Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 42 (2020) 406–413

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

International Conference on Industry 4.0 and Smart Manufacturing (ISM 2019)

Open Interfaces for Connecting Automated Guided Vehicles to a Fleet Management System

Walter Quadrini[a,]*, Elisa Negri[a], Luca Fumagalli[a]

[a]*Department of Management, Economics and Industrial Engineering, Politecnico di Milano, p.zza Leonardo da Vinci, 32, 20133 Milan Italy*

* Corresponding author. Tel.: +390223999269. *E-mail address:* walter.quadrini@polimi.it

Abstract

The recent digitalization of manufacturing companies, driven by "Industry 4.0" guidelines, introduces a series of challenges and opportunities both for enterprise and research environments. In this perspective, a debated subject is the interfacing of hardware, because logistic challenges and production assets are moving towards the adoption of Internet-of-Things-based Cyber-Physical Systems (CPS), whose flexibility allows to rethink control architectures that are traditionally based on ISA 95 and IEC 62264 formalized hierarchies. More versatile architectures also pave the way to the so-called horizontal integration. This work focuses on intra-logistics, aiming at defining a simple architecture enabling CPS assets' monitoring and control. This allows the development of an open platform that enables an easy and simultaneous connection of multi-vendor hardware, overcoming the current legacy systems' gap of requiring a significant effort to do so. The architecture is then substantiated by the development of a component interfacing Automated Guided Vehicles (AGV) and managing their navigation thanks to a Fleet Management System (FMS), which is back fed by the same AGVs.

Keywords: CPS; AGV; FIWARE; open architecture; modular architecture; Fleet Management System; Robot Operating System; ROS
Type your keywords here, separated by semicolons ;

1. Introduction

Since its introduction in 2011, Industry 4.0 has been one of the most discussed topics both in research and in industry [1]. The "digitization of processes" challenges the traditional manufacturing thanks to the accessibility of digital technologies that allow easy integration of interconnected components located in the shop floor. As a consequence, manufacturing is going to be transformed into fully integrated facilities, able to communicate with one another, boosting flexibility, speed, productivity and quality [2].

Industry 4.0 is based on the concepts of Cyber-Physical Systems (CPS), the Internet of Things (IoT) [3], [4], and Internet of Services (IoS) [5]. These Internet-related technologies enable the generation of added value for companies and are based on a continuous networked communication that allows a stable interaction and exchange of information not only between humans (Customer-to-Costumer, C2C) and human to machine (Customer-to-Machine, C2M), but also between the machines (Machine-to-Machine, M2M) [6].

It is easy to understand that the management of the flow of data and information represents one of the most critical aspects inside Industry 4.0 paradigm. The control of this flow allows both real-time monitoring of the shop-floor activities and rapid decision making, improving productivity of the facility [6].

In particular, since according to Gamberi et al. [7] the material handling process represents 15% up to 70% of the total manufacturing cost, an increasing number of enterprises

10.1016/j.promfg.2020.02.055

are taking into consideration the opportunity to switch to highly automated solutions [8] in order to save up on costs. Despite of the technological readiness of the automated solutions [9], this type of technology is a critical constraint for many companies, and especially for Small and Medium-sized Enterprises (SMEs), because of the lack of standardization and the unavailability of easy-to-interface low-cost equipment [10]. Furthermore, the adoption of solutions such as Automated Guided Vehicles (AGVs) is hindered by the fact that most of the models commercially available use guidance systems referring on fixed tapes on the floor, following them slavishly. This limits the potential adoption in manufacturing companies since the production and logistic flexibility promised by Industry 4.0 is consistently impeded [11].

This paper is intended to propose open interfaces to connect AGVs to a Fleet Management System (FMS) in order to be compatible to as large an AGV manufacturer/vendor base as possible.

The paper is structured as follows: Section 2 sets the state of the art of the existing architectures for production, Section 3 frames the research design, Section 4 shows the development of the proposed interfaces, Section 5 illustrates some application cases, Section 6 reports some discussions and final conclusions and Section 7 opens to further developments.

2. Background

A great relevance inside the aforementioned Industry 4.0 paradigm is held by the so-called Automation Pyramid, based on IEC 62264 [12], which itself is based on the ANSI/ISA 95 [13] standard, evolution of the Purdue Enterprise Reference Architecture (PERA) model [14]. The pyramid structure, explained in Fig. 1, expresses the typical hierarchical model of production systems automation before the Industry 4.0 paradigm, where relations between the various layers of the pyramid were based upon the input received from the higher level of the information system and the output given to the lower one [12].

Initial attempts at standardizing and making the represented pyramid more flexible were carried out by [15], [16], especially in the internal logistics subject [17].

As Fig. 1 shows, the pyramidal layers are built on a "layer 0", or “field”, which includes the hardware involved in the production process, and then build up layer 1 (the Programmable Logic Controller – PLC - layer), layer 2 (SCADA layer), layer 3 (Manufacturing Execution System – MES - layer) and layer 4 (Enterprise Resource Planning – ERP - layer). Each of them based on top of the previous one [12].

