Designing a multi-agent control system for a reconfigurable manufacturing system

Alexandru Matei1[0000-0003-4299-1052], Bogdan Constantin Pirvu1[0000-0003-3961-4539], Radu Emanuil Petruse1[0000-0002-4081-1161], Ciprian Candea2[0000-0002-8869-0194] and Constantin-Bala Zamfirescu1[0000-0003-0128-2436]

[1] Lucian Blaga University of Sibiu, Victoriei Blvd. 10, 550024 Sibiu, Romania
[2] Ropardo, Reconstructiei 2A, 550129 Sibiu, Romania
`{alex.matei, bogdan.pirvu, radu.petruse, constantin.zamfirescu}@ulbsibiu.ro`
`{ciprian.candea}@ropardo.ro`

Abstract. This paper introduces a multi-agent control system for a reconfigurable manufacturing system designed to provide, in the context of Industry 4.0, test-before-invest services by the FIT EDIH. The product assembled is flexible, with multiple possible assembly sequences. The manufacturing system is composed of multiple interchangeable manufacturing cells that allows any layout configuration with autonomous transporter units that move intermediate products from one cell to another. The reference architecture used to instantiate the developed prototype and its multi-agent control system with the required agents, concept and predicate ontology are subsequently presented. The hardware and software implementation are also detailed. The multi-agent system is implemented using SPADE framework. Each manufacturing cell is controlled using 4diac framework while transporters use Robot Operating System. Manufacturing orders are placed by a client using a web interface.

Keywords: Industry 4.0, Multi-Agent Systems, Intelligent Manufacturing Systems, Reconfigurable manufacturing systems, Cyber-Physical Systems.

1 Introduction

The global competition and mass customization are pushing companies to produce better products at a faster pace at an affordable price. For manufacturing systems this implies the capacity to cope effectively with lot size one production. Thus, subsystems and/or workstations of production systems have more and more processing and communication capabilities to improve resource allocation (i.e., who goes where does what) or their composition (i.e., configuration of a workstation/sub-system for a given product). This unprecedented interaction and collaboration within a network of artefacts is called Industry 4.0 and it is made possible by the progress in IT&C and by decentralizing and/or distributing decision-making in more intelligence subsystems throughout the whole production system. Going a step forward, considering sustainability and social aspects as well, Industry 5.0 is the coined term in Europe [1] for human-centered production, where the Industry 4.0 technology is augmented with the flexibility, adaptivity and creativity of humans towards a man-machine symbiosis.

Developing and utilizing such manufacturing systems is not trivial and not cheap, especially for SMEs. European Digital Innovation Hubs (EDIH) [2] are a key instrument within the European strategy to support the digital transformation throughout all economic sectors and regions by providing four main services: training, testing technologies before investing in them, augmentation of the innovation ecosystem and support to find investments. Adequate testbeds to showcase, experiment and train are required to understand how intelligent manufacturing systems work and what their impact can be at an SME.

In the following sections the collaborative manufacturing system within our FIT EDIH is presented. Chapter 2 provides the motivation for this work. Chapter 3 briefly describes the reference architecture together with the hardware description of the system prototype. Chapter 4 presents the design of the multi-agent control system (MAS) for the reconfigurable manufacturing system (RMS) together with software implementation details regarding the MAS. The last chapter concludes the paper together with future developments within the FIT EDIH ecosystem.

2 Rationale

As a result of modern manufacturing processes requirements which face an increase of products complexity and a market demand for customization, we can observe during the last decade an increase interest for Industry 4.0 related topics demonstrated by a surge of scientific publications [3]. Additionally, industrial disruptors (e.g., impact of electric vehicles on the automotive ecosystem [4]) contribute to the manufacturing processes evolution as demand prediction is becoming increasingly challenging. Improving production systems was always a top priority, one of the pioneering researchers being Martin K. Starr which in 1965 introduced the modular production concept [5] suggesting that production modularity can be a solution to avoid manufacturing offshoring. Further, the author reanalyzed his early publication in 2010 [6] where he emphasized the relevancy of the initial concept considering current production challenges. Nowadays, modular production systems, a backbone for the Industry 4.0, have transitioned from the conceptual phase towards commercial application achieving a superior technological maturity. In literature, solutions for solving the optimization problems of RMS include [7, 8]: simulated annealing, genetic algorithms, multiple objective particle swarm optimization, decision trees together with Markov analysis, etc. Moreover, recent supply chains disruptions caused by COVID-19 is accelerating the production systems innovation. In a recent literature review [7], a classification of RMSs is made based on the configuration level: both system and machine level, system level with or without layout design, and machine configuration.

