DOI: 10.48011/asba.v2i1.1743

Proposal of an Architecture based on RAMI 4.0 Layers for Flexible Manufacturing in Industry 4.0

José Z. Neto*, Joel Ravelli Jr.*, Eduardo P. Godoy*

**Universidade Estadual Paulista (Unesp), Sorocaba, São Paulo, Brasil*
(jose.zapata@unesp.br, joelravelli@gmail.com, eduardo.godoy@unesp.br)

Abstract: The Industry 4.0 (I4.0) together with the Industrial Internet of Things (IIoT) enable business productivity to be improved through rapid changes in production scope in an increasingly volatile market. This technology innovation is perceived by integrating manufacturing systems, managing business rules, and decentralizing computing resources, enabling rapid changes in production systems. The Reference Architecture Model for Industry 4.0 (RAMI 4.0) is a three-dimensional layer model to support I4.0 applications. One of the major challenges for adopting RAMI 4.0 is the development of solutions that support the functionality of each layer and the necessary interactions between the elements of each layer. This paper focuses on the proposal of architecture for flexible manufacturing in I4.0 using all the Information Technology (IT) Layers of the RAMI 4.0. In order to enable a standardized and interoperable communication, the architecture used the OPC-UA protocol to connect the low layers elements in the factory perspective and REST APIs to connect the high layers in the business perspective. The integration architecture creates an online interface to provide the client the ability to enter, view, and even modify an order based on their needs and priorities, enabling the industry to implement rapid changes to adapt to the marketplace.

Keywords: Industry 4.0, IIoT, IoT, RAMI 4.0, OPC-UA, Life Cycle Value Stream.

1. INTRODUCTION

Based on the technological advances of the last decades, Information Technology (IT) has been shaping society as we know it. Several technologies have emerged in recent decades and changed the way we think about productivity and redefining business models. The rise of the Internet of Things (IoT) in which lots of devices can interact together to produce valuable data (Hassanzadeh et al., 2015) and the information interpreted to generate valuable insights. The IoT, combined with Industry applications, originate the Industrial IoT (IIoT).

Combined with the IIoT, the Cyber-Physical Systems have transformed the way we deal with the production systems. In other words, a true revolution was noticed and called the Fourth Industrial revolution (Lee et al. 2015). Given the increasing acceptance of Industry 4.0 (I4.0) initiative all around the globe, a Reference Architecture Model for Industry 4.0 (RAMI 4.0) was developed based on vertical and horizontal integration and end-to-end engineering. Nevertheless, RAMI 4.0 initiative requires efforts in different aspects to reach the level of practical implementation (Pisching et al., 2018).

There are challenges in regards to facing the great potential of I4.0 and its adoption such as data security, communication reliability, interoperability, and scalability (Eruvankai et al., 2017). According to Melo and Godoy (2018), the OPC UA protocol (Open Platform Communications Unified Architecture) can help in overcoming these challenges. OPC UA presents a secure and fully scalable architecture, based on the concept of SOA (Service Oriented Architecture). As discussed by Steinegger et al. (2017), a solution assembled on SOA generally has a standards-based architecture for the creation of an IT infrastructure, focusing on simplifying the relationships between different systems, improving its operation, and facilitating the incorporation of new elements. Therefore, if there are changes in business needs, these factors allow the company to respond to them quickly.

In addition to meeting the requirements of I4.0, the OPC UA protocol was originally created to address limitations found in the specifications of its predecessor, the OPC Classic (DA). It is possible to consider that the OPC UA unifies all specifications of the Classic OPC and makes them state of the art using SOA (Lehnhoff et al., 2011).

Few architecture models can represent a product lifecycle through the factory floor using all layers of RAMI 4.0. Mourtzis et al. (2018) shows how a set of tools designed to integrate customer in the product design and manufacturing that consider Industry 4.0 technologies may be classified using RAMI 4.0 architecture. Melo and Godoy (2019) focused on the development of an open source control device for I4.0 based on RAMI 4.0 and OPC UA protocol. The control device provided functionalities covering the first three layers of RAMI 4.0, *Asset, Integration and Communication layers*.

