Asynchronous Communication Between Modular Cyber-Physical Production Systems and Arduino Based Industrial Controllers

Fábio M. Oliveira, André Rocha, Duarte Alemão, Nelson Freitas, Jose Barata

Asynchronous Communication between Modular Cyber-Physical Production Systems and Arduino Based Industrial Controllers

Fábio M. Oliveira[1,2] , André Rocha[1,2], Duarte Alemão[1,2], Nelson Freitas[1,2], José Barata[1,2]

[1] School of Science and Technology, NOVA University of Lisbon, Campus de Caparica, Portugal
[2] Uninova-CTS, Campus de Caparica, Portugal
{fmo, andre.rocha, d.alemao, n.freitas, jab}@uninova.pt

Abstract. In a new world where the virtual and physical world is more and more connected, there is a need to project physical devices as digital clones, but the inverse is also true, projecting physical objects from software assets. The proposed work is an approach to connect virtual (software) and the physical (machines) twins using two asynchronous solutions: persistent bi-directional communication and publish subscribe methods on Arduino based controllers. The focus will be in the interaction of virtual and physical reality in order to track the products mainly for academic and investigation proposes but with focus on the applicability on legacy controllers from shop floors, which were not conceived and projected to have these features.

Keywords: Modular cyber-physical production systems, Virtual environment, Industrial Agents, Industrial controllers, Asynchronous communication, MQTT, Websockets, FIPA

1 Introduction

The fourth industrial revolution [1] is on its way, gaining more and more enthusiasts in academy and industry. This revolution is characterized by digitalization, focusing on technology and digital transformation, concentrating on adding value to users, integration and gathering of new data and, on developing communication technologies to create sustainable solutions or highly customizable products with agile and flexible approaches [2]. Transferring all the capabilities proposed by this new paradigm is not easy and the industry is trying to adapt in a cost sustainable way.

The legacy controllers used on some assembly lines are not suitable to face this new paradigm and the substitution of those can carry a lot of investment [3]. Also, a holarchy should be present on the shop floor to enable a fast, flexible, agile, and resilient production line that features data creation and consumption [3].

A holarchy can be executed on a software platform of agents Multiagent Systems (MAS). Retrofitting some of these legacy controllers with Arduino-based microcontrollers or Raspberry Pi micro-computers seems to be a cost-effective solution with low downtimes.

In this scenario, the question that arrives is: “Is it possible to use a MAS high-level system with a retrofitted robot?”

Our proposal is to use synchronous communication protocols to provide interaction between MAS and legacy controllers through Arduino-based microcontrollers in order to use a holarchy at a high level and keep the retrofitting costs low.

This paper is divided into the following structure: chapter two where the works of integration between MAS and the physical systems are explored, chapter three where we propose a framework, chapter four in which the demonstration scenario is explained, chapter five where our results are shown and a conclusion and future work on chapter six.

2 Related Work

In recent past years, we assisted in the emergence of Cyber-Physical Systems (CPS) applied to several challenges. CPS enables a set of flexible features over data for the final users ranging from processing data acquired by tiny sensors to managing large data collection. It also enables to sharing of data, provides security, and facilitates application support [4]. CPS provides not only access to data but also empowers the connection between different computing devices, namely concurrent processing in distributed environments and supporting information sharing in heterogeneous scenarios guaranteeing responses from the user queries at a suitable time [5][6]. When CPS is applied to the fields of production and manufacturing sites it is denominated Cyber-Physical Production Systems (CPPS). This new concept of CPS, CCPS, is suitable for the development of the fourth industrial revolution [7][8].

In industry, the possibility of having a logical representation of a physical asset has many advantages since it reduces the complexity of implementing systems. Nevertheless, linking physical components with logical representations is not an easy task [9]. Some works integrate solutions based on the Internet of Things (IoT) where new sensors are copulated to the PLC to harvest data [10].