Looking at the latest related modular RMSs, we can characterize them by the following features: Mass customization (MC) or Product modularity (PM) [9]; Human-centered production (HCP); Commercial (C) or Testbed (TB). The RMS in this paper is an HCP testbed that allows both product modularity and mass customization.

Table 1. Selection of available industry or research modular manufacturing systems

Source	**MC/PM**	**HCP**	**C/TB**
UINST Testbed [10]	Both	No	TB
SMART [11]	PM	No	TB
SmartFactory^{KL} [12,13]	PM	No	TB
Bosch Manufacturing Solutions [14]	Both	Yes	C
ScalABLE4.0 [15]	MC	No	TB
EID Robotics [16]	MC	No	C
Huawei I4.0 Testbed [17]	MC	No	TB
Testbed Prague [18]	MC	Yes	TB
Innovation Lab Testbed [19]	Both	Yes	TB
Industry 4.0 Testlab [20]	PM	Yes	TB
This paper	*Both*	*Yes*	*TB*

Besides having a system and machine level configuration level, the designed manufacturing prototype was motivated by the real need to have full accessibility to all hardware and software components without requiring prohibitive third-party interventions for the specific requirements of the regional SMEs supported by FIT EDIH. Therefore, all the hardware components are standard in industrial automation, while for the software ones open-source frameworks and platforms were chosen. The prototype required functionalities were developed in collaboration with the regional companies that are partners of Smart Factory Romania [21].

3 System design

3.1 Reference architecture

This section synthetizes the reference architecture used to develop the RMS prototype. A detailed description of the reference architecture was given in [22] and is a synthesis of some high-level architectures: OSMOSE [23], IoT-A [24], and BEinCPPS [25]. This architecture has BEinCPPS structural perspective as a starting point, a middle domain is added in a similar way to OSMOSE philosophy, while IoT-A is used as a guidance for the underlying architectural reference model. Figure 1 presents the functional perspective of the reference architecture, called SoRA (Socio-centric Reference Architecture). From the structural viewpoint, it has three domains: *Design*, *Socio Cyber-Physical* and *Execution*. The *Design* domain is dedicated to cyber-physical-systems (CPS) design with a focus on the human and social factors. The *Socio Cyber-Physical* domain manages processes, actions, and system through which the human factor is prepared and trained to work with cyber-physical environment. At the same time, the cyber-physical environment is adapted to the characteristics of the human factor as described in the *Design* domain. Briefly, it helps in achieving the required balance between social and cyber-physical factors. The *Execution* domain consists of all the necessary resources to achieve the system functionality.

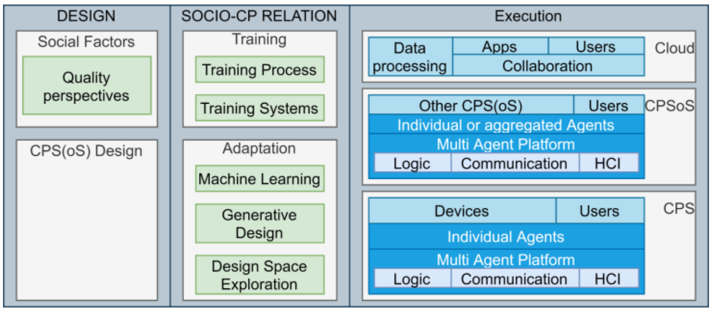


Fig. 1. Functional diagram of SoRA (adapted from [22])

Social factors define relevant human factor aspects like quality standards, security and safety standards, ergonomics, etc. *Training* defines the processes, documentation, and training systems that an operator can follow and access to familiarize with Cyber-Physical System of Systems (CPSoS). *Adaption* consists of several system capabilities that model the operator, work environment and CPS to improve and adapts the manufacturing processes. *Cloud* contains data processing functions and user collaboration applications. *CPSoS* represents the factory level, aggregating multiple CPS from the bottom layer together with external systems. *CPS* represents the device level of the physical equipment, and the human operators present on the manufacturing floor.