By embedding IoT technologies in the production system, according to Industry 4.0 paradigm, it can be stated that the layer 0 hosts CPS [18], [19], which is characterised by hardware equipped with onboard electronics and logics enabling any addressing from the upper layers. In this context, the aforementioned pyramid can be tilted to the one of Fig. 2, allowing every layer of the hierarchy to communicate with the layer 0 [15].

This new representation highlights, in a network perspective, the proximity every layer has with the field, allowing the data gathering directly from it. These data are needed by the layers 1 to 4 to monitor and control the production system and to eventually support a decision process through the computation of Key Performance Indicators (KPIs). However, this kind of rotation of the pyramid introduces the issue of the communication between the CPS and the above levels. At the time being, in fact, countless communication protocols, hundreds of Enterprise Service Buses (ESBs) [20] and dozens of integrated Platforms as a Service (iPaaS) [21] have been developed to efficiently solve the "vertical integration" problem, academically defined as the integration of processes across the entire organization, via networking of smart production systems, smart products and smart logistics [22]. Vertical integration can also be referred to as the capability of the applications belonging to a layer of the pyramid to communicate with the other layers. The high number of different protocols and technological solutions introduces also the parallel issue of the "horizontal integration", meaning the interconnection and communication among resources from different functional areas [23]. In this perspective, the interfaces of the CPS play a crucial role in in Industry 4.0 production systems, since they are supposed to enable a multicast communication either with the layers 1 to 4 of the Automation Pyramid (allowing vertical integration), or with other layers of other Automation Pyramids referring to different areas (enabling horizontal integration) [24].

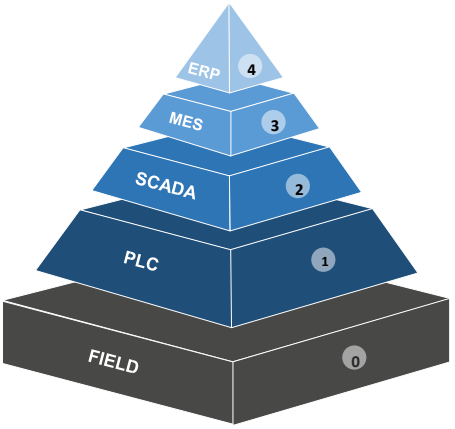

Fig. 1. The automation pyramid

In an industrial environment, the integration challenge is particularly perceived by the eventual automated logistics system, which has to interface with several layers of the Pyramid: the logistic system needs in fact to communicate with the PLCs of the single machines to load/unload the handled materials, but has also to address some information to the MES to track the products. Moreover, it is also supposed to be able to speak with a Warehouse Management System (WMS), to be instructed about the location of the goods it has to delivery to the production assets.

The high effort implied to meet these requirements is often assumed by the so-called system integrators. Their role is to develop and customize a software layer able to interface, wrap, parse and exchange data between the logistic system and the other ones which have to communicate with it.

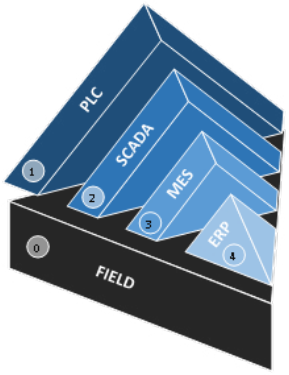

Fig. 2. Tilted automation pyramid

3. Research design

3.1. Research framework

The need of scalability over different hardware systems, makes the communication necessarily lay on a software architecture, whose aim should be to embody a deployment as modular as possible, which, as stated by Yin et al., can lead to a sub-optimal decision process [23]. This is achieved through the usage of open interfaces, which are supposed to overcome the scalability problems arisen by legacy software and proprietary protocols.

Given the importance of an investigation of open interfaces for the internal logistics domain, this work takes into consideration AGVs as material handling systems, since they are fully compliant with the computation and networking features characterizing the concept of CPS [25]. A FMS for AGVs may include a set of different software elements, that, in Table 1, are introduced and put in relation with the relative layers of Fig. 2 [10].

Table 1. Main architectural elements

Name	Description	Relative layers with Figure 2
Task Planner (TP)	Software module devoted to the scheduling and monitoring of the logistic operations	Layer 3 (MES)
Interface with the operators	It sends to the TP the jobs to be scheduled and it displays to the operators the current state of the jobs and assets	Layer 2 (SCADA)
Mapping module	A module to create a map of the environment and to share it with the connected AGVs	Layer 2 (SCADA)
Interface for a set of sensors	This element may be composed of interfaces for sensors of different nature. As examples: cameras to feed the mapping module, proximity sensors to detect the position of an AGV in a particular frame; transducers to sense the presence of products to be handled in a loading/unloading buffer	Layer 0 (field)
Interface for a set of AGVs	They interface the AGV fleet to be managed	Layer 0 (field)

Sloping the elements listed in Table 1 down to the pyramid of Fig. 2, these elements can be referred to the layers highlighted in Fig. 3.

