FACTA UNIVERSITATIS
Series: **Automatic Control and Robotics** Vol. 21, Nº 3, 2022, pp. 147 - 157
https://doi.org/10.22190/FUACR221120012S

Regular Paper

DESIGNING DISTRIBUTED CONTROLLING TESTBED SYSTEM FOR SUPPLY CHAIN AND LOGISTICS IN AUTOMOTIVE INDUSTRY

UDC (681.58:004.89)

Martin Štufi[1], Aleksandra Panajotović[2], Leonid Stoimenov[3]

[1]Solutia s.r.o., Prague, Czech Republic
[2]University of Niš, Faculty of Electronic Engineering,
Department of Telecommunications, Republic of Serbia
[3]University of Niš, Faculty of Electronic Engineering, Department of Computer Science,
Republic of Serbia

Abstract. *The arrival of the era of autonomous robots is indisputable. In this paper, innovations in the distributed control systems realized by autonomous guided vehicles in the automotive industry are provided as proof of concept. The main goal of the considered distributed control system design is to bring all-in-one dependent and independent VDA 5050 compliant robots that are easily configurable and manageable with the web-based high-quality user interface responsive business-critical application. Special attention is paid to applying a platform to manage all autonomous IoT based robots in one seamless system. In addition, a "single point of truth" as one of the main issues of modern distributed controlled systems has been considered.*

Key words: *Distributed control systems, Autonomous robots, Autonomous guided vehicles, Internet of things*

1. Introduction

The high demand, rapidly increasing customization, and differentiation of product portfolios in the shorter life cycle of car production in the automotive industry led to the development of flexible manufacturing [1]. Therefore, the automotive corporation strategy is to become very important in logistics within the automotive industry by using a modern distributed control system (DCS). As part of that strategy, many leading worldwide companies

---

Received November 20, 2022 / Accepted December 14, 2022
Corresponding author: Martin Štufi
Solutia s.r.o., Vrsovicka 1461/64, 101 00 Prague 10, Czech Republic
E-mail: martin.stufi@solutia.cz

recognize that changing global material handling markets relating to productivity requirements, labor cost, and e-commerce create a permanent demand for new logistics solutions [2]. In supplying automotive assembly lines, production automation can be implemented in Autonomous Mobile Robots (AMRs) or Autonomous Guided Vehicles (AGVs) [3].

The rapid growth in IoT and sensor networks, in combination with the challenges in automotive logistics such as operational efficiency, improving customer service level, data-driven demands, and digital transformation of automotive logistics, set to pave the way for new revenue streams. In the industry's evolution, Toyota Industries Corporation has become a leading logistics solutions provider, creating a permanent demand for new solutions. However, selecting appropriate technologies and developing an IoT distributed control system can be challenging for the automotive industry. For example, the need to set up an efficient, effective, and relevant product or service solution for managing AGVs in automotive logistics requires modern technology implementation. AGV is a vehicle with a battery and local computer control that can automatically move in a specific or predefined direction without human intervention. The most specific feature of AGV is that it can move the unit load from one place as a pickup point to another delivery point. In modern automotive logistics systems, DCS frequently manages these operations.

According to a path guidance principle [4], AGVs can be divided into static and dynamic path vehicles [5]. The static path vehicle uses a predefined path from the pickup point to the delivery point. The dynamic path vehicle uses its algorithm and autonomously determines the optimized path by detecting and avoiding obstacles on the floor plan. In the paper [6], the authors examine the limitations and prerequisites for implementing AGV technology for automating specific logistic processes. The case study [7] describes the increase in the productivity of the last workstation of an assembly line by implementing an AGV system to transport finished goods to the warehouse. A comprehensive knowledge base for designing, selecting and implementing the AMR technology, considered as the highest level of AGV, in the form of unit load carriers for supplying assembly lines in the automotive industry is presented in [8]. Another case study [9], answers the following questions: what are the requirements to be considered when using autonomous transport to move materials, and what steps must be taken to introduce autonomous technologies in the industrial environment. Using deep neural networks (DNN) in online monitoring for AGV is presented in [10]. The review of how automation and robots contribute to the increased profit demand by offering lower production costs, shorter delivery times, improved quality and higher customer satisfaction, is presented in [11]. The paper [12] focuses on developing Digital Twins for flexible manufacturing systems with AGVs on the factory floor.