Smart factories take the advantage of CPPS to face the product's shorter life cycles and high customization, required by the clients. A smart factory is composed of vertical integration (between management software like MES – Manufacturing Execution System, or ERP – Enterprise Resource Plan, to the shop floor) and horizontal integration (between shop floor machines)[11]. Some authors already detected gap’s in the vertical integration, for instance between the management layers and the producing machinery (robots and PLC) at the bottom [12].

A smart factory has, as one of the key factors, the ability to use agile, robust, and dynamic production lines. To couple with this challenge, one of the solutions is to use distributed and reconfigurable control systems [13]. These systems have an holonic approach [14], with a focus on modularity, that can be implemented using a Multi-Agent System (MAS) [15].

When using a MAS as a control system, an architecture must be chosen. The two big groups of architecture are centralized coordination, where one or more Agents are responsible to mediate and coordinate the actions of other Agents (an example can be consulted at [16]) or the decentralized coordination approach where all Agents are responsible for creating and executing a production plan. Focusing on the decentralized approaches, each Agent can communicate with the others through asynchronous messages [17]. One of the most used protocols to communicate between Agents is the Agent Communication Language standardized by Foundation for Intelligent Physical Agents (FIPA) [18]. On the decentralized architecture itself, several approaches were explored like the one depicted in Figure 1 [19], where every entity communicates with each other using the MAS platform, or the other depict in Figure 2 [20], where some entities only communicates with the ones chosen, but the most suitable to use on production sites is the Product Agent architecture [21].

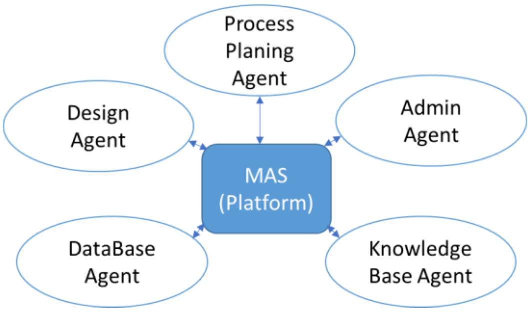

Fig. 1 - Multi Agent platform prototype

Fig. 2 - Team of agents on AGV challenge

When using the Product Agent architecture approach, the Product Agent (PA) is responsible to handle the list of actions to be done to the product and negotiate with the Resource Agents (RAs) to which manufacturing processes the product must be submitted. In order to, it must search for suitable RAs, request and schedule the actions [22].

An approach for the RA is presented in Lepuschitz, et al., [23] but, as stated in Ribeiro and Hochwallner [24] there are still some challenges in the integration between the Cyber-Representation and the production system. The work of Ding, et al., [25] identifies as a challenge the synchronization loop and states as some hypotheses the development of industrial networks, the data protocols, and the interfaces. Even in recent literature such as Hyre, et al., [26] where the DT is clearly defined, the communication between the physical and cyber world is not clearly demonstrated. The work of Samir, et al., [27] uses both persistent bi-directional communication and publish-subscribe methods on a truck company, through its Plant Service Bus (PSB), but does not use the MAS environment.

3 Proposed Framework

The proposed work is a framework with three levels: Multi-Agent System (MAS) environment, an Integration Layer, and the Real/Physical System (as depicted in **Fig. 3**).