When instantiating this reference architecture, in the *Design* phase we chose that some the manufacturing cells will use cobots that allow human-robot collaboration if needed for some advanced assembly processes. In this case, the human operator presence is not a must, but the human can always participate actively in the manufacturing process. The *Socio Cyber-Physical* domain is tackled within a training station for manual operation detailed in [26], while the CPS *adaptation* is explored in some previous works [27, 28]. The *Cloud* is represented by a database and a web application that can access the MAS of the manufacturing system. The *CPSoS* is the entire physical RMS that is composed of multiple *CPS*, each *CPS* being a manufacturing cell.

3.2 Hardware description

The developed RMS is depicted in Figure 5. The system is composed of different manufacturing cells that can be interconnected on any of the four sides using special connectors together with independent AGV that transports the intermediate products from one cell to another.

The system cells have an aluminum frame, a standard square footprint, and a Han-Modular connector on each side for easy connection to another cell. Each cell is equipped with a programmable logic controller (PLC) that controls low level devices (actuators, valves, motors, etc.), a network router that manages the ethernet connection to other manufacturing cells and a small factor PC with integrated screen for interfacing with human operators and other advanced devices or systems (cobots, CNC, MAS, data storage, etc.). Each cell is also equipped with sliding windows that when lowered they fence the AGV inside the working area and can also be raised to allow the AGV to move from one cell to another. A possible configuration of several manufacturing cells can be seen in Figure 2.

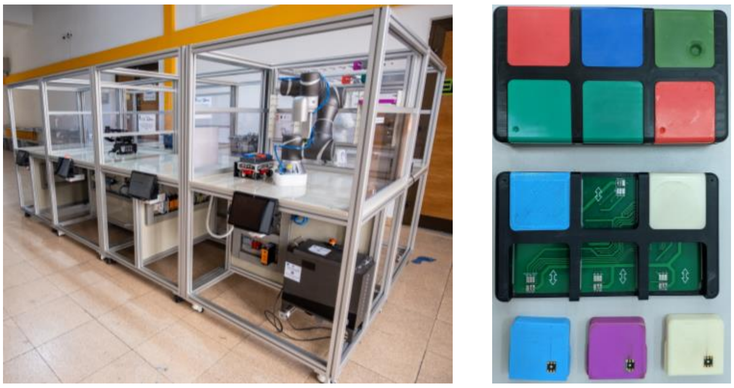


Fig. 2. Reconfigurable manufacturing system (Left). Assembled product (Right) with all slots personalized (top), intermediate assembly (center), modules (bottom)

Next, we detail each manufacturing cells responsibilities. **Warehouse** cell contains a place to store product parts together with a cobot that can load the product parts into the storage area or to unload product parts onto the AGV for transportation. **Assembly** cell contains pneumatic mechanism to unload parts carried by the AGV and assemble them. **Customization** cell contains tools to personalize the product by engraving an image using a CNC. **Testing** cell contains a verification device – industrial camera – that checks if the product is assembled and personalized correctly. **Charging** cell acts as a parking and charging station for AGVs. **Packaging** cell contains a cobot that unloads the final product from the AGV and places it inside a packaging box. This cell can also be accompanied by a human operator that can work collaboratively with the robot to further customize the product and order by placing stickers, paint, smooth the rough edges if any, or do other special requests from the client for example. **Barebones** is an empty module that can be used as parking space for AGVs. Represents the starting point for creating new manufacturing cells.

The product that is assembled on this RMS is a modular tablet – see Figure 2 – composed of a main screen, bus that has 6 slots where three types of modules can be connected: battery module, speaker module or flashlight module. The modules are available in different colors and can be further customized by engraving or with stickers. Each module can be connected to any of the six available slots, increasing the possible final tablet configurations that the final user can order.

Figure 3 describes the possible assembly sequences depending on the product customization. The technological process is flexible enough to allow multiple paths in assembling a customized product for the end user. The system can start with any of the following three possible operations: customize the modules, assemble the bus with modules or assemble the bus with the tablet. By not having a fixed assembly sequence, the manufacturing cells can have a more balanced workload.