All the above studies propose principal limitations for AGVs/AMRs, their controlling problem, supplying automotive assembly lines, and their relationship with DCS. The contribution of this article can be recognized in the improvements embraced by the Resmanio platform (www.resmanio.com), such as the integration of heterogeneous AGVs that are VDA 5050 compliant and incompliant, supporting different protocols in AGVs' communication and Continuous Integration and Continuous Delivery (CI/CD) readiness.

This paper is arranged as follows: Section 2 presents a global architecture with a high-level DCS system design focusing on modern distributed control. The use of the DCS application is presented in Section 3. Further, Section 4 contains the discussion. Finally, the authors discuss the future trends of DCS development and give concluding remarks in the last section.

2. Material and Methods

Škoda Auto continues its digital transformation journey. AGVs are operated through separate control systems of individual suppliers in their production plants. These systems are not interconnected, i.e., they provide services independently of others and represent separate fleets. After implementing the Unified Control System (UCS), a robust and scalable system will ensure the interconnection of all vehicles and fleets.

The system is built from the start to handle a large number of messages going from AGVs to the system's control section. Furthermore, the system is designed to operate in real-time. From the standpoint of architecture, it may be deployed and operated both in the Cloud environment (public or private) and in the customer's so-called on-premises environment. In this scenario, we propose system implementation in the Škoda Auto environment throughout the locations of the manufacturing plants. The entire solution is designed as an enterprise solution, where a business-critical application built in Java enables Škoda Auto differentiated products and services to their users. It requires only a light client, any web browser such as Google Chrome, Microsoft Edge, Firefox, and Opera. The data created and reported by the AGV software model are included in the data set utilized for this experimental study. The data is produced and sent via a multi-threaded Java application.

The project starts with an identification of challenges for digitalization in the logistics [13], such as (1) excess supplies with materials, spare parts, repairs, and maintenance; (2) overproduction of manufactured goods; (3) waiting for material, spare parts, information, transport; (4) excess manipulation as overlapping, translations, shifts, incorrect organization of warehouses, and material flows in production; (5) search material, documentation, information; (6) unnecessary activities and movements in human work or manipulation in movements, and (7) duplicate information, a selection from IT systems, transcription of reports.

Aligned with the customer objectives, this study focuses on four action points: (1) the testbed platform responsible for managing all AGV devices with in-built IoT sensors in one seamless system; (2) a single point of truth for all AGVs which allows the possibility of connecting these AGVs to the distributed control system; (3) comprehensive real-time notification functionality through the various communication channels as notification messages, e-mails, mobile applications or web applications, and (4) comprehensive support and management of all AGVs through a single VDA 5050 (Verband der Automobilindustrie) compliant platform.

The complexity of enterprise application growth with increases in functionality and features supported. Fig. 1 depicts the overall Microservice architecture and high-level design for the testbed horizontal scalable platform as DSC. The java-based application runs on the application server, and the main concept is built as microservice architecture. A Web Components part provides business logic from the Backend server. It provides features in a single Java application that is served up, operates on the application server, and offers supplementary services to the user.

Nowadays, creating an application with a microservice architecture typically calls for leveraging Java, a lightweight development environment, and widely used global frameworks like Spring Boot or Spring Cloud. This implementation provides enhanced features, such as new classes that provide Stream API or Lambda Expression [14] to support stream data from the sensor network. Moreover, through the API, it can be integrated into our application within external systems.

The DevOps movement has become synonymous with CI/CD, and the Resmanio platform has been designed and developed using the same principle. Continuous integration design practices result in this modular software that can be repeatedly integrated with other components into a single source. The continuous delivery ensures that this software is production ready. This testbed architecture offers a serverless Kubernetes cluster manager for Docker-based microservices with the proposed solution realized in a domain-driven design (DDD) fashion. As a result, Docker has become the standard used to develop and run containerized applications.

