tangled. While not as fast at transferring data as a wired

ethernet connection, a wireless router is still more than enough to satisfy the network needs for this project. A bluetooth wireless connection could still be a viable connection, but it would require additional materials to be purchased for no foreseeable advantage over just using the Wi-Fi router that is already mounted and integrated into the wheelchair.

## REFERENCES

- [1] A. Elkady and T. Sobli. Robotics middleware. [Online]. Available: <https://www.hindawi.com/journals/jr/2012/959013/>
- [2] O. S. R. Foundation. Ros core components. [Online]. Available: <http://www.ros.org/core-components/>
- [3] N. I. of Advanced Industrial Science and J. Technology. Openrtm general information. [Online]. Available: <http://www.openrtm.org/openrtm/en/content/general-information>
- [4] O. P. Wiki. Orocos roadmap. [Online]. Available: <http://www.orocos.org/wiki/orocos/development/roadmap>
- [5] V. O. P. W. F. Users. Orocos and ros. [Online]. Available: <http://www.orocos.org/node/1117>
- [6] Chapter 2: Robot middleware. [Online]. Available: [http://www.mogi.bme.hu/TAMOP/robot\\_applications/ch02.html#ch-2.4](http://www.mogi.bme.hu/TAMOP/robot_applications/ch02.html#ch-2.4)
- [7] O. S. R. Foundation. Bags. [Online]. Available: <http://wiki.ros.org/Bags>
- [8] Openrtm-aist development. [Online]. Available: <http://www.openrtm.org/openrtm/en/book/export/html/381>
- [9] Rtt 2.0 data flow ports. [Online]. Available: <http://www.orocos.org/wiki/rtt/rtt-20/upgrading-rtt-1x-20/rtt-20-data-flow-ports>
- [10] P. Rai, B. D'souza, Y. Sandeep, Z. Shaikh, and D. Singh. Robot controlled by android application (bluetooth). [Online]. Available: <https://www.slideshare.net/PankajRai42/bluetooth-controlled-robot-60795241>
- [11] T. Struyk. What is the difference between bluetooth and wi-fi. [Online]. Available: <https://www.techopedia.com/2/27881/networks/wireless/what-is-the-difference-between-bluetooth-and-wi-fi>