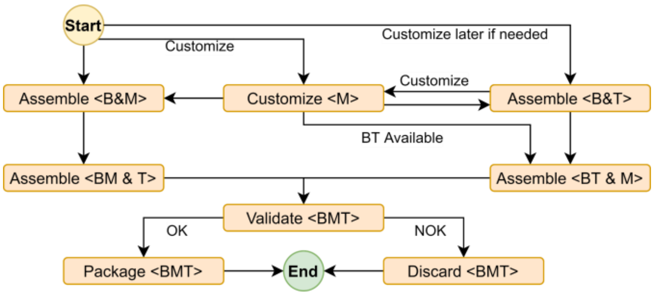


Fig. 3. Technological process for product assembly. B – Bus; M – Modules; T – Tablet Screen

4 Prototype implementation

4.1 Multi-Agent Control System

To achieve the information model specific to the reference architecture associated with the manufacturing line, a multi-agent simulation was performed. The MAS presented follows the IEEE standard on Industrial Agents [29], a hybrid loosely coupled interaction mode between high level control (agents) and low-level control (real-time hardware) devices. For the design of the simulation and respectively the design of the control level of the system, the classic method of developing a MAS was used, which involves four main phases: identify the agents, describe the interaction between agents and specific behaviors, define the ontology, implement, and test.

In the first step, we identified the following classes of agents: **Client**, **Order**, **Order Management** (OMA), **Knowledge Management** (KMA), and **Resource Agents** for every manufacturing cell and transporter units. **Client Agent** represents the human client that places order in the system using an intuitive user interface. He can create, view, update or delete orders by interacting with the OMA, fetch possible product configurations and customization from KMA and can monitor the ongoing orders by interrogating the Order Agent assigned to the order placed by the client. **OMA** manages current orders and process order requests from Client Agent. It also has responsibility to instantiate or terminate new Order Agents when necessary. An **Order Agent** is created for each new order placed in the system. It manages a single order, and it has the responsibility to a plan and negotiate with other resource agents the execution of the order. He queries the KMA regarding his personalized order to get a complete recipe, task and activities needed to be followed to execute the order. It communicates with the order agents and negotiates, assigns, and monitors the tasks. **KMA** contains information about the RMS and the products that can be manufactured using the current layout, infrastructure, and parts available. It responds to Client Agent request regarding the available products and possible customizations to be made or to Order Agent regarding assembly recipes and tasks specific to its current order. Each of the **Resource Agents** – **Warehouse**, **Assembly**, **Customization**, **Testing**, **Charging**, **Packaging** and **AGV** (Autonomous Guided Vehicle) are responsible to advertise their manufacturing or transport services, negotiate task execution and schedule accepted tasks. Negotiating tasks involves finding an available time window based on the request from an Order Agent that meets requirements like maximum execution time, earliest time it can start, or an execution deadline. In response to the Order Agent request, they can accept, decline, or propose a new task execution window that meets the order requirements. In Figure 4, a snippet of the interaction sequence, after the task negotiation phase, between an AGV agent, Warehouse agent and an Assembly agent can be observed.

