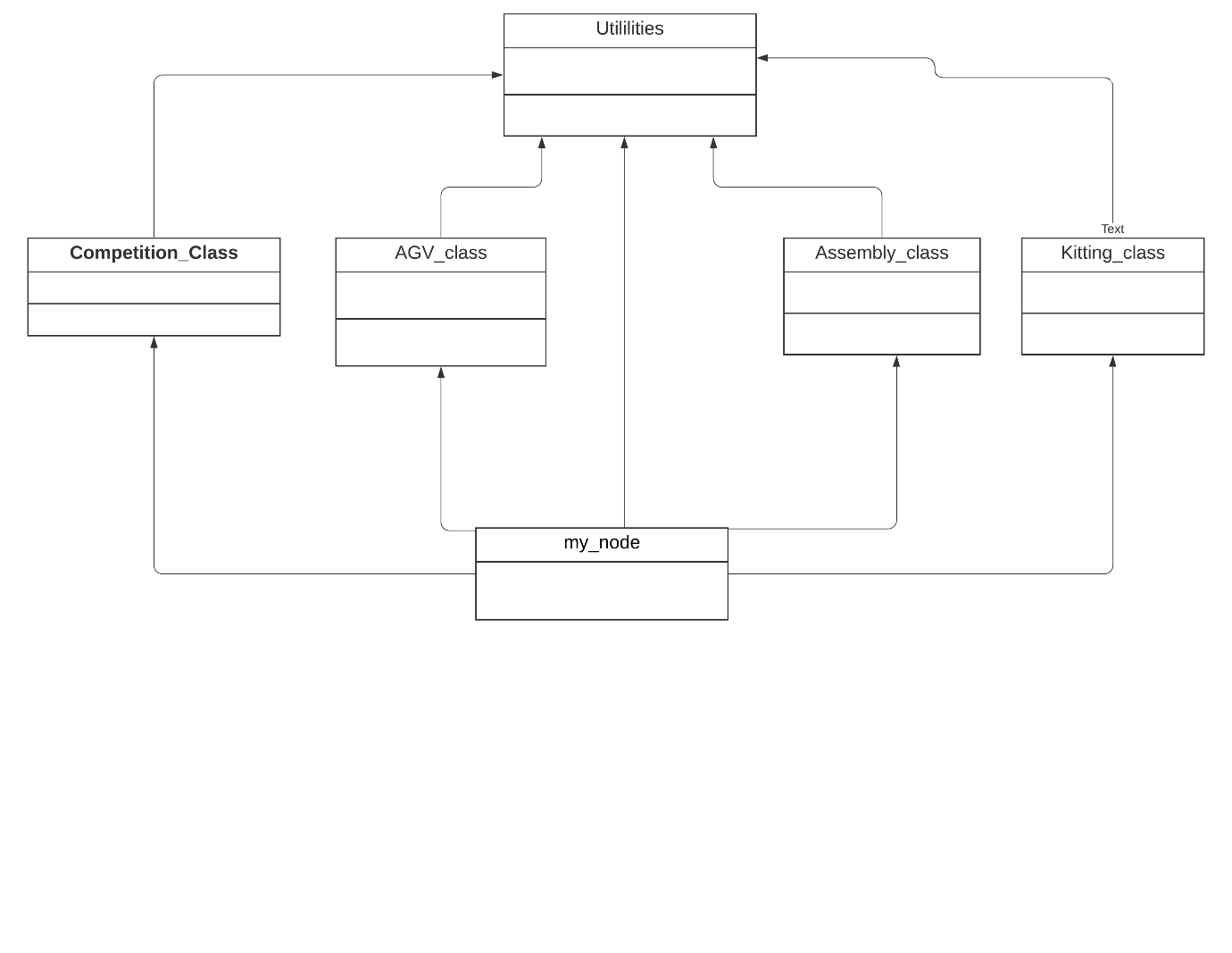
Classes

Figure: Program Structure

The above figure shows the structure of our program that we have tried to implement. It consists of four classes and two functions. The classes are as follows:

1. Competition Class.
2. AGV Class.
3. Assembly Class.
4. Kitting Class.

And the functions are as follows:

