Software engineering for cyber-physical systems

Our project used a TLA+ verification to ensure that no more than one robot can enter the same aisle simultaneously. When designing the verification, the choice was made to keep the NumberOfAisles, the AisleLength and the maximum amount of robots constant. These constants must be defined while testing the TLA+ model in the TLA Toolbox.

System Description:

- A fixed number of aisles (NumberOfAisles)
- A fixed maximum number of robots (MaxRobots)
- Each robot's position in every aisle is initialised at position 0
- A mapping (robotsInAisle) indicates which robot is where

Action Modeled

- EnterAisle(r, a): Robot r enters aisle a at position, only if the aisle is empty
- MoveForward(r, a): Robot r moves further down the aisle a if it is not at the end
- ExitAisle(r, a): Robot r leaves the aisle a upon reaching the end

When starting the program, all aisles are set to be unoccupied, and the robots are not located in any aisles. The Next action allows robots to move into an aisle, exit, or move forward in a given aisle.

The invariant captures the mutual exclusion requirement:

```
AisleMutualExclusion == \A a \in Aisles : (robotsInAisle[a] = {}) \/
(\E r \in robotsInAisle[a] : robotsInAisle[a] = {r})
```

This means that the aisle must be either empty or occupied by exactly one robot.
The model checks this invariant as a theorem to ensure it always holds throughout all possible system executions.

The formal model was successfully verified to satisfy the Aisle Mutual Exclusion property. This confirms that under all valid executions defined by the TLA+ specification, no aisle will ever be occupied by more than one robot at a time. The safety property was proven using model checking in the TLA+ Toolbox.