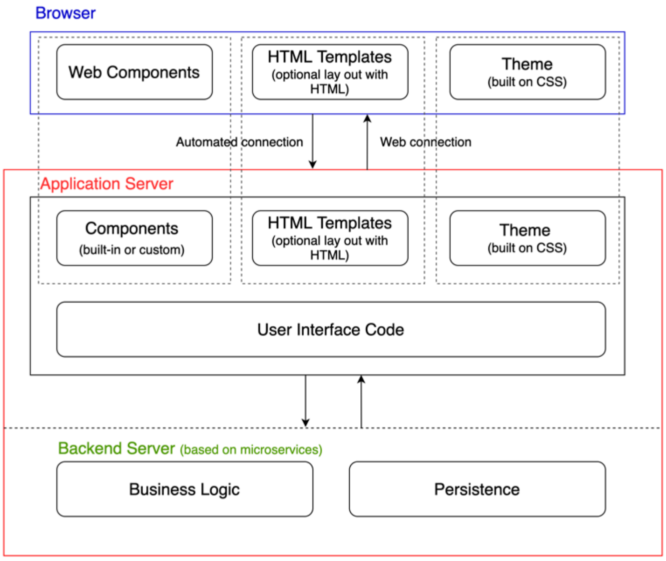

Fig. 1 High-level architecture diagram

3. Implementation of Testbed

Key characteristics of the testbed platform can be identified in Resmanio (Fig. 2), such as the Enterprise Java-based application used to build the mentioned system, which provides

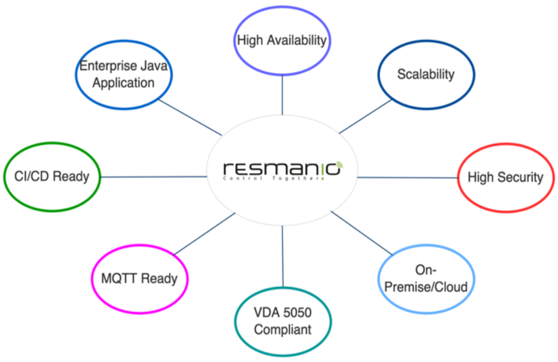

Fig. 2 Resmanio characteristics

high availability, scalability, and security. All components of our solution architecture are scalable to suit many connected devices. Resmanio can also provide on-premises or cloud-based built-in services. It has been designed as VDA 5050 compliant and Message Queuing Telemetry Transport (MQTT) ready. All software components are ready to be deployed in a cloud, such as Google, Amazon, Oracle, and Microsoft Azure.

Implemented testbed platform, shown in Fig. 3, can be recognized in the many production plants in which AGVs can operate. AGV is equipped with IoT sensors and connected to a secured dedicated network. The Data Streaming Sources part provides real-time data to the Real-Time Data Processing Platform and the Application Layer. Using Apache Kafka and Apache Spark, Event Streaming, Transformation, and Processing acquire the stream data and ensures their transformation and processing in near real-time.

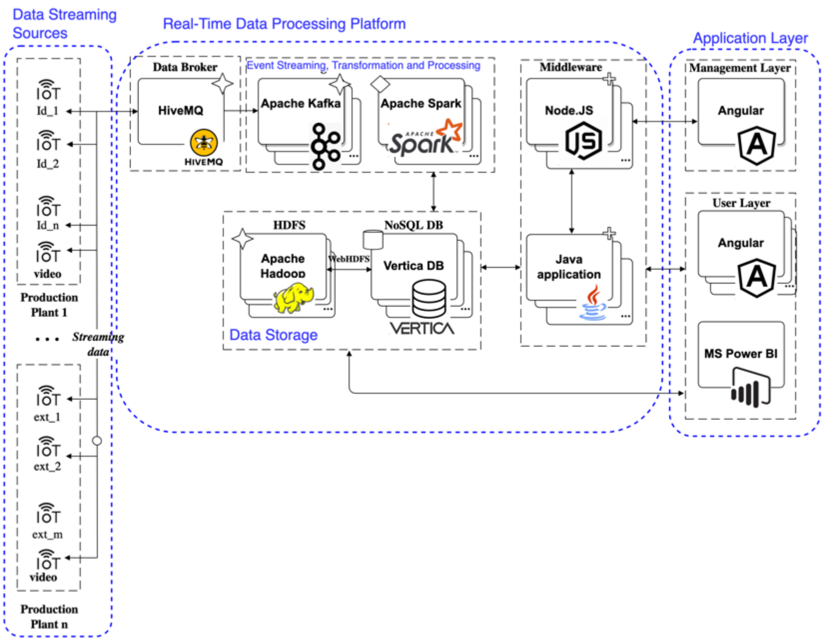

Fig. 3 System architecture design

We identify a layer where individual AGVs are equipped with IoT and location sensors. Information from the sensor is received through a specialized component called a broker, which enables the collection of these events/information in real-time. A Data Broker handles this data via HiveMQ MQTT broker, which connects AGVs and the rest of the implemented platform. It ensures that the information moves between vehicles and testbed platforms implemented on-premises or in a cloud environment. Our solution would allow up to 100,000 events per second within the existing architecture. This component is horizontally scalable. Subsequently, the information from the AGV through the broker passes through a specialized part of the Resmanio system intended for data transformation. Such data is kept in bulk distributed data storage in Apache Hadoop. Vertica is a NoSQL database used for analytics.

