Testing Warehouse Automation with Autonomous Mobile Robots

Mesut Durukal

durukalmesut@gmail.com

● Abstract

Warehouse automation is an evolving market with bleeding-edge technology. Autonomous mobile robots assist human operators in picking items from the shelves in warehouses to increase efficiency in these systems. In this way, human operators make a shorter distance and don't have to carry heavy boxes.

Testing such a system has various challenges. Firstly, ensuring that the most fundamental functionality is one of the biggest difficulties. The expected result of the algorithm in each picking sequence is the calculated path that has to be followed. This is a minimum distance problem [1]. Testing this and all the other functionalities with physical robots is not easy when the environmental conditions are considered.

Quality is not only functionality. We can consider several aspects such as operability, maintainability, recoverability, performance, usability, and efficiency. These are important for all products, but especially for the Warehouse Automation systems, even more, since the main idea is increasing productivity. Recoverability is also very significant since all the operations may be stuck in case of a fatal error.

Finally, as far as test automation is taken into consideration, there are several challenges coming into place. Along with the hardware testing, every customer has a different configuration. Various options like the number of robots moving around, the number of items in the warehouse, the map of the area, and the priority of the orders should be tested. The reusability of tests is important in terms of duplication and maintenance.

These challenges can be coped with by applying several solutions such as:

- Performing longevity and endurance tests along with benchmarks
- Test Simulation
- Evidence Collection and ROS Bag
- Improving Code Quality in the test automation framework

By improving the testing approach and minimizing drawbacks, a system of Warehouse Automation and Autonomous Mobile Robots development can be verified and validated. By applying the proposed solutions, quality issues are reduced, and customer satisfaction is ensured.

● Biography

Mesut Durukal is a Quality Assurance and test automation enthusiast with experience in several domains. Along with having proficiency in CMMI and experience in Agile practices under his belt, he has taken various roles like Quality Owner, Hiring Manager and Chapter Lead in the organization, leading multiple QA squads in multinational projects.

He has expertise in test automation and integration to CI/CD platforms supporting continuous testing with logging, reporting and root cause analysis packages from scratch. Besides, he has been facilitating test processes and building test lifecycles in the projects.

1 Introduction

Warehouse Automation and Autonomous Mobile Robots (AMR) development is a hot topic. Nowadays, in an online shopping and delivery world, the biggest companies are investing in developing these systems. Companies doing online delivery to either external individual or corporate customers or internal sites, have warehouses to keep their items. These items are categorized on the shelves on the racks in warehouses. Racks are aligned on the isles in a warehouse and items on the shelves have unique barcode numbers. Deliveries are managed by orders. Whenever a delivery is to be done, an order is created with the list of items and products to be packed and sent. Operators collect those items from the relevant locations and leave them at the delivery points. After an order is fulfilled, the next is accepted and processed.

Warehouses are of different sizes and accordingly working operators and shift times are decided by the warehouse operation managers. Optimizing costs and allocated resources are some of the key success metrics. In terms of the operation performance, the completed number of orders and the average order completion pace are other metrics to track. The order of picking is very important since the visiting points will constitute the travel path. Selecting the path, which will result in minimum walking distance, is the main problem to be solved. Figure 1 illustrates how picking order affects the walking distance. Besides, since some of the orders are more urgent than others, order priorities should be managed properly.