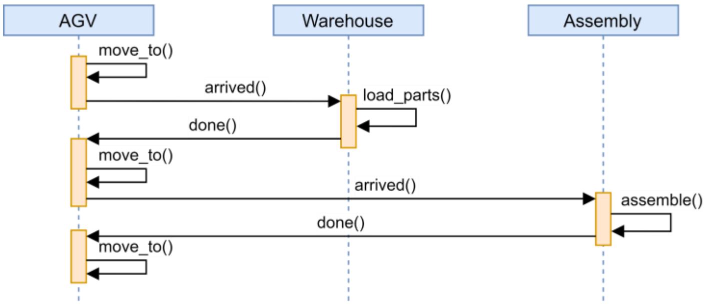


Fig. 4. Interaction sequence diagram between agents

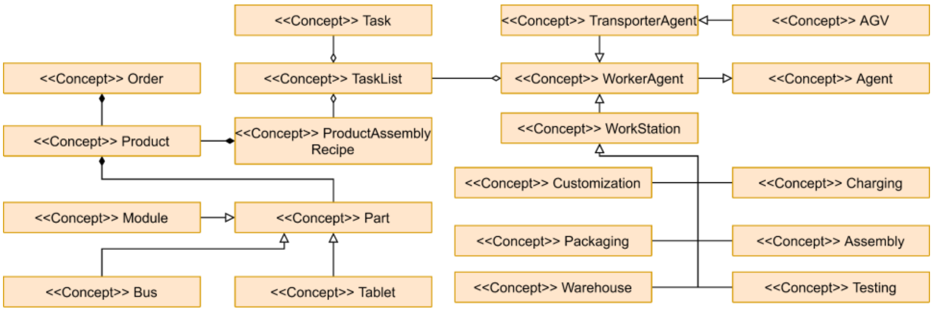


Fig. 5. Concept ontology

Figure 5 and Figure 6 presents relevant parts of the concept ontology and predicate ontology respectively, both created to define the RMS's MAS. The ontology for concepts broadly describes each concept, but also highlights the relationships between concepts as some more advanced concepts inherit from basic concepts. This ontology contains concepts for Parts, Product, Agents, Tasks and Recipes. In an ontology, predicates play the role of connecting elements between other concepts, describing actions and states. The ontology of predicates is graphically highlighted with green in Figure 6, together with their relationship to other concepts. It contains the necessary predicates that describe transport actions or capabilities, task dependencies, parts reservation or what parts compose a certain product.

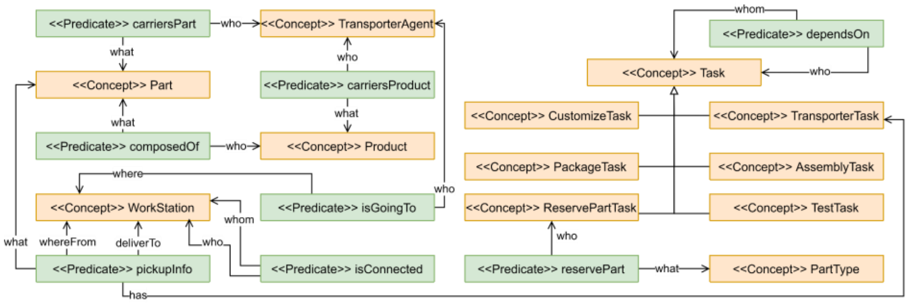
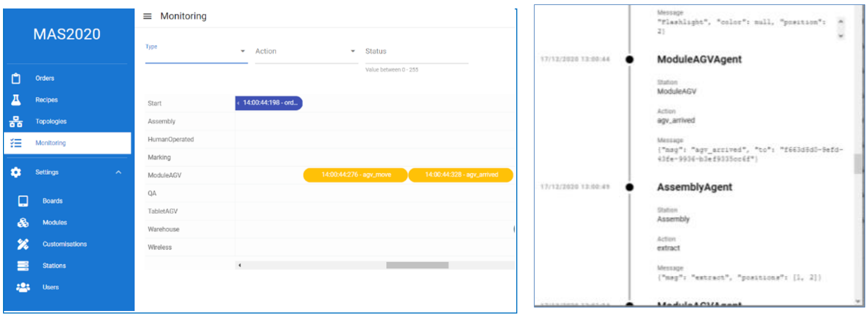


Fig. 6. Predicate Ontology

Beside defining the system on an abstract semantic level, the concepts and predicates defined in the ontology contains the domain knowledge and are used by the agents of the MAS to be able to communicate in a structured yet flexible way. The ontologies define the information model efficiently, allowing easy data and information retrieval. Ontologies are also meant to facilitate collaboration and prototype usage by dispersed team members belonging to different SMEs.

4.2 Software implementation details

For the client user interface, a web portal – see Figure 7 – was developed using Node.js and Quasar framework that is connected to the MAS through an API that allows configuring the manufacturing line and placing manufacturing orders to the system. Advantages of this framework includes being opensource, is compatible with different browsers and auto adapts to screen size, making it easy to be accessed from a mobile device, without much code changes. The web portal has features like creating an account for clients, placing personalized order for logged in clients, monitoring order status. An administrator user has features like configure possible products parts, configure available product personalization modes, enable manufacturing cells, visualizing all orders, order status, and monitoring – as an overview of the entire MAS or for each order, a message sequence together with message details and timestamps. For archiving purposes, an SQL database managed by PostgreSQL is used.