Key functionalities and features of this platform are as follows: (1) create standardized real-time communication between different AGVs/AMRs; (2) the unlimited number of

AGV/AMR depending on the size of the floor plan; (3) enable the integration of any AGV/AMR vehicles complying/not complying with the VDA 5050 standard; (4) enable decisions about route selection or intersection behavior using the Resmanio Collision Prevention System; (5) VDA 5050 enabled as a standard interface for all kinds of autonomous robots, not only in the automotive industry; (6) in-built collision prevention system (CPS) enables safe operation in the defined floor plan of the production hall; (7) enables immediate order redirection, order priority and orders reorganization; (8) no predefined path, but the suggested way is the result of the AI/ML system; (9) very high flexibility of route planning and AGV/AMR management; (10) "Plug & Play" functions enable close cooperation with any VDA-compliant vehicle 5050.

Fig. 4 shows communication between UCS Resmanio and AGVs/AMRs to provide a sustainable solution. These AGVs/AMRs can be VDA 5050 compliant or not. This should enable full fledge coverage with legacy AGVs/AMRs, for a guided vehicle. This adapter represents the uniform interface between the USC (Resmanio) and legacy AGVs/AMRs. On this basis, standard communication with AGVs/AMRs integrates transport systems into a continuous process automation. The following features can be provided through the VDA 5050 adapter, which is necessary for managing not compliant AGVs/AMRs: increasing manufacturer independence by using the VDA 5050 adapter, complexity reduction, and increasing vehicle autonomy by consuming existing hardware and software equipment.

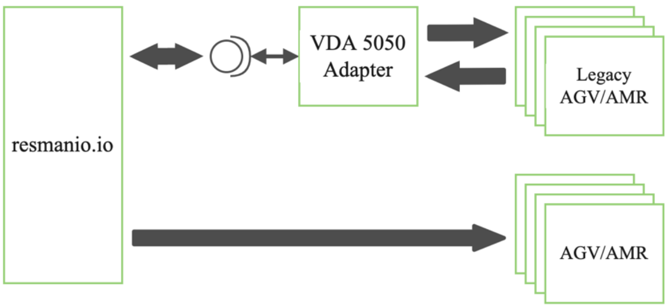

Fig. 4 Resmanio VDA 5050 Adapter

As shown in the sequence diagram for managing all AGVs/AMRs (Fig. 5), there are at least the following actors: (1) USC Resmanio, using resman.io Java Enterprise Application, provides the initial configuration to AGVs/AMRs; (2) VDA 5050 Adapter ensures a seamless communication with non-compliant AGVs/AMRs; (3) MQTT Broker transfer VDA 5050 protocol-based MQTT messages to all AGVSs/AMRs. In the case of VDA 5050 compliant AGVs, instructions or messages generated by resman.io will be broadcasted via MQTT broker to VDA 5050 compliant AGVs/AMRs. Otherwise, these instructions will pass through the VDA 5050 Adapter.

Once the setup of AGVs/AMRs is done, their management is transparent for USC Resmanio in both directions. Some of the key functionalities which Resmanio provides can be recognized as Order planning, Scheduling, Maintenance, Order tracking, Disorders, States of individual devices, AGV report, AGV changes and settings, Delays in orders/processes, Definition of resources or Order Execution which combines action, status, and notification kinds of messages.

As a USC platform, Resmanio provides a feature for integrating with other information systems, such as Enterprise Resource Planning (ERP), Production Service Bus, Messaging-type integration, integration with various identities, Web Sockets (Secured) and Web Services (REST API), application integration, BI and Reporting Server integration, and integration with CAD systems for floor planning design.