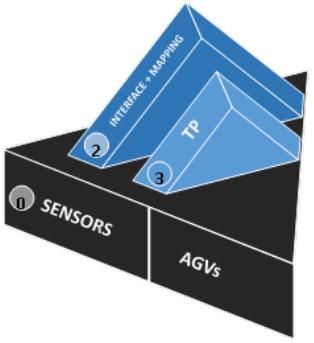

Fig. 3. Elements grouped according to the pyramid of Fig. 2

3.2. Research objectives

Among all the displayed elements, human interfaces and sensors are not an exclusive prerogative of logistics systems and their interfacing with informative systems is a subject widely studied, both for industrial applications [26] and for other environments (e.g. home automation [27] and computer networks [28]). On the other hand, TP (as the core of any FMS), AGVs and the mapping module are typical modules of an automated logistics system. In the recent days, several research contributions concerning the interconnections of these modules needed in a FMS have been provided: in particular, a proof of concept about openly interfacing layer 3 (MES layer) of Fig. 2 to a CPS fleet has been exposed [29], while Seder et al. proposed a full integration scenario of a CPS-based FMS with a particular focus on the mapping system, but without mentioning the interfaces which allow the AGV to connect to the FMS [10].

This contribution aims at proposing the interfaces to integrate AGVs in an FMS enhancing service discovery functionalities. In order to reach this broad objective, the following sub-objectives are pursued (refer to Fig. 4):

1. creation of a **message/context-oriented open middleware**, easy to interact with for the above and below applications;
2. development of an **AGV interface**, i.e. a component able to manage both the tasks assigned to the AGV and the navigation operations needed to accomplish these tasks and with an open interface to allow low-level communication with transducers ad actuators of the AGV;
3. elaboration of a set of **messages to be exchanged between the middleware and the AGV interface**: this set of messages should be properly structured in order to be as general as possible, to cover all the capabilities of a flexible logistic solution;
4. elaboration of a set of **messages to be exchanged between the AGV interface and the AGV**.

These features are mapped in Fig. 4.

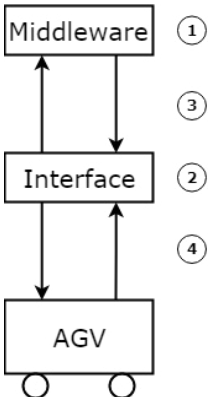

Fig. 4. Proposed model

3.3. Research methodology

For the implementation of the robotic code on the interface, Robot Operating System (ROS) platform [30], [31] has been chosen. In fact, in addition to being a de-facto standard for research about robotics [32], ROS is designed with a package-based modular design which perfectly scales down to the architecture-oriented work. Moreover, the communication policy is based on two different approaches: services and messages. Services are the classical web-service like policy, where a client node asks for some information to a server node. Messages are the more interesting approach, since they refer to a pub/sub structure: this policy makes an arbitrary number of nodes to publish or subscribe over a topic, such that every time a node publishes a new message on a topic, the subscribers are notified about the new message and can access the related information [30]. This feature enhances an event-triggered communication, which is more compliant to an automated logistic scenario (where, for example, an AGV is warned about the availability of a product to be handled upon ending the processing on a machine).

This proposed architecture to connect the AGV is in alignment with the reference frameworks proposed by the H2020 program in the European research projects L4MS [10], BEinCPPS [33], FASTEN [34] and follows the guidelines given by the previous FITMAN project [35], [36]

4. Development

4.1. Message/context-oriented open middleware

This contribution takes, as a proof of concept, the integration of a generic AGV in the architecture above, interfacing it with a middleware thanks to the development of a component accomplishing tasks following the Logistic Task Language domain (LoTLan) [37].

The middleware component was chosen as a FIWARE [38] generic enabler: Orion Context Broker (OCB) [39] which can be basically connected as a Global Data Space in a Data Distribution Service (DDS) [40] and which, technically, is a Mongo NoSQL database with a REST API based on the Open Mobile Alliance's Next Generation Service Interface (NGSI) [41].