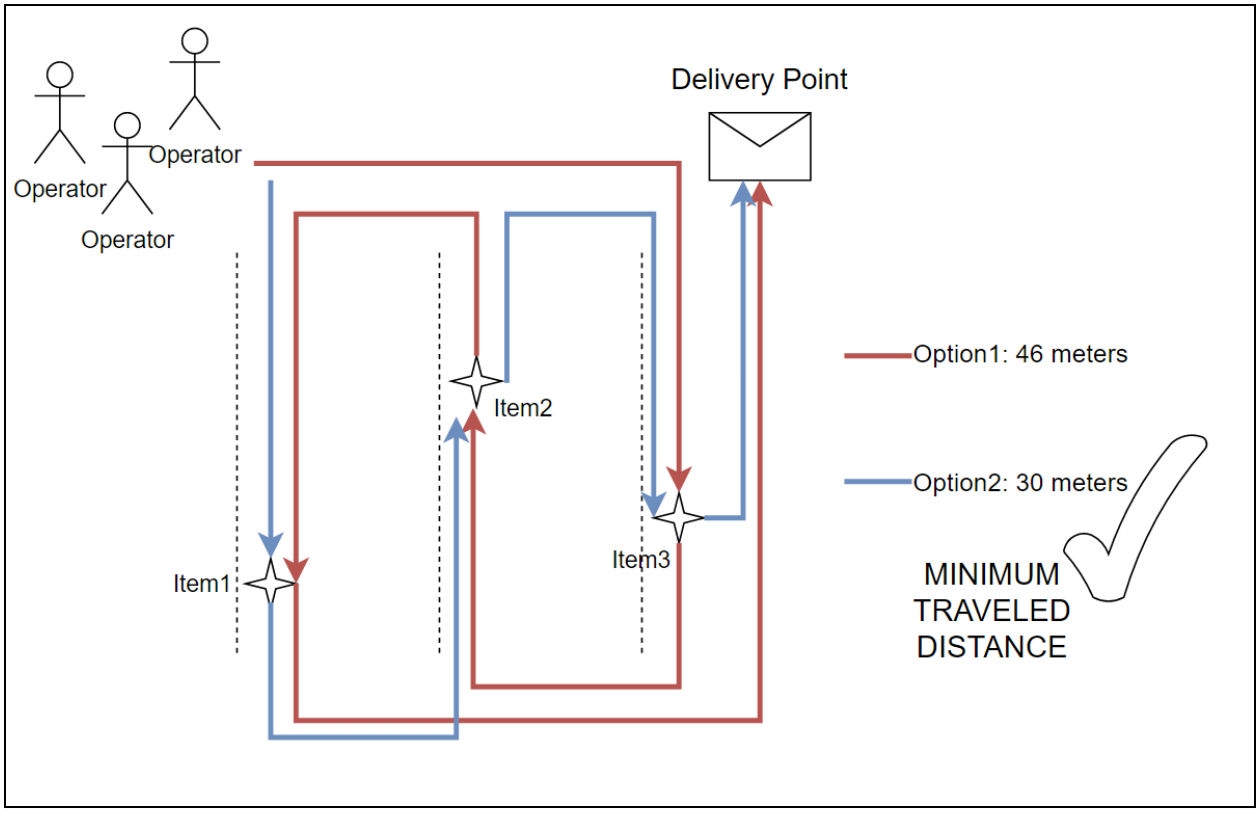

Figure 1: Effect of order of items picking in walking distance.

This is a very basic illustration, shown in Figure 1, where there are only 3 items and 1 operator picking them. But in real-life cases, there are more complex situations with more than 1 operator. Even in the illustrated case, it can be seen that picking items with the order (3,2,1) and the order (1,2,3) results in almost a 50% efficiency difference.

Warehouse automation systems are developed to cope with these challenges. Autonomous robots are deployed in the warehouse to assist the operators. In this way, orders are managed properly with the correct priority. To fulfill each order, the minimum distance is calculated and operators are notified about the next location to proceed. Apart from these benefits, after picking the items, operators have the chance to place them in the trays on the robots to avoid carrying heavy products.

Testing such a system has various challenges. Firstly, ensuring that the proposed travel path by the system is the minimum possible alternative. Additionally, hardware components like barcode scanners, cameras, radar, and all the others introduce more interfaces. To be able to automate the test executions, several simulation environments should be prepared. Finally, developing reusable test cases is needed for managing several warehouses. Each warehouse has a different configuration such as the number of robots moving around, the number of items in the warehouse, the map of the area, and the priority of the orders.

1.1 System Under Test

In this section, the components of the system are described in detail to give a better understanding.

1.1.1 Autonomous Mobile Robots (AMR)

Autonomous Mobile Robots are installed in warehouses, and they move around to assist human operators in picking operations. In the warehouses, there are racks placed in numbered isles and on the racks, there are items to be packed for the deliveries. Whenever there are orders to be processed, human operators go to the relevant locations to pick up the items.

1.1.2 Robot UI

On robots, there is a touchscreen. On the screen, the instructions including the order progress, next picking location, and other operational information are shown. But users are not only notified but also interact with the system using the touchscreen. By these means, they click the buttons on the screen to acknowledge that the current picking is completed, and they are ready to go to the next item. So, basically, it is a web application running in the robotics operating system in full-screen mode. The general overview of a robot with trays, a base, and a screen can be seen in Figure 2.

Figure 2: An AMR [2].

1.1.3 Inventory Management System (IMS)

Items are managed by the Inventory Managed System (IMS). All the details including dimensions (length, height, width, weight), barcode number, name, and location in the warehouse are found in each entity. IMS has an API to respond to the queries made by Central PC and the other components.

