Roslyn: The Tour Guide Robot

by

Jared Rhizor, Timothy Sweet, and Nishok Yadav

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements of the Degree
DOCTOR OF PHILOSOPHY
(Electrical Engineering)

May 2014

Acknowledgements

The authors would like to thank of number of people who made this project possible. First, our advisor, Dr. Dave Feil-Seifer, whose enthusiasm and support drove this project to success. Second, the University of Nevada Robotics Research Lab, including its directors Dr. Dave Feil-Seifer (again) and Dr. Monica Nicolescu for providing us a place to work and equipment to use. Third, to Dr. Sergiu Dascalu for providing guidance on the project as a whole, and especially in developing all of the specifications, diagrams, etc. in this document. Fourth, to the University of Nevada, Reno Honors Program and Office for Interdisciplinary and Undergraduate Research for providing funding for Roslyns additional hardware. Fifth, to Team 4 (Luke, Jessie, Jake, and Blake) from our 2014 CS 426 Senior Projects course, who helped at every step of development, contributing their time, expertise, food, and study partners. Sixth, to the Open Source Robotics Foundation, Willow Garage, and the entire Robot Operating System community for doing most of the heavy lifting in developing the robotic control and interface components.

Table of Contents

List of Tables

List of Figures

Abstract

The tour guide robot Roslyn is an autonomous system that can provide tours to anybody on the second floor of the Scrugham Engineering and Mines building. The robot is not only able to navigate the halls, but navigate them in a socially acceptable manner. This means it does not collide with pedestrians while in motion and takes walking speed into consideration. In order to interact with the user, there is a touch-screen display mounted on top of the moving robot base. As the robot moves throughout the hall, information appears on the screen related to the landmarks being passed (i.e. laboratories and offices). The user is also able to select a destination on the screen in order to begin the tour.

First Chapter

The tour guide robot Roslyn is a socially aware autonomous system that can lead tours of the Computer Science and Engineering Department at the University of Nevada, Reno. This requires socially acceptable navigation within the second floor of the Scrugham Engineering and Mines building. Tours of the Computer Science and Engineering Department are currently given by faculty and staff, or student ambassadors (typically from other departments). This creates some difficulties for visitors:

- A tour can only be scheduled when a faculty or staff member is available. This may conflict with class times, office hours, or other responsibilities
- Giving tours distracts faculty and staff from their normal studies, teaching, research, etc. especially with large groups
- Tours given by student ambassadors are generally limited to superficial knowledge of the department, such as the location of the department office. A typical student

ambassador would not be able to answer specific questions about a program, lab, or other facility.

A robotic tour guide could address many of these issues:

- A tour with a robot could be scheduled for any time the building is accessible, even after normal working hours. Tours could be scheduled in real-time without concerns about a faculty or staff members conflicting schedule. Tours could also be given on demand for visitors on site without an appointment

- A robot could be dedicated entirely to giving tours, and would not become distracted by other activities

- A robot could be provided up-to-date information by a departmental authority, or lookup information in real time.

Some challenges are present for a robotic tour guide in a situated academic environment, all of which shall be addressed in this thesis:

- Safe navigation: both globally and locally. A global plan must be generated for every destination which will optimize the route taken to the destination. A local plan must also be generated which avoids obstacles such as walls, items in a hallway, and moving people. This local plan must also lead the user on a reasonable path and a reasonable pace.

- Self safety: the robot must not endanger itself. Some potential safety concerns include stairwells and objects mounted outside of the robots view (ie drinking fountains, which are mounted above a laser range finders plane of view).

- Power and portability constraints: the wireless network installed in the Scrugham Engineering and Mines building is insufficient for real-time sensor data transmission, thus all processing must be performed on the robot. Thus there are constraints on the amount of power the robot can deliver to its computational components, the battery life of the robot and these components, and the maximum reasonable size of these components.