This paper is focuses on creating an architecture model for flexible manufacturing in I4.0 applications using all the six IT Layers of RAMI 4.0. Therefore, the proposal is the

standardization of the manufacturing process in industrial equipment through the identification of requirements, and the implementation of functional components for integration with the RAMI 4.0 architecture. It covers from the Business perspective of I4.0 to *Asset layer*, on factory floor, promoting a flexible and scalable architecture.

The following text organization: in Section 2, the Reference Architecture for Industry 4.0 is presented in details. Specifics of the architecture proposed for this paper are shown in Section 3. In Section 4, a discussion and critiques about the architecture implementation is presented. Finally, Section 5 presents the conclusions and future works.

2. REFERENCE ARCHITECTURE MODEL FOR INDUSTRY 4.0 – RAMI 4.0

The three-dimensional structural axes of RAMI 4.0 can be divided into the following hierarchies: Life Cycle & Value Stream, Hierarchy Levels and IT Layers. According to Dorst et al. (2016), I4.0 offers huge potential for improvement over the life cycle of products to visualize and standardize relationships with machinery.

The Life Cycle & Value Stream axis, as shown in Fig. 1, was based on the IEC 62890 standard that deals with the management of the lifecycle for systems and products used in industrial processes, measurements, control, and automation. Furthermore, a distinction is made between "types" and "instances". A "type" becomes an "instance" when design and prototyping have been completed and the actual product is being manufactured (Gotze et al., 2018).

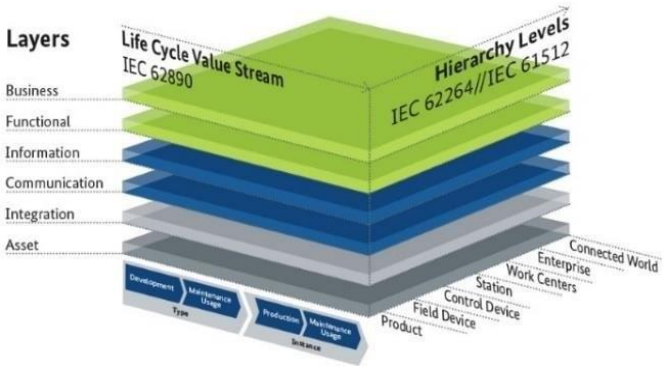

Fig. 1 Reference Architecture Model for Industry 4.0

The axis of Hierarchical Levels was based on the IEC 62264/61512 standard and defines the model of interconnection of all elements of the productive system, including information, people and machines. This axis is related to the enterprise-control system integration.

The Layers axis of the RAMI 4.0 is related to the IT layers such as in the OSI model. Each layer has specific responsibilities that can be resumed as follows:

- *Business:* related to abstract business models (rules) and process logic, as well as ensuring the integrity of functions along the value chain (Wang et al. 2017). This layer ensures the integrity of functions in the value stream, enables mapping business models and the resulting of the overall process. It contains legal and regulatory framework conditions, enables modeling of the rules which the process has to follow. This layer also creates a link among different business processes.

- *Functional:* enables the formal description of parts and creates a platform for horizontal mixing of various functions. It contains runtime and modeling environment for services for support of business processes and a runtime environment for applications and technical functionality. Rules and decision-making logic are generated in the *Functional Layer*. Some use cases can be executed in lower layers as well. But remote access and horizontal integration can take place within the *Functional Layer* only because of the necessity of data integrity (Zezulka et al. 2016).

- *Information:* provides run time for preprocessing of events and execution of event-related rules. It enables formal description of the business rules (Zezulka et al. 2016). For this, data is checked for its integrity, summarized in a new, structured and organized information model, and made available to upper layers such as Functional (Wang et al. 2017). It also receives events and transforms them to match the data which are available for the upper layers (Zezulka et al. 2016).