1.1.4 Central PC

In this system, a main application is controlling the whole operation. The PC where this software is installed is called the central PC. Whenever an order is created, the following steps are performed by central PC to make decisions:

- Items to be picked within the order are listed
- Location of each item is defined
- Dimensions of items are defined
- Based on dimensions and the tray sizes on the robots, needed tray number is calculated
- According to the calculated tray number, the number of needed robots is defined
- Current number of available and busy robots are checked
- Depending on all the collected information, robot allocation is calculated
- All individual tasks are sent to the relevant robot.

There is a web app, which runs on the central PC and has a UI, that shows order statuses, robot tasks, and some other graphs to the operation managers.

1.2 Testing Strategy

Several customers using this system have different warehouses. The challenge in this regard is, verifying the functionalities of each warehouse configuration. Their size, racks, map, items, and operators are customized. The following test activities are performed when the product is developed for any of those warehouses:

- Unit tests
- Map of the warehouse is generated and simulation is executed
- System tests are executed with simulation
- Tests with real robots in the test area
- Tests with real robots in the warehouse
- Acceptance tests

For managing different levels of testing, separate frameworks are used. Where gtest and C++ based code is developed for unit tests, Python and E2E frameworks like Selenium are used for system-level tests.

Some warehouses are far away from the development offices. Going to the customer environment before seeing the developed software with real robots in in-house environments would be risky. By performing tests in the designated areas inside the company first, trivial bugs are found in first place before going to real execution environments.

Figure 3 shows the test automation framework. System-level tests are implemented with Selenium [3] and Requests [4] library. The API and UI interactions are automated in this framework.

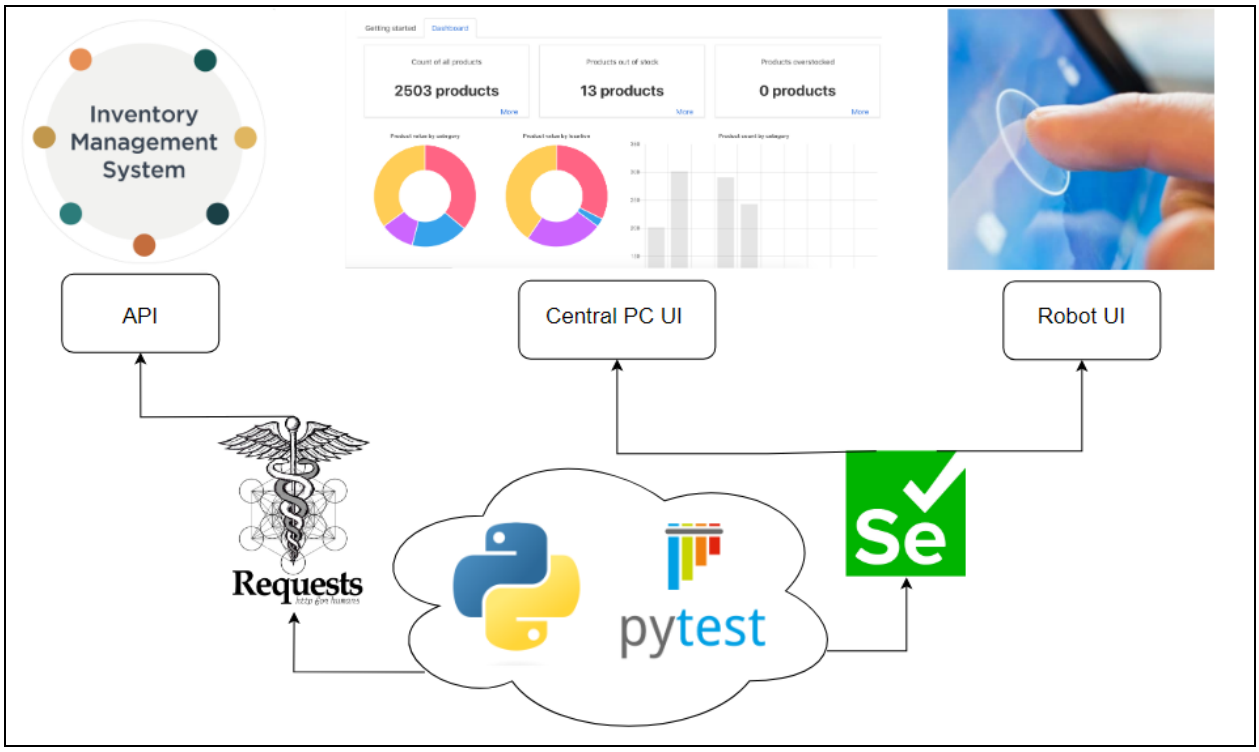

Figure 3: Test automation environment.