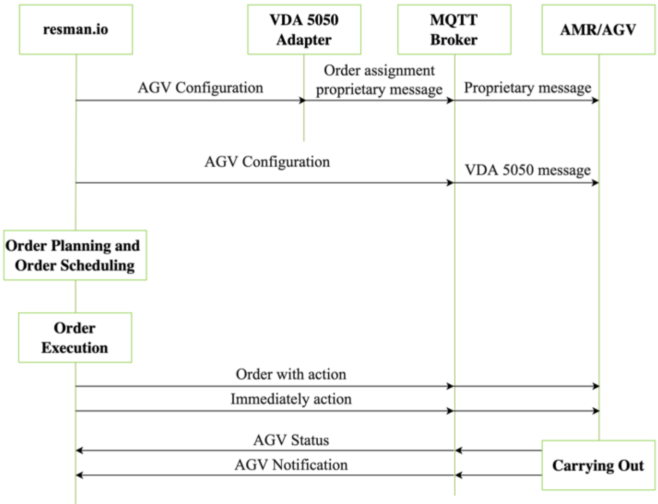

Fig. 5 Time sequence diagram within resman.io and AGV communication using MQTT

The platform's application part (Fig. 6) has the modules realized in a microservice architecture fashion [15]. Distinguished modules are Production Plants, Control Management Centers, Device Settings, Integration, Reporting and Administration. Also, Resmanio provides a central overview within the platform, an overview of all events, order tracking, statuses of individual AGVs/ARMs, and delays or resource definition. We can ensure, for instance, effective AGV planning and control, set special responsibilities within the shifts by fleet, and gather vehicle information or information on progress orders within a production plant. Furthermore, Resmanio offers Floor Plan Designer functionalities inside the modules as Control Management Centers, AGV Management, Order Management, and Fleet Management.

The production line floor layout and various obstacles, such as Barriers, Charge points, Pick points, Delivery points, Emergency and Exits, may be seen in the AGV Management module, shown in Fig. 6. Resmanio provides information about AGVs/AMRs in a real-time fashion. Car production is planned according to shifts based on ordered cars. Using Resmanio as USC makes it possible to plan a shift several weeks in advance. An automated system gives Pick points upon receipt of commodities or specific automobile parts. As a result, AVG is fully automated and capable of supporting the assembly of cars.

The AGV Management module also offers a thorough overview of each AGV/AMR, including identification, battery status, the total distance since the last charge, current status, the dates of the last and subsequent maintenance, and order history with the location of the product factory.

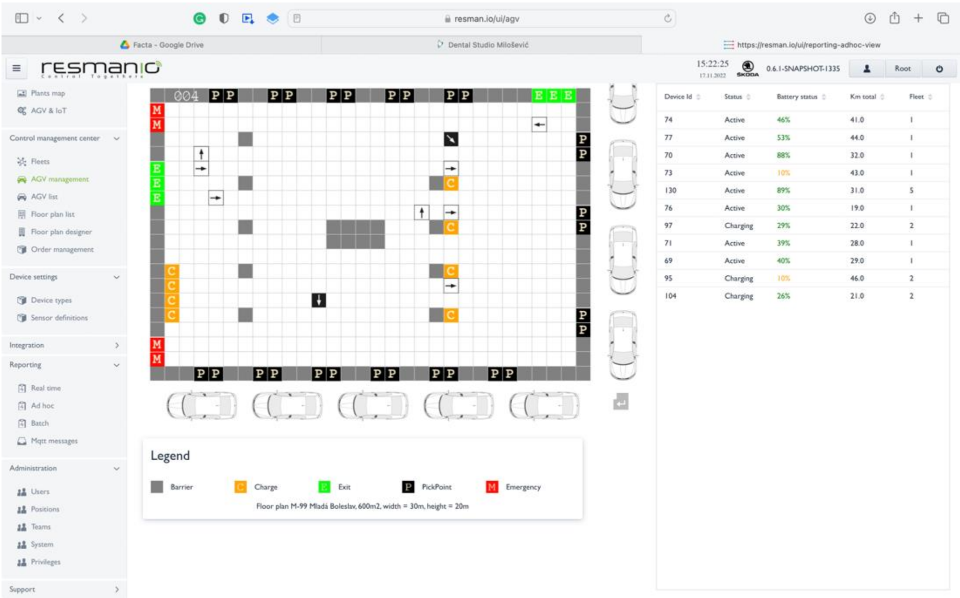

Fig. 6 AGV Management module

Real-time data on orders, users, operating tasks and processes, sensors on AGVs, and monitoring are provided via the Reporting module. There are two report types: batch and online (Fig. 7 and Fig. 8). On a daily, weekly, or monthly basis, these reports offer varied information for all orders or specific orders. Information on VDA messages sent from AGV/AMR to USC or DCS may be found in the online report section (Fig. 8). This report also includes a graph that shows patterns in handling certain orders that have or have had AGVs.