1. Utilities function.
2. My Node function.

The Below explanation is of the classes and functions in detail:

1. Competition class: It can be considered as the main class of the entire program. It checks various topic for the data published on the topic and stores the data in appropriate variables for future need. It consists of following sub functions:
2. Order\_callback – The following function listens to the "/ariac/orders" topic on which the orders are published and stores the data in the appropriate variables which can be accessed in the entire programme using various getter functions.
3. Competition\_State\_callback – The function listens to the "/ariac/competition\_state" topic on which the state of the competition is published which has the following three values:

* Init – Before the competition starts.
* Go – Once competition has started.
* Done – When competition has ended.

And sets the value in a variable to perform appropriate action.

1. Sensor callback functions – There are various sensor call back functions in the competition class which listens to the data published on various sensor topic to give the program the details about the of parts in the environment. Some functions also return the type and the location of the part in the environment.
2. Getter Functions – There are various getter functions in the class that help to access the private data member of the class in the main function. They are as follows:

* Get Competition state – It returns the state of the competition.
* Get order list – It returns the vector that contains the order stored in a data variable.
* Get Product list – It returns the list of products that have to delivered to complete the shipment.

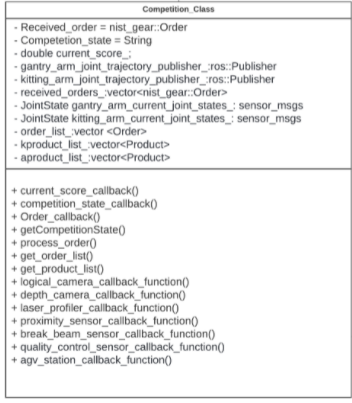
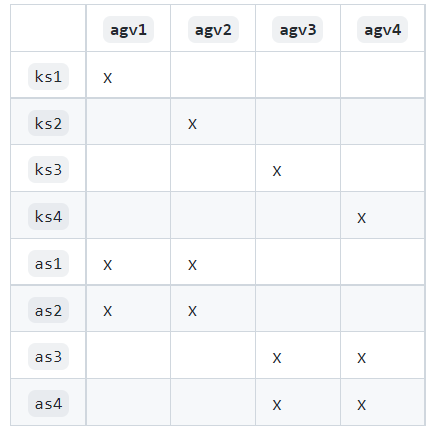
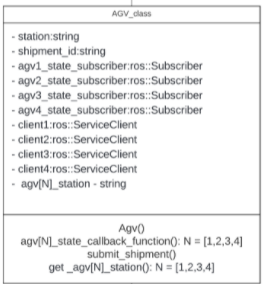
1. Order\_process – The following function is used to segregate the list of products from the order list into kitting and assembly.

Figure: Competition class

1. AGV class - This is the class that controls the part of the program related to the AGVs. It contains 4 service clients one for each AGV It also contains the following function:

* Agv[N] state callback function(N = [1,2,3,4]) – The following functions subscribe to the ariac/agv[N]/state topic where N = [1,2,3,4] which publishes the state of AGV at the time of callback
* Agv[N] state callback function(N = [1,2,3,4]) - The following functions subscribe to the ariac/agv[N]/station topic where N = [1,2,3,4] which publishes the assembly station on which the following AGV is present. At any given time, the AGVs can have the location as given in table below on the topic.
* Submit Shipment – The function is used to submit the AGV to the Assembly station based on the AGV mentioned in the order.

Figure: AGV class Table: AGV Possible location

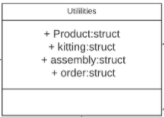
1. Utilities Function – The function consists of various libraries, data variables and functions that have to be included in all the classes and functions.

Figure: Utilities Function

1. My Node Functions: This function can be considered as the main entry point to the program.

It performs the following functions.

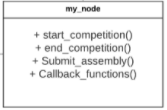
* It initializes the ROS handle, object for the competition and the AGV class.
* It starts the competition.
* It initializes various subscribers and calls the callback functions when something is published on the topic.
* Currently it also submits the order to the assembly station.
* It ends the competition once all the orders are submitted.

Figure: My\_Node