2 Testing Challenges and Solutions

2.1 Ensuring Efficiency

The main purpose of the system is to reduce the traveling distance. But this is a complex and complicated problem that can be solved with mathematical optimization. With functional testing, it is not easy to verify that operators are traveling the minimum distance.

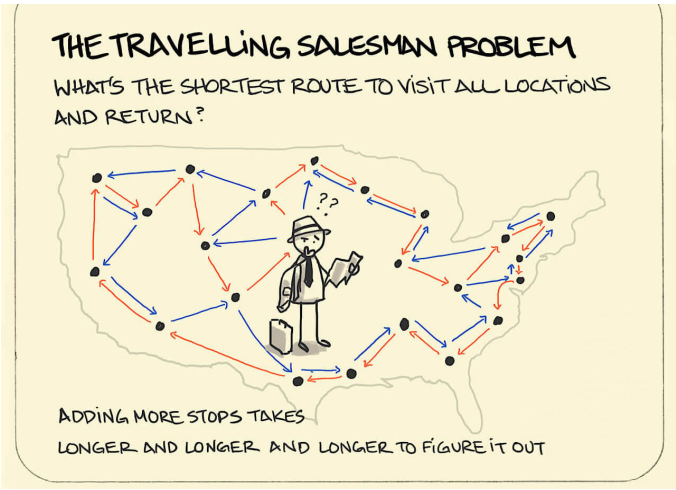

Figure 4: The Traveling Salesman Problem [5]

To prove the benefit of the system, what we do is the benchmarks. We calculate key metrics after a longevity test on both an operation with only human operators without robots and an operation with robots. The performance metrics that we collect are:

- Order Completion Time
- Walking Distance
- Human Operator Needed

At the end of the operations, demonstrating that they are completed in a smaller time with robots is already convincing for the customers. Running similar comparisons with lots of versions paves the way for finding the best-performing algorithm.

2.2 Hardware Components

Since there are hardware modules in the system, test automation is not straightforward. The sensors, camera, or other hardware modules like the brake and emergency button should be integrated into the testing environment in order to simulate the scenarios. Manually triggered operations like brake, or camera detecting an obstacle in front of the robot can be simulated by generated signals. They can be mocked in the simulation, which is developed on top of the Robot Operating System (ROS) [6].

In real life, the robot is developed as a network composed of nodes communicating with each other through ROS. Sensors and processors are the nodes on the robot, forming a network. All nodes subscribe to the messaging channel to receive generated messages by other nodes. Publishing a message in the relevant messaging channels by mocking the hardware modules enables automating the relevant test cases.

2.3 Non-functional quality aspects

Not only functionality but also several other aspects of quality contribute to customer satisfaction. One of the most important non-functional aspects is usability. Especially in this system, the ease of use is crucial to assist the operation. If the operators struggle to use the system, they lose time to complete the interactions, and the waste of time will increase.

For the sake of efficiency, usability should be ensured. One thing which can be done to collect feedback about the usability of the system and figure out its weaknesses is customer surveys. After revealing what the customers are struggling with most or what they request, they can be implemented in the next versions.

Besides, in warehouse automation systems, recoverability is very significant. In warehouses, some orders might not be completely fulfilled, which means they might just fail. For instance, if an item to be picked inside the order is missing, the order can not be completed. Due to a similar reason, if an order fails, the next orders should be able to proceed. Otherwise, the whole chain would be blocked, and all orders would be pending. The system should be designed flexibly enough to manage these kinds of failures. In an unexpected failure, the order can be marked as failed and the system proceeds with the next pending order.

The recoverability of the system can be tested by chaos testing activities. Some parts of the systems can be made down intentionally to observe the other parts. Similarly, intentional order failures are triggered to observe the system's capabilities to handle the failures.

2.4 Reproduction of Issues

In test automation, reproduction of the issues is an important challenge. Tests are executed automatically in the pipelines and the results are analyzed after the execution is done. In case of failed tests, the root cause should be understood to take action items. If they are false alarms, tests should be fixed. On the other hand, if they are real bugs, they should be fixed from the product side.

But to be able to perform the root cause analysis, the issues should be reproducible. Sometimes it is not easy, because the conditions under which the issue happened may be arbitrary. Considering the moving robots and the environmental conditions like humans around or temperature, humidity, and other factors which may affect sensors, it is highly possible to struggle to reproduce the failure.

What can be done to understand what happened during the issues is, recording all the ROS messages and playing with the ROS bag. Figure 5 shows a depiction of the ROS bag. You can replay the recorded messages and in this way better understanding of the incident can be had.