In order for the robot to be socially aware, it must not only lead its user at a comfortable distance and pace, but also keep other humans and obstacles at a safe distance. A human tour guide would not walk exceptionally fast away from their group or walk uncomfortably close to them, thus the robot should not either.

The tour guide robot is primarily composed of three distinct nodes, which are each discussed independently due to their complexity. The first node, labeled ROS Nodes in Fig. 1, is a collection of pre-existing open source ROS nodes which provide data and processing facilities to the system. These nodes are not being developed by the team, but are distributed by the Open Source Robotics Foundation.

The second node, labeled User Interface in Fig. 1, provides a user-friendly interaction between the robot and user(s). This allows the users to select a destination, as well as request and view more information about their surroundings via a touchscreen tablet on the robot.

The third node, labeled Social Navigation in Fig. 1, safely navigates the robot through the dynamic environment and ensures proper social interactions with the user.

Roslyn is composed of two independent programs:

- Roslyn is able to map an environment and use the map for future navigation. This is done the first time Roslyn is introduced to a particular building and does not have to be repeated.

- Roslyn uses the building map in its second program along with its sensors and communication pathways to communicate with the user interface and drive itself.

Roslyn is composed of a number of hardware components:

- The robotic base: a Pioneer 3DX
- An internal computer: a Raspberry Pi Model B Revision 2
- A laser rangefinder: a SICK LMS 100 or a Hokoyu URG-04LX-UG01
- A laptop: Intel Core i5-2450M @ 2.50GHz with 8 GB of RAM or better
- The user interface: a Nexus 7 2013
- A custom stand for the Nexus 7 2013

The rest of this thesis is organized as follows:...

I.1 Need Finding

A survey was distributed to the engineers developing this project, several current students, and two prospective CS students currently in high school. A total of ten individuals responded. The interview questions are included in Appendix ###.

I.1.1 Interview Findings

Almost all of the interviewees claimed to prefer voice or touch screen interfaces to the tour guide robot. Half of the individuals believed that a robot tour guide had the potential to be as good as a human tour guide. The choice between an established path and a custom path selected by the users was highly controversial. This result suggests that users may want the option to select either option. The one unanimous answer was that the robot should move at a normal walking pace. Unfortunately, our robot cannot travel at full walking speed.

Surprisingly, almost all individuals want the robot to stop at interest points and wait for the users input to continue. None selected to continue automatically, and only a couple wanted the robot to slow near the interest point. The desired humanness of the robot varied greatly across responses. For human questions, the results were split; some wanted a FAQ, some wanted to have the question forwarded to a human, and some wanted both. Additionally, there was an even split between who would select a human tour guide versus a robot tour guide given the option.

Regardless of the type of tour guide, interviewees wanted a primarily friendly and informative tour guide. When asked to evaluate positive experiences with tour guides, most referred to tour guides telling personal stories, being funny, or simply friendly. On the other hand, poor tours consisted of bored guides who were uninformative, rushed, boring, and not funny. The final question asks for feature requests. No trends existed in the responses, but several sought to address the limitations of a robot; one person wanted the robot to have a human remotely answer questions.

I.2 Requirements Specification and Use Cases

This section describes the design of Roslyns internal systems.

I.2.1 Functional Requirements

Table I.1 describes the functional requirements of Roslyn. All functional requirements have been implemented.

Table I.1: The functional requirements of Roslyn

Requirement	**Description**
R01	The possible destinations shall include professor offices.
R02	The possible destinations shall include classrooms.
	Continued on next page

Table I.1 – continued from previous page

Requirement	Description
R03	The possible destinations shall include the Engineering Computing Center (ECC).
R04	The possible destinations shall include the Computer Science & Engineering (CSE) Office.
R05	The possible destinations shall include the male and female restrooms and drinking fountains.
R06	The possible destinations shall include the elevator.
R07	The possible destinations shall include the stairwells and emergency exits.
R08	The possible destinations shall include the research labs.
R09	The robot shall keep an internal map of the environment.
R10	The robot shall select a new navigation route if a stationary object is detected in its global plan.
R11	The robot shall select a local plan which is not obstructed by any obstacles.
R12	The robot shall have a software or hardware emergency stop which will cease all external movement within 1 second of activation.
R13	A software emergency stop shall be activatable from a remote terminal.