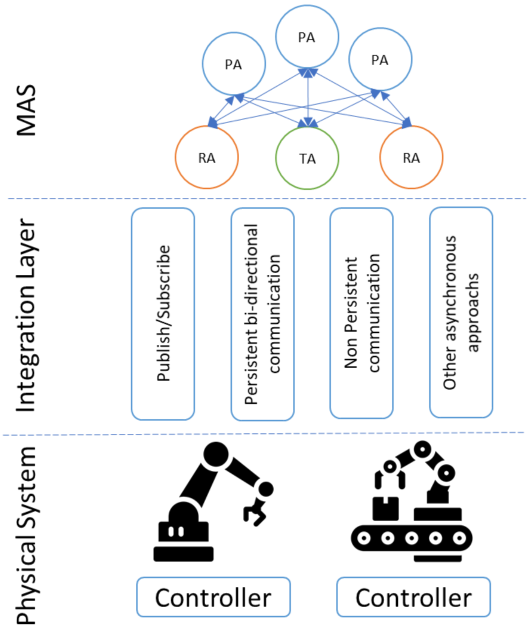

Fig. 3 - Proposed framework

On the first layer, the MAS environment, it is applied a Product / Resource architecture as presented in the IDEAS project [28]. The Product Agents (PA) and the Resource Agents (RA) can communicate with each other's using a FIPA compliant protocol to require and make available Skills on the system. The RAs are modular cyber-physical production systems representations. A Transport Agent (TA), also FIPA compliant, can provide transport for the products from the beginning of the line to the exit and between RAs. The RA has the capability of using asynchronous communication mechanisms to provide low-level integration. On the other side, Physical Systems also have the capability to use asynchronous communication mechanisms to exchange messages. The main contribution of this kind of framework is that between the RA and the Physical systems there is no need of using a FIPA compliant protocol since the proposed integration layer requires only the use of an asynchronous protocol chosen by the end-user. This integration layer is not dependent on the asynchronous protocol chosen and more than one protocol can be used at a given time.

The advantage to using this approach is to enable controllers that already have asynchronous protocols made available by the manufacturers (for instance OPC protocol) also complaint to receive messages from a MAS environment, in particular FIPA compliant communications, without the need of instantiating their own Agents on the default hardware which can be a big challenge.

4 Demonstration Scenario

Using JAVA and JADE, a virtual demonstration scenario was created. It consists of several conveyors and three stations. In this scenario, the conveyors are managed by TA and the RAs associated with stations D_7, D_8, and D_9. The RA associated with station D_7 communicates with the corresponding controller through the MQTT protocol. The RA associated with station D_8 communicates with the controller through WebSocket protocol and station D_9 only executes a virtual skill.

The graphical interface, that helps understand how the PAs are running on the production lines is present in Fig. 4. Several RAs offering several skills can be deployed but the ones associated with station D_7 and station D_8 will always use asynchronous communication protocols. RAs associated with station D_9 will always use FIPA protocol to execute the virtual skill.

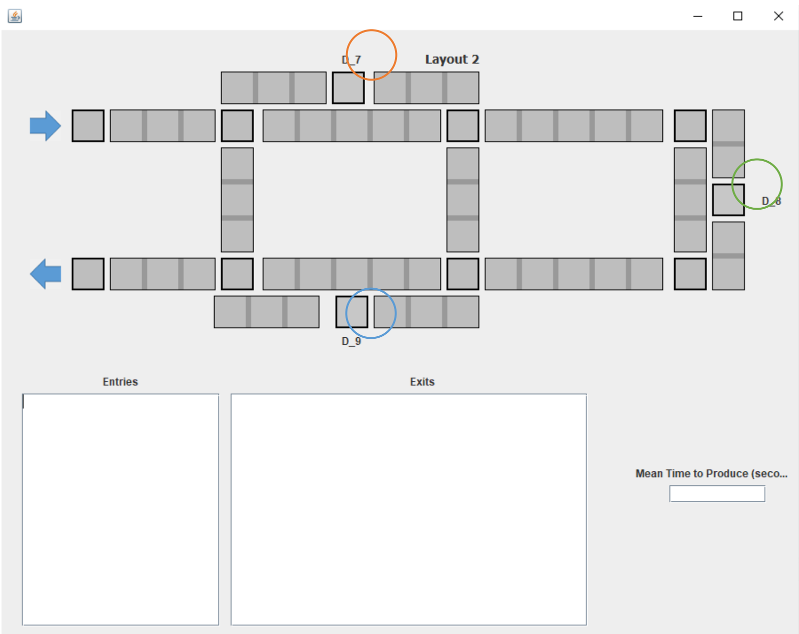

Fig. 4 - Simulation environment with three stations: D_7, D_8, D_9

The controllers used are M-Duino since they are compatible with Arduino and can use 24V sensors and actuators.