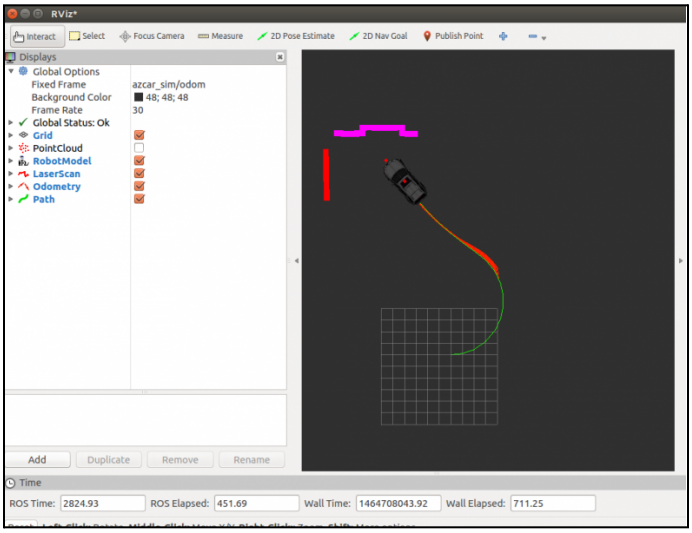

Figure 5: Rosbag [7]

2.5 Maintainability of Tests

Some warehouses are smaller, while others have very wide areas. In other words, in some warehouses, a few robots are enough for the operation but for some others, up to 30 robots are installed. Tests should be flexible and scalable enough to cover various configurations. Otherwise, each time the same scenarios should be implemented over and over again which results in duplication.

Figure 6 shows an example warehouse. To reduce duplication and improve scalability, tests are implemented in a flexible way, and environmental variables are separated from the test code. They are managed as a configuration project, providing the necessary flexibility to the tests.

Figure 6: Warehouse view [8]

Maintainability of the test automation framework is not only related to the warehouse automation systems but any test automation project. Test code should be kept up-to-date covering the behaviors, flakiness should be removed and reliability should be continuously maintained.

Code quality can be maintained via either static code analysis tools or peer review activities. Potential considerations to improve code quality:

- Test smells such as flakiness, fragile tests
- Reliability
- Robustness and stability
- Understandability
- Execution duration
- Resource consumption
- Efficiency
- Compatibility (cross-browser testing)

3 Conclusion

Warehouse automation testing is not easy and there are various challenges. However, there are ways to cope with these challenges on the other hand. All aspects of quality can not be verified by functional testing. They should be supported by various non-functional testing activities like performance, longevity, resilience and chaos testing.

The structure of the system with moving robots and other environmental factors like obstacles in the robots' paths, and human operators, cause some difficulties in testing activities. Perception modules and sensors are connecting robots with the real world.

A summary of challenges and proposed solutions is shown in Table 1.

Challenge	**Proposal**
Ensuring efficiency	Longevity testing
Hardware components	Simulation
Usability, Recoverability	Customer communication, Chaos testing
Reproduction of issues	Evidence collection
Maintainability of Tests	Reusable tests, Code quality

Table 1: Challenges and Solutions in WA Testing

Since the operators are the direct users of the system, their opinion and their satisfaction with the usability of the system is very important. End User surveys and warehouse visits can give a chance to collect feedback including the potential improvements to the product.

Finally, as for all the other test automation projects, code quality is one of the most impactful dimensions of the automation framework. The quality of the test code affects the reliability, robustness and accuracy of the tests. Furthermore, maintenance and analysis efforts can be reduced by improving the quality.

In short, there are several challenges to performing verification and validation of a warehouse automation system. But with the correct adaptations, it is possible to achieve quality goals.

References

[1] Chih-Ming Hsu, Kai-Ying Chen, Mu-Chen Chen, Batching orders in warehouses by minimizing travel distance with genetic algorithms, Computers in Industry, Volume 56, Issue 2, 2005, Pages 169-178, ISSN 0166-3615, https://doi.org/10.1016/j.compind.2004.06.001.

[2] https://www.rapyuta-robotics.com/ja/ (accessed June 10, 2023)

[3] https://www.selenium.dev (accessed June 10, 2023)

[4] 'Requests: Http for Humans', https://requests.readthedocs.io/ (accessed June 10, 2023)

[5] 'Traveling Salesman Problem (TSP) with Miller-Tucker-Zemlin (MTZ) in CPLEX/OPL', https://co-enzyme.fr (accessed June 10, 2023)

[6] https://www.ros.org/ (accessed June 10, 2023)

[7] https://cps-vo.org/node/26614 (accessed June 10, 2023)

[8] https://www.rapyuta-robotics.com/ja/ (accessed June 10, 2023)