4.2. AGV interface

The intercommunication between NGSI and ROS messages has been ensured by FIROS, an open source ROS module which basically translates ROS messages into FIWARE entities by decomposing the input structure, copying the single fields in the target entity and adding a timestamp to avoid duplicates (and vice versa from entities to ROS messages) [42]. The communication between the context broker and FIROS is managed through REST calls [43]. In Listing 1, the generic model of a movement LoTLan task has been provided. The fields include:

```
Header {header}
point_id {UUID}
task_id {UUID}
motion_id {UUID}
point {array}
max_velocity {array}
max_acceleration {array}
motion_area {array}
sequence {int}
```

Listing 1. Example of LoTLan Task

1. a header, containing a sequential number, a timestamp and a frame;
2. a Universally Unique Identifier (UUID) which refers the target point into the graph provided by the mapping module;
3. a UUID which uniquely identifies the task the AGV must accomplish (where "task" may be defined as a combination of points to be reached and actions to be performed by the AGV);
4. a UUID which uniquely identifies the motion the AGV must perform;
5. a set of absolute coordinates identifying the point the AGV must reach;
6. an array which defines the velocity constraints on the different axes;
7. an array which defines the acceleration constraints on the different axes;
8. an array of points defining the corners of a polygon which represents the area the AGV can move in;
9. an incremental integer number, expressing the progression of the task to be accomplished.

It can be noticed that the interface widely uses UUIDs to identify tasks, actions and motion assignments; it also assigns a random UUID to declare the AGV it manages, as a de-facto standard in multiagent systems [44].

The "task" class has been introduced in order to structure data in a clearer way: accomplishing a task means moving to a target goal through many waypoints and, upon reaching destination, performing a succession of actions. The grouping of points and actions is set using the task_id field named in Listing 1. Actions and motions with the same task_id belong to the same task and therefore create a sequence (the last field of Listing 1). It is important to notice that the component has been developed such that in a task there cannot be an action followed by a movement. In order to do so, the task must be split in smaller sub-tasks. This is clearly illustrated by the

state diagram shown in Fig. 5.

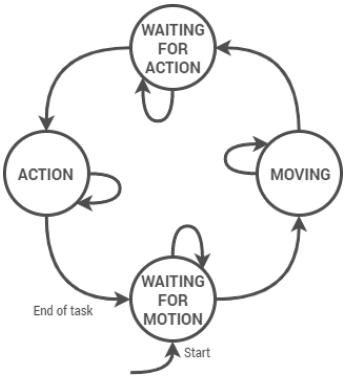

Fig. 5. Task state diagram

In order to allow an easy deployment of the developed interface among a heterogeneous robot fleet, the interface has been completed with a navigation system. This has been done to make the AGV agnostic about the goal of its action, keeping the control inside the interface: in this way, the AGV receives from the interface only low level speed setpoints and is supposed to provide only raw data coming from its sensors (Micro Electro-Mechanical Systems – MEMS – and laser-scanners). A deeper explanation about the data and their structures is given in 4.3 and 4.4 sub-sections.

4.3. Messages between middleware and AGV interface

As displayed in Fig. 6, for what concerns the messages exchanged between the robot component and the middleware, five different communication topics are used:

- action_channel: the interface subscribes to this topic, where it receives action commands (i.e. impositions of logical states such as loading/unloading/idle); these messages are processed and forwarded to the robot;
- motion_channel: the interface subscribes to this topic, where it receives motion commands (as showed in Fig. 6); these messages are processed to send low-level motion control messages to the robot drives;
- movement_channel: the interface publishes on this topic periodic updates about its position and kinematical data, to allow the higher level information systems and applications a complete perspective over the AGVs running in the facility;
- status_channel: the interface publishes on this topic data about AGVs' machine states;
- description_channel: the interface publishes on this topic data about the robot physical properties and capabilities. This makes the TP aware of every new AGV connecting to the system with respect to its performable actions, enhancing eventual Plug&Play functions proper of the FMS.

4.4. Messages between AGV interface and AGV

For what concerns the messages exchanged between the AGV interface and the AGV, three topics have been opened:

- cmd_vel: on this topic the interface publishes the cmd_vel ROS message [45];
- status_channel_AGV: on this topic status information is sent by the AGV to the interface (e.g. battery state of charge);
- description_channel_AGV: on this topic, a message containing the AGV's properties and capabilities is shared with the interface.

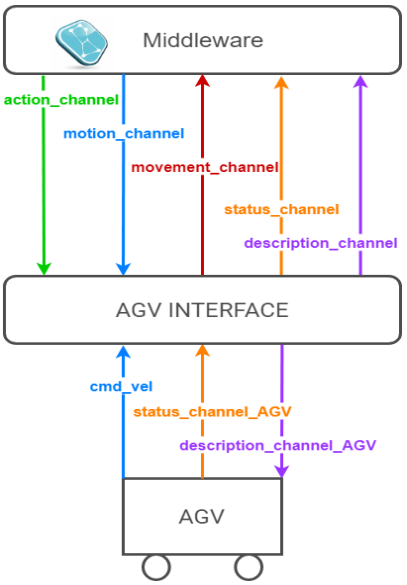

Fig. 6. Communication topics

All the messages exchanged were defined as ROS messages, but several packages and implementations are available to interface non-ROS applications and hardware with ROS ones through ROS messages [46].