Message
"flashlight", "color": null, "position": 2}

17/12/2020 13:00:44
ModuleAGVAgent
Station
ModuleAGV
Action
agv_arrived
Message
{"msg": "agv_arrived", "to": "f663d8d0-9efd-43fe-9936-b3ef9335cc6f"}

17/12/2020 13:00:49
AssemblyAgent
Station
Assembly
Action
extract
Message
{"msg": "extract", "positions": [1, 2]}

ModuleAGVAgent

Fig. 7. Monitoring interface. Right – overview of MAS. Left – single order message sequence

For the selection of MAS developing framework, the communication capabilities of the platform, scalability, transparent integration of people and agents and support for integration with IoT systems were considered. For MAS implementation, we used the SPADE (Smart Python Agent Development Environment) [30] framework, an open-source MAS platform available on [31]. It is written in Python, uses the XMPP (extensible messaging and presence protocol) instant messaging protocol for communication between agents, supports FIPA metadata and has a web-based interface available. SPADE is also capable to integrate with other FIPA compliant MAS platform due to the flexibility of XMPP [30]. The programming model for SPADE agents is based on behaviors, including not only classical behaviors, such as: Cyclic, One-Shot, Periodic, Time-Out and Finite State Machine, but also BDI (belief desire intention) behavior [32], allowing the mixing of procedural, object-oriented and logic programming in the same agent. It allows creating personalized more complex behaviors. All agents exist inside an Agent Logic Container (ALC). Communication with the *Web Interface* component is done through the API exposed by ALC.

AGVs are controlled on low level by a microcontroller and on high level by an SoC that runs Robot Operating System (ROS) on top of a Linux distribution. Low level control manages motors, drivers, battery, sensors, and wireless charging. High level control manages AGV functions like environment mapping, positioning inside the map, path planning and obstacle avoidance. Each manufacturing cell PLC is programmed using 4diac, an opensource framework for distributed control of industrial processes based on IEC61499 industrial standard.

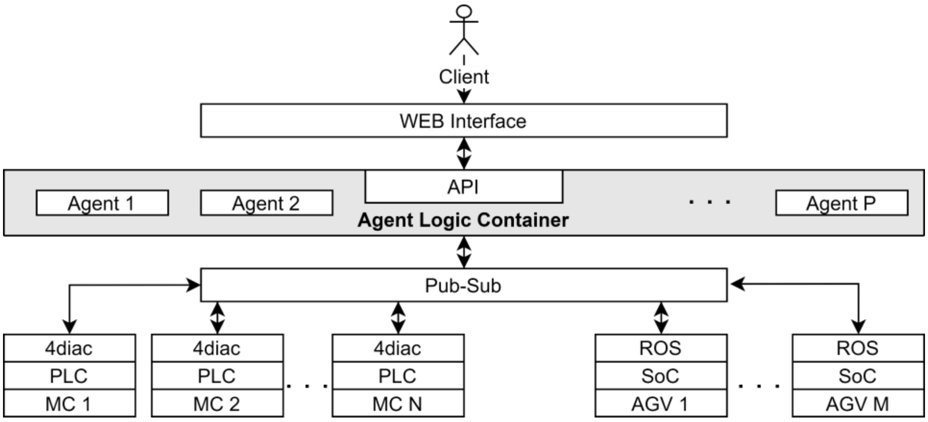


Fig. 8. System overview

Communication between hardware level control – 4diac, ROS – and MAS is realized through MQTT pub-sub protocol for its lightweight, IoT oriented capabilities. Each manufacturing cell has its own id that is used to define its topic filter. The JSON formatted messages received from MAS are translated by 4diac or ROS to the hardware system. Table 2 contains a part of the low-level interface for the assembly cell: the topics preceded by the station id and message required fields. Figure 8 describes an overview of the information flow and the main systems involved, from high level control systems and human interfaces, down to the hardware devices level.

Table 2. Assembly Station low level interface

Subscribing Topics	Publishing Topics
Init Check Topic: *id*/Initialization	Returns initialization status Topic: *id*/InitializationOut RETURN {"status": X, ["failed": "debug message"]}
Check Status Topic: *id*/CheckStatus	Returns station status [*free/busy/fault*] Topic: *id*/CheckStatusOut RETURN {"status":X}
Extract Topic: *id*/Extract Body: {"Position1":True, "Position2": False, …}	Returns extraction status [done/fault] Topic: *id*/ExtractOut RETURN {"status": X, ["failed": "debug message"]}
Deploy Topic: *id*/Deploy Body: {"Position1":True, "Position2": False, …}	Returns deploy status [done/fault] Topic: *id*/DeployOut RETURN {"status": X, ["failed": "debug message"]}

5 Discussion and further development

The paper presents a prototype of an RMS that is controlled by a MAS developed using SPADE framework. It provides basic functionalities for real-time layout reconfiguration, mass customization, and product modularity. The prototype will be exploited as a testbench within the FIT EDIH by desiring SMEs that will have the opportunity in the next three years to test-before-invest in different scenarios based on their specific use cases. This will be done in the recently won project FIT EDIH in the call DIGITAL-2021-EDIH-01 funded by the European Commission.