Fig. 7 Reporting Module (batch)

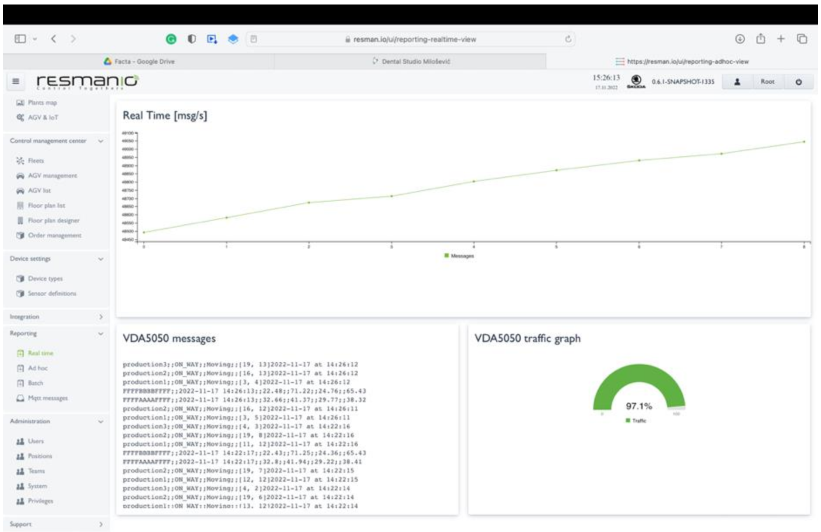

Fig. 8 Reporting Module (online)

4. Discussion

In the case of the testbed platform for Škoda Auto, the proposed platform has to meet various and complex requirements intended to modernize the current system. For example, data streaming and horizontally scalable unified controlling systems intend to advance automotive distributed control systems. The number of conducted AGVs during the test was 150. This is the number of AGVs that Škoda Auto a.s. currently has and we have not recognized performance degradation during the test with the equivalent number of AGV software models.

A Unified Control System as a kind of DSC, branded as Resmanio, can provide AGV status data, reports faults, tracks order material, includes information about order delays, and recalls reports of maintenance intervals. Three modes can be recognized at USC. Emergency mode provides information about process delay and all system data collected through the logs in real-time traffic mode, which allows changing or setting routes. Self-repair mode determines the AGV as a source, determines various navigation procedures for a particular AGV and can be optimized via CI/CD principle. The remaining mode is the Operational mode, which determines the AGV behavior in production plants.

5. Conclusion and Future Work

In this study, we designed a testbed platform to meet Škoda Auto Requirements and expectations for the horizontal scalable and reliable universal and distributed control system. The presented platform confirms that we can manage heterogenous AGVs by using distributed control system as universal.

The case study reported minimal viable platforms for real or near-real-time stream data processing using HiveMQ messaging broker heavily focused on the consumer-facing web enterprise Java application. All the components used to build this platform are provided as open-source software components except Vertica. Vertica may be swapped out for open-source NoSQL databases like Apache Hive and Casandra for analytics. The Resmanio platform serves as the base for delivering the distinctive value of automotive business.

Future work will provide a solution to support various AGVs. Depending on the Skoda Auto preference, future work is also likely to consider: (1) integration with well-established analytical software used for business intelligence (such as Tableau, Qlik and Microsoft's Power BI); (2) integration with identity management and security enhancement tools (such as CAS Protocol for Single Sign-On and other Network monitoring and Application Performance tools); and (3) distributed processing scenarios with multiple-node performance measures and other technologies such as EMQX broker, RabbitMQ and ActiveMQ messaging systems that could be used for integration via APIs from other components of the Skoda Auto a.s. Universal Control System; (4) the MQTT protocol ensures communication with specific AGVs. The entire solution will be developed with the option of implementing algorithms for artificial intelligence and machine learning needs. Based on these algorithms, it would be possible to use additional functionalities based on a large amount of data. For example, it can predict when any AGV/AMR will be down due to failure or battery life by task.

This business-critical application, built in Java, enables the organization to deliver differentiated products and services inside the automotive industry's production. Examples abound in every industry, including software that manages manufacturing processes, financial transactions, health care delivery, data analytics, and client support.