For what concerns the implemented navigation algorithms, the ROS navigation stack was followed as reference [45]. For the localization, an Adaptive Monte Carlo Localization algorithm [47] has been configured, while the local path planner ensuring collision avoidance has been set through the Dynamic Window Approach (DWA) [48], preferred to the Trajectory Rollout [49], because of its higher flexibility with respect to computational constraints and maximum acceleration supported by the AGV.

5. Application cases

The proposed architecture has been tested with all its subcomponents, in different environments, with different hardware and in the context of EU research projects.

At first, the entire stack was simulated in a Player/Stage environment [50], to simulate AGVs behaviour in the Industry 4.0 Lab of the School of Management of Politecnico di Milano [51], which was chosen as it is a real-like industrial plant, compliant with IEC 62264 standard and RAMI4.0 paradigm [52].

The test consisted in assigning two tasks to two different AGVs (rendered as blue and green rectangles in the Player/Stage environment shown in Fig. 7). The tasks consisted in:

- Two different sets of target points to be reached by AGVs in a predefined order (one set per AGV). Those coordinates have been sent by querying the OCB over the motion_channel through well-formatted RESTful calls;
- Two waiting actions (one per AGV), which, according to Fig. 5, ended the task. Those coordinates have been sent by querying the OCB over the action_channel through well-formatted RESTful calls.

The assigned tasks were performed in the simulation environment correctly and in the right sequence, as displayed in Fig. 7. Furthermore, an obstacle (the red square in Fig. 7) has been placed in order to test the obstacle avoidance capabilities of the ROS standard navigation stack [45]: the capability proved successful.

Fig. 7 shows as top figure the starting environment and as bottom figure the environment at the end of the test.

The test has been also validated on Jmeter [53] which, querying the status_channel, notified about the conclusion of the tasks.

The application was then tested in all its features at the Innovation Centre Nikola Tesla (ICENT) in the context of European project Logistics for Manufacturing in SMEs (L4MS). ICENT research department provided a physical in-house created AGV. For its specific design, the robot needed a custom navigation stack, provided by ICENT researchers [54] and based on D* [55]. The simulation test, based on a routine operation described by a Croatian manufacturer of Polyethylene blow films, was developed on Visual Components digital twin suite [56] and demonstrated that the proposed message structure was able to handle real-like industrial tasks. The manufacturer joined the L4MS project to investigate AGV solutions able to minimize the manipulation of heavy Polyethylene rolls for its operators, in order to reduce the risks of occupational injuries. Hence, the industrial environment was recreated in ICENT laboratories, in order to reproduce a real-like scenario. The complete setup of the equipment the test procedure has been described by Seder et al. [10].

The logistic workflow of the test, compliant with the architectural elements of table 1, is however here listed:

1. the operator at the loading station starts the task through a Human-Machine Interface (HMI);
2. the task request is sent to the TP, which assigns it to the AGV;
3. the AGV moves to the target position, acknowledging the TP when the goal is reached;
4. the TP notifies to the operator that the AGV is ready be loaded;
5. the operator loads the AGV and pushes a button (interpreted as a sensor) to acknowledge the TP that the AGV is ready to move;
6. the TP creates a new movement request (containing the unloading station coordinates) and sends it to the AGV;
7. the AGV moves to the unloading position, acknowledging the TP when the goal is reached;
8. the TP notifies to the HMI of the operator at the unloading station that the AGV is ready to be unloaded;
9. the operator unloads the AGV and pushes a button to acknowledge the TP that the AGV has been unloaded and is ready to move;
10. the TP creates a new movement request (containing the waiting area coordinates) and sends it to the AGV;
11. the AGV moves to the waiting area, acknowledging the TP when the goal is reached and ending the task.