Continued on next page

Table I.1 – continued from previous page

Requirement	Description
R14	The robot shall maintain a reasonable distance from the user.
R15	The navigation system shall consider a goal unachieveable if it cannot generate a global plan to it.
R16	The robot shall log its odometry, laser sensor input, provided navigation goals, and position/orientation transformation tree in a ROS bag file.
R17	The robot shall have an instruction to take the user on a predefined tour.
R18	The interface shall show a map of the SEM building with markers for possible destinations.
R19	The interface shall send a goal to the navigation system when the user selects a destination.
R20	The navigation system shall send a confirmation to the interface when a goal is achieved.
R21	The interface shall accept a goal confirmation from the navigation system and then show the user a table of information about the goal.
R22	The robot shall provide auditory, locale-based announcements.

I.2.2 Non-Functional Requirements

Table I.2 describes the non-functional requirements of Roslyn. All non-functional requirements have been satisfied.

Table I.2: The non-functional requirements of Roslyn

Requirement	**Description**
T01	The robot shall help guide users to destinations on the second floor of the Scrugham Engineering Mines (SEM) building
T02	The robot shall accept instructions from an onboard touch screen tablet.
T03	The interface shall be appealing to users.
T04	The robot shall not deliberate for more than one second at any time unless such deliberation is necessary to comply with safe navigation requirements.
T05	The robot shall utilize the Robot Operating System (ROS).
T06	The robot shall utilize the ROS Navigation Metapackage (see Glossary).
T07	The robot shall allow parallel processing on multiple distinct machines.
T08	The robot shall be platform-independent among wheel-driven, ROS-compliant robots.
T09	The robot robot shall not injure a person or, through inaction, allow itself to harm a person.
	Continued on next page

Table I.2 – continued from previous page

Requirement	**Description**
T10	The robot shall halt all movement within one second if any safety-related sensor provides impossible data or no data for more than one second, and alert the operator.
T11	The robot shall halt all movement within one second if any safety-related processing node does not provide expected data within one second of scheduled time.
T12	The interface shall utilize the Android Webview Development kit.

I.2.3 Hardware Requirements

Table I.3 describes the hardware requirements of Roslyn. All hardware requirements have been satisfied.

Table I.3: The hardware requirements of Roslyn

Requirement	**Description**
H01	The robot shall have a maximum linear velocity of 1 meter per second
H02	The robot shall have a maximum linear velocity of 0.5 meters per second
H03	The robot shall have a maximum linear acceleration of 4.5 meters per second per second

Continued on next page

Table I.3 – continued from previous page

Requirement	Description
H04	The robot shall have a minimum linear acceleration of 0.5 meters per second per second.
H05	The interface shall have a wireless communication system to transmit and receive messages with the ROS system.
H06	The computer(s) used for navigation shall have at least the combined equivalent of a quad-core 2.5 GHz Intel i5 processor.
H07	The robot shall have a high-torque shutoff system which prevents excessive force being applied to any obstacle which may come in contact with it.
H08	The robot shall be powered by batteries for no shorter than two hours on a single charge.
H09	The laser rangefinder shall return laser scans with a field of view no less than 180 degrees.
H10	The laser rangefinder shall return laser scans at a frequency no less than 4 Hz (note: see T10 for requirements regarding processing frequency requirements of laser scans).

Continued on next page

Table I.3 – continued from previous page

Requirement	Description
H11	The robots odometry system shall provide odometry information at a frequency no less than 4 Hz (note: see T10 for requirements regarding processing frequency requirements of odometry messages).

I.2.4 Use Cases

Table I.4 describes the use cases of Roslyn. All use cases have been implemented.