Although the physical kit also has a conveyor, it is considered that it is part of the station and not part of the transport system.

When the product reaches the position D_7 or D_8, a message is sent through one of the message protocols (MQTT or Websockets, respectively) and the physical station waits for a product to be detected on the sensor from the first conveyor. As soon as the product is detected on the sensor the station executes the skill and when the product is detected on the sensor from the last conveyor a message is sent, through the same communication protocol to the virtual environment. From this point on, the TA resumes its work and conduces the product through the rest of the virtual system.

Depending on the 'skills' required by the product, it can go from one physical station to another. In Figure 5 the blue arrow indicates the sensor from the position on the first conveyor (initial position) and the green arrow indicates the sensor from the position on the last conveyor (final position).

Fig. 5 - Physical station

Since JAVA is a high-level programing language that supports threads, the communication protocols can be executed at the same time. Therefore, a product can be at station D_7 and another at D_8 and another on station D_9 and the entire system is able to continue. On the controller side, although possible to react to several messages in parallel from the software point of view, it makes no sense to force a second *product* to be processed at the same time at the same station, since each station is only available for a *product* at a time. The control and management of the operation is controlled by the MAS and the *TA* knows if the position on the station is free or not to deploy the *product.* The message protocol is only activated when the simulation environment deploys a *product* in the station (the *product* is on the *resource*).

It is out of the scope of this work to handle to multiple requests when the physical system is busy or to make priorities from different *products.*

5 Results

To demonstrate that this is a valid framework a set of tests was made. The first batch of tests conducted on the demonstration scenario was designed to verify that the concept was able to be applied. The second batch of tests was done to evaluate the performance of two asynchronous protocols.

Fifty products were created, and they asked for a skill available on a resource that was associated with station D_7 (usage of MQTT protocol). Afterward, another fifty products were created that asked for a skill available on a resource that was associated with station D_8 (usage of WebSocket protocol).

All *products* were able to be produced in both situations.

In the second batch of tests, it was measured the round trip time (RTT) of the messages between the Resource Agent and the hardware. Fifty products were created to use each protocol, in three different runs.

When a product arrives at the station, a message is sent with the timestamp. The controller that receives the message replies with the same content. Upon the delivery of this message, the MAS is able to calculate the RTT using the “current” timestamp and the message timestamp. The execution time is not within the scope of this work since it can depend on the skill required, so the skill was empty and only the RTT was measured. The resume of RTT is depicted in Table 1.

Table 1 - RTT resume table

Protocol - Run / RTT (ms)	Min	Max	Average	Standard deviation
MQTT – Run1 (50 products)	368	1338	837,10	283,59
MQTT – Run2 (50 products)	365	1329	858,52	286,44
MQTT – Run3 (50 products)	405	1349	879,40	285,92
Websockets – Run1 (50 products)	45	1026	504,66	285,52
Websockets – Run2 (50 products)	28	1009	530,48	306,21
Websockets – Run3 (50 products)	43	1010	528,06	286,34

6 Conclusion

This work proposes a framework with an integration layer using asynchronous communication to connect a holarchy and physical devices. There were conducted tests using JADE (MAS environment), Websockets (persistent bi-directional communication), and MQTT (publish/subscribe communication) as examples of asynchronous protocols to validate this approach in a demonstration scenario where Agents ask for skills on Resources that communicate with the hardware. It was possible to use this approach with both protocols and the time difference between both on the laboratory tests in NOVA University facilities let us infer that the Websockets protocol is slightly faster. Also, the fact of removing the FIPA protocol represents a decrease in the complexity on the controller's side. In future work, more protocols should be tested such as OPC or Rest Services and a benchmark should be created to understand on which conditions a protocol should be used in place of another. It is also interesting to create a Hub-like system on the integration layer that could accept requests from several asynchronous protocols and transfer them into the hardware-capable protocols. Arduinos seems to be a cost-effective platform to deploy these solutions because of the cost, flexibility, and the easiness to program them but further work should also include other microcontrollers and microcomputers such as Raspberry Pi or Zynq boards.