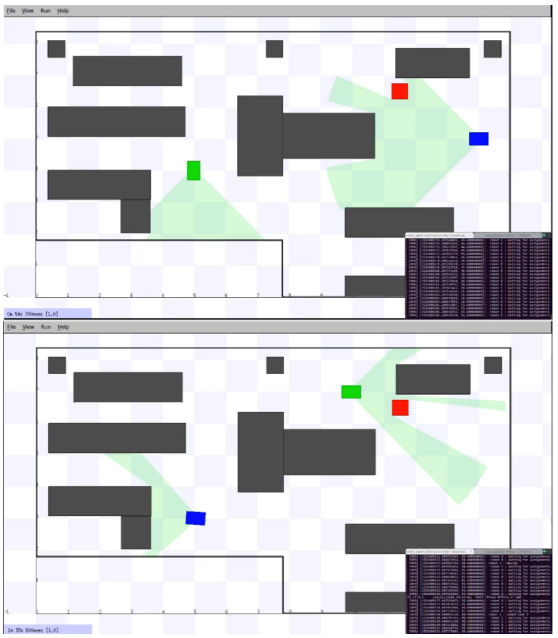

Fig. 7. Player/Stage test

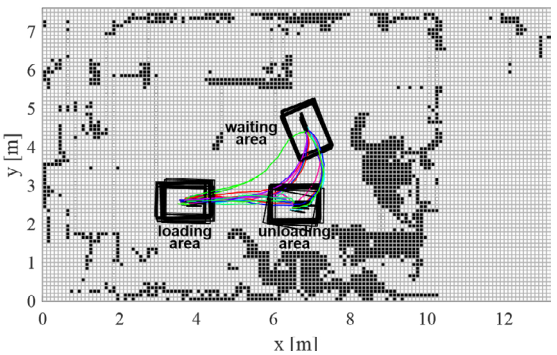

Fig. 8. Long run of the experiment

Fig. 8 displays the paths of the AGV in the simulated scenario over different tests. The suitability of the proposed architecture and of the developed interface to carry out real logistics tasks in a production facility have been therefore demonstrated. The results also show the main benefits of the interface as an open platform that acts as a container, which allows deploying a custom made module designed for a specific robotic system [10].

6. Discussion and conclusions

Based on the proofs of concept developed in simulation and in field experiments as illustrated in Section 5, the proposed architecture and open interfaces appeared to be affordable and robust. The defined set of messages seemed to be adequate to manage the AGV in a logistic scenario and to make it recognisable by the TP in a Plug&Play perspective.

Furthermore, the project Business Experiments in Cyber-Physical Production Systems project (BEinCPPS) provided the proof of concept for an aggregation of the OPC Unified Architecture (OPC UA) [57] data mined from the I4.0Lab's plant through the IDAS Generic Enabler [21].

7. Future works

Thanks to the achieved results, NGSI entities published on the OCB could be used to warn the TP about the availability of finished products and to instruct consequently the AGVs, as summarized in Fig. 9.

Future developments of the proposed work could involve the effective full horizontal integration between the production informative system and the FMS, enabling timely (or even preemptive) AGV calls.

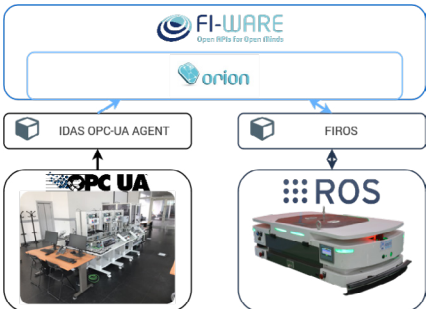

Fig. 9. Connection of FIROS and OPC-UA agent to Orion

Acknowledgements

This work is supported by European Union funded projects L4MS (GA 767642), FASTEN (GA 777096) and BEinCPPS (GA 680633). The authors also gratefully acknowledge the team of professor Ivan Petrovic and Andrea Caielli for their precious support.

References

[1] L. Da Xu, E. L. Xu, and L. Li, "Industry 4 . 0 : state of the art and future trends," *Int. J. Prod. Res. ISSN*, vol. 56, no. 8, pp. 2941–2962, 2018.

[2] M. Rüßmann *et al.*, "Industry 4.0:Future of Productivity and Growth in Manufacturing," *Bost. Consult. Gr.*, no. April, p. 20, 2015.

[3] E. Negri, L. Fumagalli, and M. Macchi, "A review of the roles of Digital Twin in CPS-based production systems," *Procedia Manuf.*, vol. 11, no. June, pp. 939–948, 2017.

[4] L. Cattaneo, M. Rossi, E. Negri, D. Powell, and S. Terzi, "Lean thinking in the digital Era," in *IFIP Advances in Information and Communication Technology*, 2017, vol. 517, pp. 371–381.

[5] R. Ruggaber, "Internet of Services SAP Research Vision," in *WETICE '07 Proceedings of the 16th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2008.

[6] A. Varghese and D. Tandur, "Wireless requirements and challenges in Industry 4.0," in *Proceedings of 2014 International Conference on Contemporary Computing and Informatics, IC3I 2014*, 2014, pp. 634–638.

[7] M. Gamberi, R. Manzini, and A. Regattieri, "An new approach for the automatic analysis and control of material handling systems: Integrated layout flow analysis (ILFA)," *Int. J. Adv. Manuf. Technol.*, vol. 41, no. 1–2, pp. 156–167, 2009.

[8] M. Bonini *et al.*, "Evaluating Investments in Emerging Automation Solutions for Logistics," in *Proceedings of the Hamburg International Conference of Logistics (HICL)*, 2015, vol. 20, no. February 2017.

[9] A. De Carolis, M. MacChi, E. Negri, and S. Terzi, "Guiding manufacturing companies towards digitalization a methodology for supporting manufacturing companies in defining their digitalization roadmap," in *2017 International Conference on Engineering, Technology and Innovation: Engineering, Technology and Innovation Management Beyond 2020: New Challenges, New Approaches, ICE/ITMC 2017 - Proceedings*, 2018, pp. 487–495.