Table I.4: The use cases of Roslyn

Use Case	Name	Description
UC01	CreateMap	The operator runs the mapping launch file. The robot sets itself up for mapping and prompts for waypoints.
UC02	EnterTourMode	The operator runs the touring launch file. The robot sets itself up for navigation, localizes itself, then displays the user interface.
UC03	UpdateMap	The operator has run the touring launch file. The robot updates the internal map with every observation.
UC04	GotoPOI	The user selects a POI using the touchscreen. The robot guides the user to it, maintaining social acceptability.

Continued on next page

Table I.4 – continued from previous page

Use Case	Name	Description
UC05	RemainSafe	The user has caused navigation to run and an obstacle is detected. The robot executes evasive maneuvers to avoid colliding with the object.
UC06	GiveFeedback	The user is following the robot to a waypoint. The robot provides information about POIs in the area.
UC07	EmergencyOff	The operator activates the emergency stop. The robot halts all movement within one second and saves the log file.
UC08	EmergencyHold	The operator allows a safety-related node or sensor to stop reporting correct data. The robot halts all movement within one second and saves the log file.

I.2.5 Requirements Traceability Matrix

Table I.5 is a requirements traceability matrix between the functional requirements and each use case.

Table I.5: Requirements Traceability Matrix

	U01	**U02**	**U03**	**U04**	**U05**	**U06**	**U07**	**U08**
	U01	U02	U03	U04	U05	U06	U07	U08
R01	X	X		X				
R02	X	X		X				
R03	X	X		X				
R04	X	X		X				
R05	X	X		X				
R06	X	X		X				
R07	X	X		X				
R08	X	X		X				
R09	X	X		X				
R10				X	X			
R11				X	X			
R12							X	X
R13							X	X
R14		X		X	X	X		
R15	X	X	X	X	X			
R16	X	X					X	X

Continued on next page

Table I.5 – continued from previous page

	U01	**U02**	**U03**	**U04**	**U05**	**U06**	**U07**	**U08**
R17		X		X				
R18		X		X		X		
R19		X		X	X	X		
R20		X		X	X	X		
R21		X		X	X	X		
R22		X						

I.3 System Design

This section provides details about the implementation of Roslyn.

I.3.1 System Classes

This subsection presents a set of figures describing Roslyn's internal class structure.

Figure I.1 shows the system architecture showing relationship between high level nodes in the system. The ROS Nodes are open source nodes (not developed in this project) which provide data to the system and control the robot hardware.

Figure I.2 shows the Social Navigation class diagram showing the relationship between classes and the rest of the systems data flow.

Figure I.3 shows the internal description of the user interface bridge, its intraconnections to itself, and connections to other nodes and to the interface itself.

I.3.2 Top Level Activity Diagram

Figure I.4 shows the top-level activity diagram for Roslyn. It presents two views: the operator view is used by the system maintainer to create a map of the world and set a robot up to navigate in it. The guest view is used by a non-technical guest to interface with the robot via the touchscreen user interface.

I.3.3 Detailed Design

This section presents state charts Roslyn's components.

Figure I.5 shows the message passing statechart. ROS uses an asynchronous and parallel messaging scheme, however it is convinient and appropriate to represent it in this discrete way.

Figure I.6 shows the navigation statechart.