Therefore, its further development is market-driven and depends on the use-cases required by regional SMEs. Apart from the AGV, the implementation was straightforward and did not pose any specific challenge. Requiring a suitable small size AGV, that was unavailable on the market, we had to build it inhouse. While developing the AGV, we encountered hardware issues in integrating all the requited components in a small factor encasing, or software issues in localization and mapping due to sensor being partially obstructed that needed AI augmentation [26]. Therefore, the control systems have not been yet tested with multiple AGVs in human robot collaboration scenarios.

Acknowledgements

This work is supported by the DiFiCIL project (contract no. 69/08.09.2016, ID P_37_771, web: http://dificil.grants.ulbsibiu.ro), co-funded by ERDF through the Competitiveness Operational Programme 2014-2020.

References

1. European Commission; Directorate-General for Research and Innovation; Renda, A.; Schwaag Serger, S.; Tataj, D.; Morlet, A.; Isaksson, D.; Martins, F.; Mir Roca, M.; Hidalgo, C.; Huang, A.; Dixson-Declève, S.; Balland, P.; Bria, F.; Charveriat, C.; Dunlop, K.; Giovannini, E. Industry 5.0, a Transformative Vision for Europe: Governing Systemic Transformations towards a Sustainable Industry; (2022). https://doi.org/10.2777/17322.
2. European Digital Innovation Hubs | Shaping Europe's digital future, https://digital-strategy.ec.europa.eu/en/activities/edihs, Accessed July 2022
3. Miqueo, A., Torralba, M., Yagüe-Fabra, J.A. Lean Manual Assembly 4.0: A Systematic Review. Appl. Sci., 10, 8555 (2020). https://doi.org/10.3390/app10238555
4. Arden, W. The EV disruption. 2019.. https://www.fia.org/marketvoice/articles/ev-disruption, Accessed July 2022
5. Starr, M.K. Modular production—A new concept. Harvard Business Review, 43(6), pp. 131–142 (1965).
6. Starr, M.K. Modular production – a 45-year-old concept. International Journal of Operations & Production Management, 30 (1), pp. 7-19 (2010). https://doi.org/10.1108/01443571011012352
7. Sabioni, R.C., Daaboul, J., Le Duigou, J. Optimization of Reconfigurable Manufacturing Systems Configuration: A Literature Review. In: Roucoules, L., Paredes, M., Eynard, B., Morer Camo, P., Rizzi, C. (eds) Advances on Mechanics, Design Engineering and Manufacturing III. JCM 2020. Lecture Notes in Mechanical Engineering. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-70566-4_67
8. Sabioni, R.C., Daaboul, J., Le Duigou, J. Concurrent optimisation of modular product and Reconfigurable Manufacturing System configuration: a customer-oriented offer for mass customisation, International Journal of Production Research, 60:7, pp. 2275-2291, (2022), DOI: 10.1080/00207543.2021.1886369
9. Doran, D., Hill, A. A review of modular strategies and architecture within manufacturing operations. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 223(1), pp. 65–75 (2009). https://doi.org/10.1243/09544070JAUTO822
10. Kim, D.Y., Park, J.W., Baek, S. et al. A modular factory testbed for the rapid reconfiguration of manufacturing systems. J Intell Manuf 31, pp. 661–680 (2020). https://doi.org/10.1007/s10845-019-01471-2
11. Kovalenko, I., Saez, M., Barton, K., Tilbury, D. SMART: A system-level manufacturing and automation research testbed. Smart and Sustainable Manufacturing Systems, 1(1), pp. 232–261 (2017). http://doi.org/10.1520/SSMS20170006
12. Gorecky, D., Weyer, S., Hennecke, A., Zühlke, D. Design and instantiation of a modular system architecture for smart factories. IFAC-PapersOnLine, 49(31), pp. 79-84 (2016). https://doi.org/10.1016/j.ifacol.2016.12.165