[10] M. Seder *et al.*, "Open Platform Based Mobile Robot Control for Automation in Manufacturing Logistics," in *Joint 12th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles 1st IFAC Workshop on Robot Control (Joint CAMS and WROCO 2019*, 2019.

[11] A. Keller, J. Meyer, A. W. Colombo, and R. Harrison, "Enhancing the behaviour of system of cyber-physical systems through environment parameters: An Industry4.0-compliant approach," in *Proceedings: IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018, vol. 1, pp. 2884–2889.

[12] IEC, "IEC 62264-2: Enterprise-control system integration - Part 2: Objects and attributes for enterprise-control system integration"," in *IEC 62264-2 standard*, 2015.

[13] International Society of Automation, *Enterprise-Control System Integration Part 2 : Object Model Attributes*. 2001.

[14] P. Bernus, L. Nemes, and G. Schmidt, *Handbook on Enterprise Architecture*. Springer, 2003.

[15] L. Fumagalli, S. Pala, M. Garetti, and E. Negri, "Ontology-based modeling of manufacturing and logistics systems for a new MES architecture," in *APMS 2014, IFIP Advances in Information and Communication Technology 438 (PART I)*, 2014, pp. 192–200.

[16] E. Negri, L. Fumagalli, M. Macchi, and M. Garetti, "Ontology for Service-Based Control of Production Systems," in *APMS 2015, Part II, IFIP AICT 460*, 2015, pp. 484–492.

[17] E. Negri, S. Perotti, L. Fumagalli, G. Marchet, and M. Garetti, "Modelling internal logistics systems through ontologies," *Comput. Ind.*, vol. 88, pp. 19–34, 2017.

[18] National Science Foundation (NSF), "Cyber-Physical Systems (CPS)," *NSF 15-541*, 2014. .

[19] L. Fumagalli, E. Negri, O. Severa, P. Balda, and E. Rondi, "Distributed control via modularized CPS architecture. Lessons learnt from an industrial case study," in *IFAC-PapersOnLine*, 2018, vol. 51, no. 11, pp. 803–808.

[20] F. Menge, "Enterprise Service Bus," 2007.

[21] M. Marian, "iPaaS: Different Ways of Thinking," *Procedia Econ. Financ.*, vol. 3, pp. 1093–1098, 2012.

[22] L. M. Camarinha-Matos, R. Fornasiero, and H. Afsarmanesh, "Collaborative networks as a core enabler of industry 4.0," in *IFIP Advances in Information and Communication Technology*, 2017, pp. 3–17.

[23] Y. Yin, I. Kaku, and C. Liu, "Product architecture, product development process, system integrator and product global performance," *Prod. Plan. Control*, vol. 25, no. 3, pp. 203–219, 2014.

[24] W. M. Mohammed *et al.*, "Generic platform for manufacturing execution system functions in knowledge-driven manufacturing systems,"*Int.J.Comput.Integr.Manuf.*,vol.31,no.3,pp.262–274, 2017

[25] M. Garetti, L. Fumagalli, and E. Negri, "Role of Ontologies for CPS Implementation in Manufacturing," *MPER - Manag. Prod. Eng. Rev.*, vol. 6, no. 4, pp. 26–32, 2015.

[26] M. Tandel, U. Joshi, and A. Golhani, "Scripting engine for SCADA HMI," in *2017 2nd International Conference for Convergence in Technology, I2CT 2017*, 2017, pp. 492–496.

[27] Y. T. Lee, W. H. Hsiao, C. M. Huang, and S. C. T. Chou, "An integrated cloud-based smart home management system with community hierarchy," *IEEE Trans. Consum. Electron.*, vol. 61, no. 1, pp. 1–9, 2016.

[28] S. S. Yau and F. Karim, "An adaptive middleware for context-sensitive communications for real-time applications in ubiquitous computing environments," *Real-Time Syst.*, vol. 26, no. 1, pp. 29–61, 2004.

[29] E. W. T. Ngai, T. K. P. Leung, Y. H. Wong, M. C. M. Lee, P. Y. F. Chai, and Y. S. Choi, "Design and development of a context-aware decision support system for real-time accident handling in logistics," *Decis. Support Syst.*, vol. 52, no. 4, pp. 816–827, 2012.

[30] M. Quigley *et al.*, "ROS: an open-source Robot Operating System," *IEEE Conf. Robot. Autom.*, vol. 3, no. 3.2, 2009.

[31] G. Bardaro, D. A. Cucci, L. Bascetta, and M. Matteucci, "A simulation based architecture for the development of an autonomous all terrain vehicle," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8810, pp. 74–85.

[32] P. Estefo, J. Simmonds, R. Robbes, and J. Fabry, "The Robot Operating System: Package reuse and community dynamics," *J. Syst. Softw.*, vol. 151, pp. 226–242, 2019.