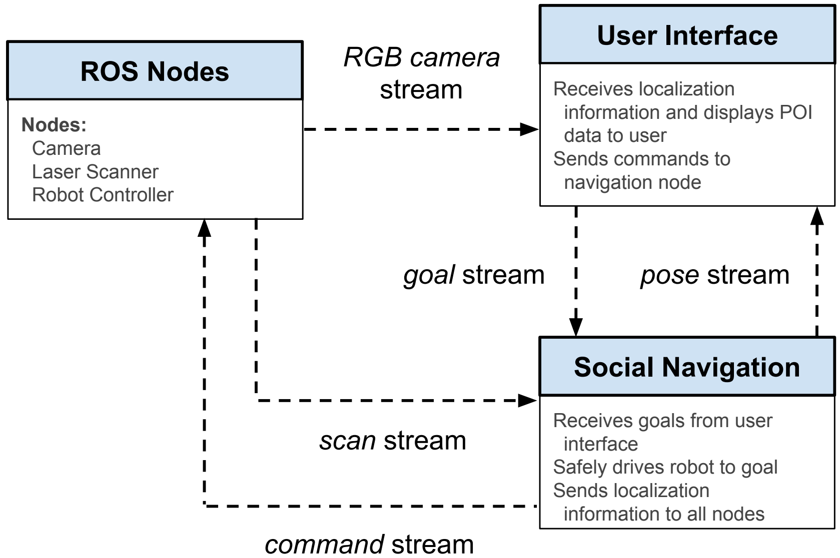

Figure I.1: System architecture

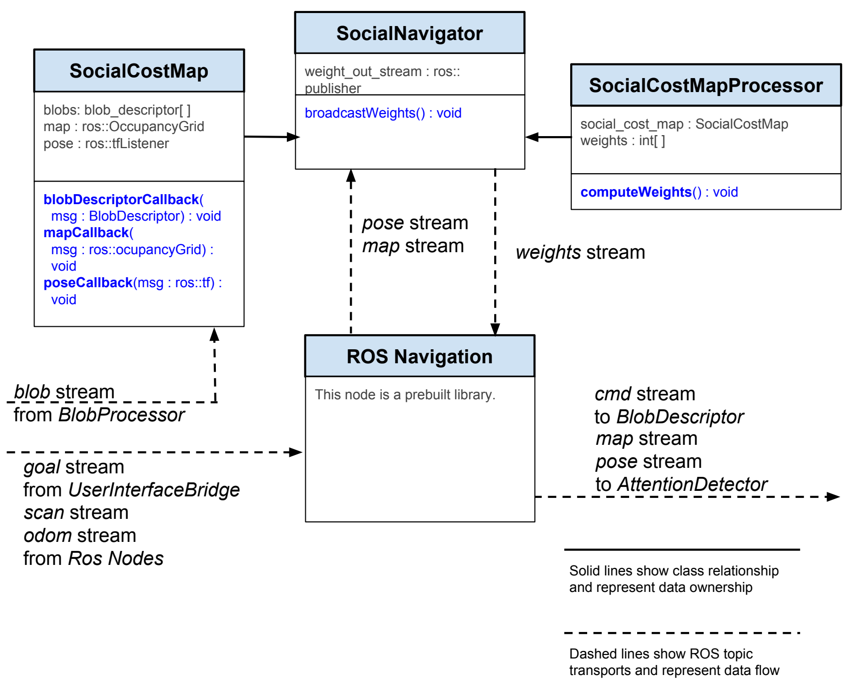

Figure I.2: Social Navigation classes

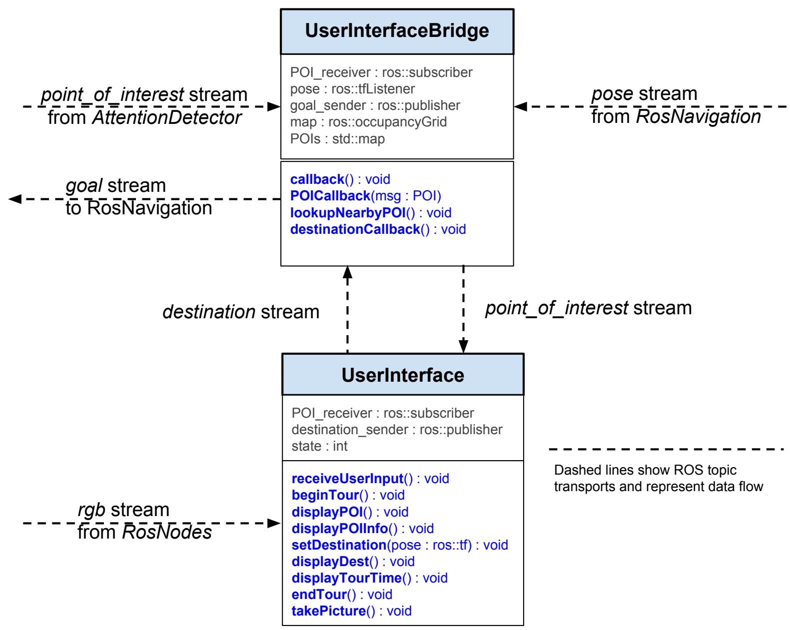

Figure I.3: User Interface classes