Acknowledgment. We acknowledge the Portuguese FCT program UIDB/00066/2020 for providing partial financial support for this work.

References

[1] A. Gilchrist, "Industry 4.0. Opportunities and Challenges of the New Industrial Revolution for Developing Countries and Economies in Transition," *2030 Agenda Sustain. Dev. Goals*, 2016, doi: 10.1007/978-1-4842-2047-4.

[2] G. Li, Y. Hou, and A. Wu, "Fourth Industrial Revolution: technological drivers, impacts and coping methods," *Chinese Geogr. Sci.*, vol. 27, no. 4, pp. 626–637, 2017, doi: 10.1007/s11769-017-0890-x.

[3] V. Mařík and D. McFarlane, "Industrial adoption of agent-based technologies," *IEEE Intelligent Systems*, vol. 20, no. 1. pp. 27–35, 2005, doi: 10.1109/MIS.2005.11.

[4] M. Jirgl, Z. Bradac, and P. Fiedler, "Human-in-the-Loop Issue in Context of the Cyber-Physical Systems," *IFAC-PapersOnLine*, vol. 51, no. 6, pp. 225–230, 2018, doi: 10.1016/j.ifacol.2018.07.158.

[5] J. Sztipanovits, T. Bapty, X. Koutsoukos, Z. Lattmann, S. Neema, and E. Jackson, "Model and Tool Integration Platforms for Cyber-Physical System Design," *Proc. IEEE*, vol. 106, no. 9, pp. 1501–1526, 2018, doi: 10.1109/JPROC.2018.2838530.

[6] D. Zeng, L. Gu, and H. Yao, "Towards energy efficient service composition in green energy powered Cyber–Physical Fog Systems," *Futur. Gener. Comput. Syst.*, vol. 105, pp. 757–765, 2020, doi: 10.1016/j.future.2018.01.060.

[7] L. Ribeiro, "Cyber-physical production systems' design challenges," *IEEE Int. Symp. Ind. Electron.*, pp. 1189–1194, 2017, doi: 10.1109/ISIE.2017.8001414.

[8] L. Monostori, "Cyber-physical production systems: Roots, expectations and R&D challenges," *Procedia CIRP*, vol. 17. pp. 9–13, 2014, doi: 10.1016/j.procir.2014.03.115.

[9] P. Hehenberger, B. Vogel-Heuser, D. Bradley, B. Eynard, T. Tomiyama, and S. Achiche, "Design, modelling, simulation and integration of cyber physical systems: Methods and applications," *Comput. Ind.*, vol. 82, pp. 273–289, 2016, doi: 10.1016/j.compind.2016.05.006.

[10] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of Cloud computing and Internet of Things: A survey," *Futur. Gener. Comput. Syst.*, vol. 56, pp. 684–700, 2016, doi: 10.1016/j.future.2015.09.021.

[11] W. E. Lucke D., Constantinescu C., "Smart Factory - A Step towards the Next Generation

of Manufacturing," in *Manufacturing Systems and Technologies for the New Frontier*, London: Springer, London, 2008, pp. 115–118.

[12] L. Tamas and M. Murar, "Smart CPS: vertical integration overview and user story with a cobot," *Int. J. Comput. Integr. Manuf.*, vol. 32, no. 4–5, pp. 504–521, 2019, doi: 10.1080/0951192X.2018.1535196.

[13] M. Onori, J. Barata, and R. Frei, "Evolvable assembly systems basic principles," *IFIP Int. Fed. Inf. Process.*, vol. 220, pp. 317–328, 2006, doi: 10.1007/978-0-387-36594-7_34.

[14] H. Van Brüssel, L. Bongaerts, J. Wyns, P. Valckenaers, and T. Van Ginderachter, "A conceptual framework for kolonic manufacturing: Identification of manufacturing holons," *J. Manuf. Syst.*, vol. 18, no. 1, pp. 35–52, 1999, doi: 10.1016/S0278-6125(99)80011-9.