[33] C. Palasciano, B. Thiede, M. Taisch, and C. Herrmann, "Deployment architecture for energy and resource efficient cyber physical systems," in *IFIP Advances in Information and Communication Technology*, 2017, pp. 159–167.

[34] R. Arrais *et al.*, "Application of the Open Scalable Production System to Machine Tending of Additive Manufacturing Operations by a Mobile Manipulator," in *19th EPIA Conference on Artificial Intellligence*, 2019, pp. 345–346.

[35] O. Lazaro, A. Gonzalez, and J. Sola, "FITMAN future internet enablers for the sensing enterprise: A fiware approach & industrial trialing," in *CEUR Workshop Proceedings*, 2015.

[36] K. Jansson *et al.*, "Forecasting impact of technology developed in RandD projects: The FITMAN approach," in *2015 IEEE International Conference on Engineering, Technology and Innovation/ International Technology Management Conference, ICE/ITMC 2015*, 2015.

[37] P. Detzner, J. Jost, and T. Kirks, "A Novel Task Language for Natural Interaction in Human-Robot Systems for Wharehouse Logistics," in *14th International Conference on Computer Science and Education, ICSSE 2019*, 2019.

[38] T. Zahariadis *et al.*, "FIWARE lab: Managing resources and services in a cloud federation supporting future internet applications," in *Proceedings - 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, UCC 2014*, 2014.

[39] C. Steinmetz *et al.*, "Ontology-driven IoT code generation for FIWARE," in *Proceedings - 2017 IEEE 15th International Conference on Industrial Informatics*, 2017, pp. 38–43.

[40] G. Pardo-Castellote, "OMG Data-Distribution Service: Architectural overview," in *Proceedings - 23rd International Conference on Distributed Computing Systems Workshops, ICDCSW 2003*, 2003, pp. 200–206.

[41] Á. Alonso, A. Pozo, J. M. Cantera, F. de la Vega, and J. J. Hierro, "Industrial data space architecture implementation using fiware," *Sensors (Switzerland)*, vol. 18, no. 7, 2018.

[42] R. Limosani, A. Manzi, L. Fiorini, P. Dario, and F. Cavallo, "Connecting ROS and FIWARE: Concepts and Tutorial," *Stud. Comput. Intell.*, vol. 778, pp. 449–475, 2019.

[43] F. Herranz, J. Jaime, I. González, and Hernández, "Cloud robotics in FIWARE: A proof of concept," in *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, 2015, pp. 580–591.

[44] Z. Balkić, D. Šoštarić, and G. Horvat, "GeoHash and UUID identifier for multi-agent systems," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, pp. 290–298.

[45] E. Marder-Eppstein and M. Ferguson, "navigation - ROS Wiki," *ROS.org*, 2016. .

[46] C. Crick, G. Jay, S. Osentoski, B. Pitzer, and O. C. Jenkins, "Rosbridge: ROS for non-ROS users," in *Springer Tracts in Advanced Robotics*, 2017, vol. 100, pp. 493–504.

[47] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "{Monte Carlo} localization for mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999, pp. 1322–1328.

[48] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, 1997.

[49] B. P. Gerkey and K. Konolige, "Planning and Control in Unstructured Terrain," in *ICRA Workshop on Path Planning on Costmaps*, 2008.

[50] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The Player / Stage Project: Tools for Multi-Robot and Distributed Sensor Systems," in *Proceedings of the 2003 International Conference on Advanced Robotics*, 2003, pp. 317–323.

[51] L. Fumagalli, M. Macchi, A. Pozzetti, M. Taisch, G. Tavola, and S. Terzi, "New methodology for smart manufacturing research and education: The lab approach," in *Proceedings of the Summer School Francesco Turco*, 2016, pp. 42–47.

[52] K. Schweichhart, "Reference Architectural Model Industrie 4.0 (RAMI 4.0) - An Introduction," *Plattform Industrie 4.0*. 2016.

[53] Apache, "Apache JMeter - Apache JMeter™," *Apache JMeter*. 2014.

[54] M. Dakulovic, C. Sprunk, L. Spinello, I. Petrovic, and W. Burgard, "Efficient navigation for anyshape holonomic mobile robots in dynamic environments," in *IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 2644–2649.

[55] A. Stentz, "Optimal and Efficient Path Planning for Partially Known Environments," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 1997, pp. 3310–3317.

[56] V. Kuts, T. Otto, T. Tähemaa, and Y. Bondarenko, "Digital twin based synchronised control and simulation of the industrial robotic cell using virtual reality," *J. Mach. Eng.*, vol. 19, no. 1, pp. 128–144, 2019.

[57] F. Zezulka, P. Marcon, Z. Bradac, J. Arm, T. Benesl, and I. Vesely, "Communication Systems for Industry 4.0 and the IIoT," *IFAC-PapersOnLine*, vol. 51, no. 6, pp. 150–155, 2018.