13. First skill-based application in operation in Kaiserslautern. https://smartfactory.de/en/first-skill-based-application-in-operation-in-kaiserslautern/, Accessed July 2022
14. BOSCH. How future production looks like – The Modular Production System. https://www.boschmanufacturingsolutions.com/news-and-highlights/modular-production-system/, Accessed July 2022.
15. ScalABLE4.0. https://www.inesctec.pt/en/projects/scalable4-0, Accessed July 2022.
16. EID Robotics. Building an Automated Assembly Line. https://eidrobotics.com/building-an-automated-assembly-line-how-to-minimize-costs-and-ramp-up-time/, Accessed: July 2022.
17. Huawei. Introducing Huawei Industry 4.0 Test Bed. https://e.huawei.com/en/material/local/ccb33d2eeab84e68a70938335d1a6e83, Accessed July 2022
18. Testbed Prague. https://ricaip.eu/testbed-prague/, Accessed July 2022
19. L Watson. Use a testbed to enable Industry 4.0 manufacturing. Manufacturing AUTOMATION. https://www.automationmag.com/use-a-testbed-to-enable-industry-4-0-manufacturing/, Accessed May 2021
20. Industry 4.0 Testlab. https://techlab.uts.edu.au/lab/industry-4-0-testlab/, Accessed July 2022
21. Smart Factory Romania. https://smartfactoryromania.ro/en/, Accessed June 2022
22. Ghetiu, T., Pirvu, B. Insights into SoRa: A Reference Architecture for Cyber-physical Social Systems in the Industry 4.0 Era. In Proceedings of the International Conference on Innovative Intelligent Industrial Production and Logistics - IN4PL, ISBN 978-989-758-476-3, pp. 47-52 (2020). DOI: 10.5220/0009982300470052
23. Gusmeroli, S., Agostinho, C., Lucena, C., Sesana, M., Felic, A., Fischer, K., Monteleone G. OSMOSE: a Paradigm for the Liquid-Sensing Enterprise. IWEI Workshops 2015.
24. Bauer, M., et al. Internet of Things – Architecture IoT-A Deliverable D1.5 – Final architectural reference model for the IoT v3.0. IoT-A (257521) 2013.
25. Isaja, M., Fischer, K., Rotondi, D., Coscia, E., Rooker, M. D2.2 - BEinCPPS Architecture & Business Processes. 2017. Ref. Ares(2017)3817966 - 30/07/2017
26. Matei, A., Ţocu, NA., Zamfirescu, CB., Gellert, A., Neghină, M. Engineering a Digital Twin for Manual Assembling. In: Margaria, T., Steffen, B. (eds) Leveraging Applications of Formal Methods, Verification and Validation: Tools and Trends. ISoLA 2020. Lecture Notes in Computer Science, vol 12479. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-83723-5_10
27. Gellert A, Sorostinean R, Pirvu B.C. Robust Assembly Assistance Using Informed Tree Search with Markov Chains. Sensors. 22(2), 495 (2022). https://doi.org/10.3390/s22020495
28. Gellert, A., Sarbu, D., Precup, S.A., Matei, A., Circa, D., Zamfirescu, C.B. Estimation of Missing LiDAR Data for Accurate AGV Localization. IEEE Access, 10, pp. 68416-68428 (2022). doi: 10.1109/ACCESS.2022.3185763.
29. IEEE Standards Committee. IEEE Recommended Practice for Industrial Agents: Integration of Software Agents and Low-Level Automation Functions. IEEE Std 2660.1-2020, pp.1-43, 2021, doi: 10.1109/IEEESTD.2021.9340089
30. Palanca, J., Terrasa, A., Julian V., Carrascosa, C. SPADE 3: Supporting the New Generation of Multi-Agent Systems. IEEE Access, 8, pp. 182537-182549 (2020). doi: 10.1109/ACCESS.2020.3027357.
31. Palanca, J. SPADE Documentation. https://spade-mas.readthedocs.io/en/latest/index.html, Accessed June 2022
32. Palanca, J., Rincon, J., Julian, V., Carrascosa, C., Terrasa, A. Developing IoT Artifacts in a MAS Platform. Electronics, 11, 655 (2022). https://doi.org/10.3390/electronics11040655