**Eteration Case Report**

Enes Cansu 18.04.2022

In this document, there is the report of the case study prepared by the Eteration. In the first part of the report, information about personal characteristics will be given, while in the second part, the method which is used for communication of two nodes in the ROS environment will be mentioned.

The following has been scored on a scale of 1 to 5 as "how much of it I have or how much I can achieve."

* Ability to read/write/think/understand in English (Score : 4)
* Ability to be physically present at the office when required (the consistency matters) (Score : 5)
* Ability to get your way out in a Linux environment. (Score : 4)
* Ability to understand a logical set of instructions (whether it is a tutorial or an article with academic foundations). (Score : 4)
* Ability to ask questions without hesitation or any long/overdue personal over-thinking (Score : 4)
* Ability to work in a transparent manner (audacity to say 'this is not done' or 'i did not do it' instead of hiding and ghosting in a dark room) (Score : 4)
* Having a development environment on top of a Linux distro, using an Ubuntu LTS: 18.04 or 20.04 (#general) (Score : 4)
* Beginner/intermediate skills of scripting in Python (preferable) or C++ (#ros, #general). (Score : 4)
* Ability to use command line interface (a terminal) in a \*nix based environment (#general) (Score : 4)
* Ability to contribute and to collaborate with others via Git (#general) (Score : 3)
* Ability to read and debug a code segment (#general) (Score : 4)
* Basic understanding of web and web technologies (#general) (Score : 2)
* Basic understanding of robots and/or robotics (#ros) (Score : 4)
* Having a basic understanding of communication in a distributed architecture (#ros). (Score : 4)
* Having a basic understanding of how a pub-sub information flow works (#ros). (Score : 4)
* Having a basic understanding of OOP (#general, #ros). (Score : 3)

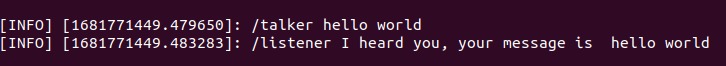
In this section of the report, it will be explained how two nodes communicate with each other in the ROS environment. For this study, wiki.ros.org site was used.

Nodes publish to topics in order to transmit information to other nodes. Likewise, they subscribe to topics in order to receive information from other nodes. Topics are a kind of channels that contain messages.

In this study, the following steps are applied for the communication of two nodes:

* A ROS package named composiv\_tryouts was created.
* The node 'talker' that will publish is created in the composive\_talker.py file inside this package. This node publishes a message of type std\_msgs.msg.String in a topic named 'chatter'.
* Then, a node named 'listener' was created in composive\_listener.py and this node subscribed to the 'chatter' topic. The 'listener' node notifies users that it has subscribed to the 'chatter' topic with the 'callback' function.
* A launch file named composive\_tryouts.launch has been created to run these two files together. When this file is run, the communication between the two nodes is provided through the 'chatter' topic.

Messages coming to the terminal screen when the launch file is run are shown in figure 1.



Figure

With the help of ROS's rqt\_graph tool, it can be seen whether nodes and topics are published in the ROS environment and whether they communicate with each other. Figure 2 shows the rqt\_graph representation of the above study.

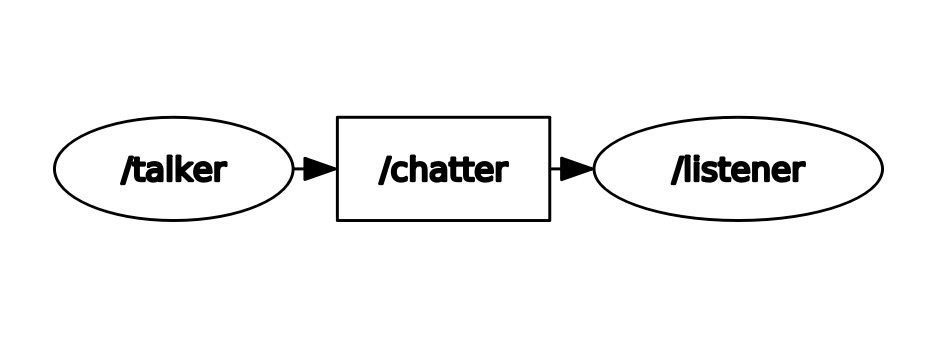


Figure : rqt\_graph representation