Acknowledgement: *The Solutia s.r.o. the company from Prague, Czech Republic, has sponsored this study and PoC work. The data collection, benchmarks, and the testbed platform setup required considerable collaborative efforts from Solutia s.r.o. (Prague) employees. We are grateful for their contribution and effort while developing, benchmarking, and setting up infrastructure for these testbed purposes.*

Also, this work has been supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

REFERENCES

[1] T. Kalsoom, N. Ramzan, S. Ahmed, and M. Ur-Rehman, "Advances in sensor technologies in the era of smart factory and industry 4.0," *Sensors*, vol. 20, no. 23, p. 6783, 2020, doi: 10.3390/s20236783.

[2] B. Bigliardi, E. Bottani, and G. Casella, "Enabling technologies, application areas and impact of industry 4.0: a bibliographic analysis," *Procedia manufacturing*, vol. 42, pp. 322-326, 2020, doi: 10.1016/j.promfg.2020.02.086.

[3] J. Mehami, M. Nawi, and R. Y. Zhong, "Smart automated guided vehicles for manufacturing in the context of Industry 4.0," *Procedia manufacturing*, vol. 26, pp. 1077-1086, 2018, doi: 10.1016/j.promfg.2018.07.144.

[4] R. Chakma *et al.*, "Navigation and tracking of AGV in ware house via wireless sensor network," in *2019 IEEE 3rd International Electrical and Energy Conference (CIEEC)*, 2019, vol. 14, pp. 233-247, doi: 10.1109/CIEEC47146.2019.CIEEC-2019589.

[5] F. Gul and W. Rahiman, "An integrated approach for path planning for mobile robot using bi-RRT," in *IOP Conference Series: Materials Science and Engineering*, 2019, vol. 697, no. 1, doi: 10.1088/1757-899X/697/1/012022.

[6] D. Hrušecká, R. B. Lopes, and E. Juřičková, "Challenges in the introduction of AGVS in production lines: Case studies in the automotive industry," *Serbian Journal of Management*, 2019, doi: 10.5937/sjm14-18064.

[7] N. Correia, L. Teixeira, and A. L. Ramos, "Implementing an AGV system to transport finished goods to the warehouse," *Advances in Science, Technology and Engineering Systems,* vol. 5, no. 2, pp. 241-247, 2020, doi: 10.25046/aj050231.

[8] M. Čech *et al.*, "Autonomous mobile robot technology for supplying assembly lines in the automotive industry," *Acta logistica,* vol. 7, no. 2, pp. 103-109, 2020, doi: 10.22306/al.v7i2.164.

[9] A. F. Costa, M. do Sameiro Carvalho, M. Henriques, and P. V. Ferreira, "Strategy for the introduction of autonomous driving technologies: a case study in the logistics area of an automotive company," *Procedia Computer Science,* vol. 204, pp. 337-345, 2022, doi: 10.1016/j.procs.2022.08.041.

[10] M. Elsisi and M.-Q. Tran, "Development of an IoT architecture based on a deep neural network against cyber attacks for automated guided vehicles," *Sensors,* vol. 21, no. 24, 2021, doi: 10.1016/j.procs.2022.08.041.

[11] R. Patricio and A. Mendes, "Consumption patterns and the advent of automated guided vehicles, and the trends for automated guided vehicles," *Current Robotics Reports,* vol. 1, no. 3, pp. 145-149, 2020, doi: 10.1007/s43154-020-00007-4.

[12] M. Azangoo, A. Taherkordi, J. O. Blech, and V. Vyatkin, "Digital Twin-Assisted Controlling of AGVs in Flexible Manufacturing Environments," in *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, 2021: IEEE, pp. 1-7, doi: 10.1109/ISIE45552.2021.9576361.

[13] C. L. Garay-Rondero, J. L. Martinez-Flores, N. R. Smith, S. O. C. Morales, and A. Aldrette-Malacara, "Digital supply chain model in Industry 4.0," *Journal of Manufacturing Technology Management,* vol. 31, no. 5, pp. 887-933, 2020, doi: 10.1108/JMTM-08-2018-0280.

[14] K. Gos and W. Zabierowski, "The comparison of microservice and monolithic architecture," in *2020 IEEE XVIth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH)*, 2020: IEEE, pp. 150-153, doi: 10.1109/MEMSTECH49584.2020.9109514.

[15] T. Ueda, T. Nakaike, and M. Ohara, "Workload characterization for microservices," in *2016 IEEE international symposium on workload characterization (IISWC)*, 2016: IEEE, pp. 1-10, doi: 10.1109/IISWC.2016.7581269.