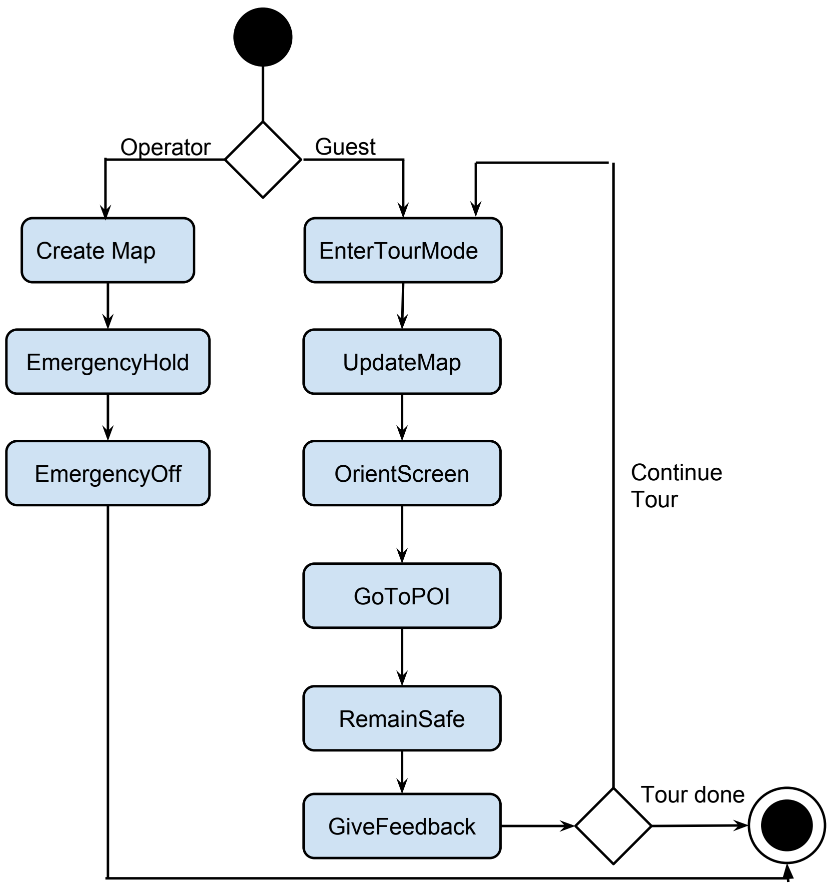

Figure I.4: Top-level activities

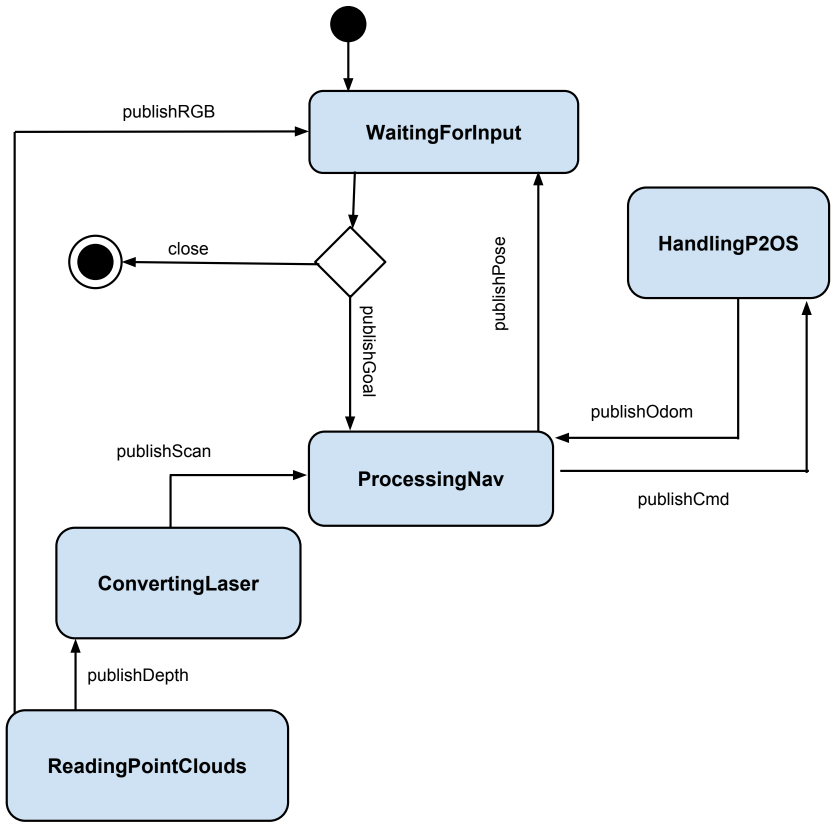

Figure I.5: Statechart for message passing system.

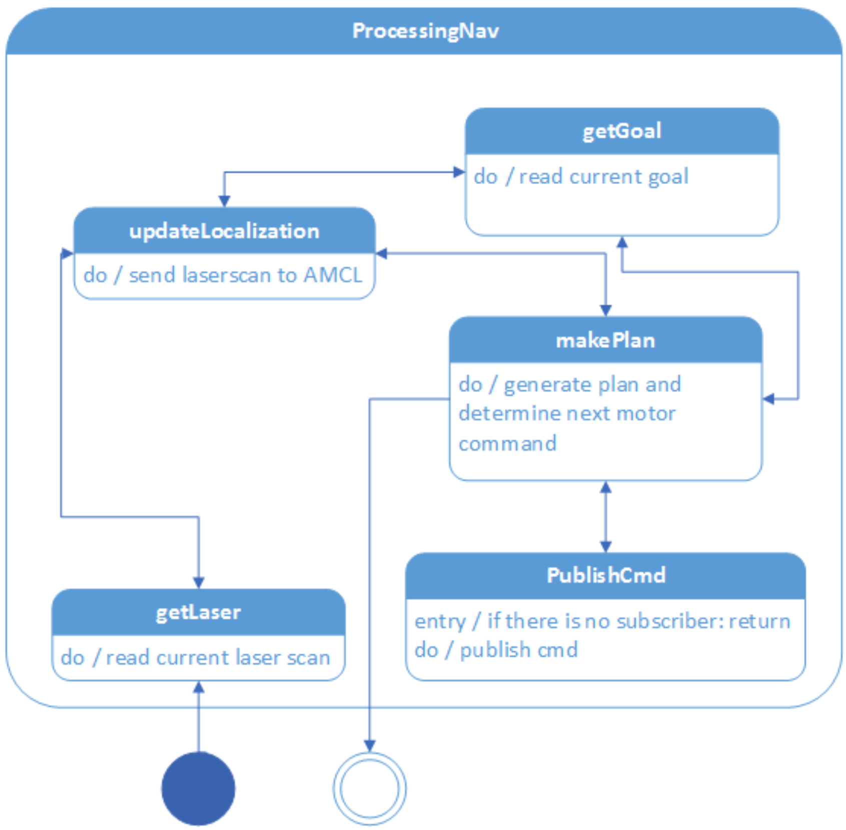

Figure I.6: Statechart for navigation system.

BIBLIOGRAPHY

[1] `USCthesis2000.sty`,
LaTeX 2_ε style file for USC dissertations
and theses according to the regulations published by the Graduate
School, Feb 2000.