# 1.0 Problem Statement

The application of technologies in households have been drastically evolving in the past decades. This was particularly notable with the prevalence of smart electrical appliances like vacuum robots, personal assistant speakers, IoT electrical switches, etc. Focusing on vacuum robots, various sensors such as bumper sensors, infrared sensors, etc. were installed to help the vacuum robots to navigate across and clean up a room effectively. Also, the robots can also collaborate with each other in a multi-agent environment to further improve the effectiveness of the robots by maximizing the coverage and shorten the time needed to clean the rooms.

With that being said, the implementation of the communicative vacuum robots using the belief-desire-intention (BDI) architecture was not clearly defined previously. Hence, to study the application of the BDI architecture in building the communicative vacuum robots, JAM, which was based on the BDI architecture will be used to build a system that contains two communicative vacuum robots in a grid. Then, dusts will be scattered over the grid which will be required to be cleaned by both the robots collaboratively. After cleaning all the dusts, both robots will be directed to an exit on the grid, denoting that the operations were completed.

In a nutshell, the main objective was to build a system with the BDI architecture to showcase the operation of two vacuum robots collaborating to clean up a room. Apart from the objective, additional features will be added to the robots to improve the effectiveness in the collaboration between the robots which includes preventing to revisit a cell that has been visited previously. Also, the capability to handle the existence of obstacles in the middle of the grid which emulates the real-world scenarios were added to the robots for a more thorough view on the capabilities of the collaborative robots using the BDI architecture in the real-world.

# 2.0 Program Design

## 2.1 Introduction

In this section, the design of the JAM program will be walked through for each and every plan involved. The flowchart will be shown followed by the JAM source code. Then, an explanation on the implementation will be included to justify the heuristics involved in the design of the algorithm especially on the navigation of the robots through the grid cells.

In a nutshell, the program consists of two top-level goals, which were to clean and return. The goal to clean must be attempted first, whereby the robots should attempt to clean so long the dusts are present in the room. Otherwise, the goal to return will be initiated, denoting that all the dusts that can be cleaned were cleaned and the robots should exit from the grid.

## 2.2 Goals and Facts

### 2.2.1 Source Code

/\* BEGIN: Top-level Goals \*/

GOALS:

ACHIEVE clean :UTILITY 10;

ACHIEVE return :UTILITY 1;

/\* END: Top-level Goals \*/

/\* BEGIN: World Model \*/

FACTS:

// Robot A (starting (0, 0))

FACT robotA 0 0;

FACT robotADirection "right";

FACT robotASteps 0;

// Robot B (starting (7, 0))

FACT robotB 7 0;

FACT robotBDirection "left";

FACT robotBSteps 0;

// List of dusts

FACT dust 3 0;

FACT dust 6 0;

FACT dust 2 1;

FACT dust 1 2;

FACT dust 4 2;

FACT dust 0 3;

FACT dust 3 3;

FACT dust 4 4;

FACT dust 1 5;

FACT dust 5 5;

FACT dust 3 6;

// List of obstacles

// The dusts that are surrounded by obstacles cannot be cleaned

// The robot will attempt to search through the grid for an entry

// But if none, the robot will give up after the time limit (MAX\_X \* MAX\_Y \* 10)

// FACT obstacle 4 1;

// Exit location

FACT exit 7 6;

// Grid boundaries

FACT MAX\_X 7;

FACT MAX\_Y 6;

// Whether if the all the grids were visited and the dusts were cleaned

FACT dustCleaned "false";

/\* END: World Model \*/

*Excerpt 2.1: Goals and facts of the JAM program*

### 2.2.2 Explanation

The excerpt above shows the code in declaring the goals and facts in the program. The two top-level goals are to clean and return, which were assigned with the utility of ten and one respectively, denoting that the robots should prioritize the plan to clean over exiting with one plan provided for each goal. After that, the facts form the world model. Firstly, the information of both robots such as the location, direction, and steps taken while cleaning were stored.

Subsequently, the information regarding the location of the dusts on the grid were stored. Should the list of dusts be empty, the robots will attempt to exit from the grid immediately when the program starts. Besides that, the facts on the location of the obstacles in the grid follows the list of dusts. Here, the location of the obstacles were commented out from the code as the obstacles were not provided in the question and is optional in the code. However, the implementation to avoid the obstacles were done and will be discussed in the code.

Lastly, the facts on the point of interests such as the location of the exit, boundaries of the grid, and whether if all the dusts in the grid were cleaned were also included. The boundaries refer to the minimum and maximum points in the grid, whereby the minimum for the x-axis and y-axis are both zero, while the maximums of the grid is specified in the facts.

## 2.3 Plan: Clean