[15] P. Leitão, "Agent-based distributed manufacturing control: A state-of-the-art survey," *Eng. Appl. Artif. Intell.*, vol. 22, no. 7, pp. 979–991, 2009, doi: 10.1016/j.engappai.2008.09.005.

[16] M. E. Leusin, M. Kück, E. M. Frazzon, M. U. Maldonado, and M. Freitag, "Potential of a Multi-Agent System Approach for Production Control in Smart Factories," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1459–1464, 2018, doi: 10.1016/j.ifacol.2018.08.309.

[17] G. Andreadis, P. Klazoglou, K. Niotaki, and K. D. Bouzakis, "Classification and review of multi-agents systems in the manufacturing section," *Procedia Eng.*, vol. 69, pp. 282–290, 2014, doi: 10.1016/j.proeng.2014.02.233.

[18] J. Dale, "FIPA Agent Communication Language Specification," 2002. http://www.fipa.org/repository/aclspecs.html (accessed Jan. 05, 2022).

[19] S. C. Feng, "Preliminary design and manufacturing planning integration using web-based intelligent agents," *J. Intell. Manuf.*, vol. 16, no. 4–5, pp. 423–437, 2005, doi: 10.1007/s10845-005-1655-4.

[20] G. Rzevski, "A framework for designing intelligent manufacturing systems," *Comput. Ind.*, vol. 34, no. 2, pp. 211–219, 1997, doi: 10.1016/s0166-3615(97)00056-0.

[21] I. Kovalenko, K. Barton, and D. Tilbury, "Design and implementation of an intelligent product agent architecture in manufacturing systems," *IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA*, pp. 1–8, 2017, doi: 10.1109/ETFA.2017.8247652.

[22] I. Kovalenko, D. Ryashentseva, B. Vogel-Heuser, D. Tilbury, and K. Barton, "Dynamic Resource Task Negotiation to Enable Product Agent Exploration in Multi-Agent Manufacturing Systems," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2854–2861, 2019, doi: 10.1109/LRA.2019.2921947.

[23] W. Lepuschitz, A. Zoitl, M. Vallée, and M. Merdan, "Toward self-reconfiguration of manufacturing systems using automation agents," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 41, no. 1, pp. 52–69, 2011, doi: 10.1109/TSMCC.2010.2059012.

[24] L. Ribeiro and M. Hochwallner, "On the design complexity of cyberphysical production systems," *Complexity*, vol. 2018, 2018, doi: 10.1155/2018/4632195.

[25] K. Ding, F. T. S. Chan, X. Zhang, G. Zhou, and F. Zhang, "Defining a Digital Twin-based Cyber-Physical Production System for autonomous manufacturing in smart shop floors," *Int. J. Prod. Res.*, vol. 57, no. 20, pp. 6315–6334, 2019, doi: 10.1080/00207543.2019.1566661.

[26] A. Hyre, G. Harris, J. Osho, M. Pantelidakis, K. Mykoniatis, and J. Liu, "Digital Twins: Representation, Replication, Reality, and Relational (4Rs)," *Manuf. Lett.*, vol. 31, pp. 20–23, 2021, doi: 10.1016/j.mfglet.2021.12.004.

[27] K. Samir, A. Maffei, and M. A. Onori, "Real-Time asset tracking; a starting point for digital twin implementation in manufacturing," *Procedia CIRP*, vol. 81, pp. 719–723, 2019, doi: 10.1016/j.procir.2019.03.182.

[28] L. Ribeiro, J. Barata, M. Onori, C. Hanisch, J. Hoos, and R. Rosa, “Self-organization in automation - The IDEAS pre-demonstrator,” *IECON Proc. (Industrial Electron. Conf.*, no. November, pp. 2752–2757, 2011, doi: 10.1109/IECON.2011.6119747.