- *Communication:* This layer provides standardization of communication by means of uniform data format and content (Zezulka et al. 2016) and also provides services to control the *Integration layer* (Wang et al. 2017). The RAMI 4.0 recommends the usage of the OPC UA protocol for this layer.

- *Integration:* provides information related to technical assets such as hardware, software, and device network communication. This layer contains all elements associated with IT, including HMI generating events based on information acquired and performing final process control (Wang et al. 2017). It is also responsible for managing the events from all assets, such as RFID readers, sensors, and actuators, as well as the integration of user inputs (Zezulka et al. 2016).

- *Asset:* represents the physical reality, containing objects such as machines, sensors, actuators and documentation. This layer identifies products and parts in relation to processing requirements for machines and workstations (Wang et al. 2017). Humans are a part of the *Asset Layer*, by being connected by the *Integration Layer*. The handling of the assets to the *Integration Layer* is done by identification technologies such as RFID tags, beacons and barcodes.

Within the elements of these three axes, all crucial aspects of I4.0 can be addressed, allowing any application such as a

workstation logic control to be mapped according to the model. Highly flexible I4.0 concepts can thus be described and implemented based on RAMI 4.0. The model allows for step-by-step migration from the present into the world of I4.0 (Lydon. 2019).

Chapter 3 proposes and discusses an architecture implementation of six Life Cycle Value Stream layers of RAMI 4.0, using OPC UA to bring to life the *Information Layer*, and REST APIs to bring Business Layer, decision making to a factory perspective.

3. ARCHITECTURE PROPOSAL

This paper proposes a standardized and fully integrated architecture based in Life Cycle and Value Stream Layers of RAMI 4.0 to deliver a modular, secure, and exteriorized control perspective for flexible manufacturing in Industry 4.0, using OPC UA Server, OPC UA Orchestrator and RESTful APIs. To guarantee compatibility, security and modularity in the factory floor perspective, the OPC UA Orchestrator coordinates information using microservices oriented architecture. Alongside, the Restful APIs exteriorize the factory data, allowing the *Business Layer* to communicate with the factory for fast and escalated changes in production orders while they are being produced. The proposed architecture can be seen in Fig. 2.

In the *Asset Layer*, the item, goods, raw material or product is identified and becomes part of the production environment, by having a way to interact with the factory (by the Integration layer). The unique identification of the object is done by the Integration layer. The Integration layer allows the system to control the product manufacturing by means of the IEC 61131 standard. Also, it enables managing the product lifecycle, guiding or changing production routes according to the orders and providing guidance to the product along the manufacturing process.

The *Communication Layer* was designed to be fully based in the OPC UA protocol. According to Mathias et al. (2020), many use cases in manufacturing domains demonstrate the vast interoperability and cross-platform connectivity strengths of OPC UA. The *Communication Layer* is responsible to translate and standardize the upper layer business rules to the product manufacturing procedure, guaranteeing that the product roadmap is clear and error proof. The architecture was designed to be as generic and simple as possible. According to Ausberger et al. (2019), the OPC UA gateway must require minimal configuration and obtain information from data source automatically. In order to guarantee security, this layer is end-end encrypted by certificate.

The *Information Layer* was designed to the core where all the events come together. In this layer, the information is modelled to guarantee the business rules are being applied, and to guarantee that the functionality of the production factory is being followed. This layer must be plug and play, and that means to be flexible to quickly adapt to new products and items to be introduced in the manufacturing process. Additionally, this layer must also be scalable for different kinds of companies who want to start the production, it's just based on database information, not client's production system, which is differential but highly related to the I4.0 requirement of great customization.

The data modeling was designed to allow the mapping of any Industry from the physical to the virtual environment based on the JSON format. Generally, the assets in a factory have different functions, such as a stepper motor, conveyors, sensors, RFID tags, and etcetera. This work, proposed a data modelling of different assets with the same information model. The goal is to be agnostic, so the data model allows us to model different specifications of assets and store your data in a standardized way within the factory ensures the delivery and security of data in real-time to the lower layers of RAMI 4.0 and stores information in the database for upper layers as a high availability and security solution.

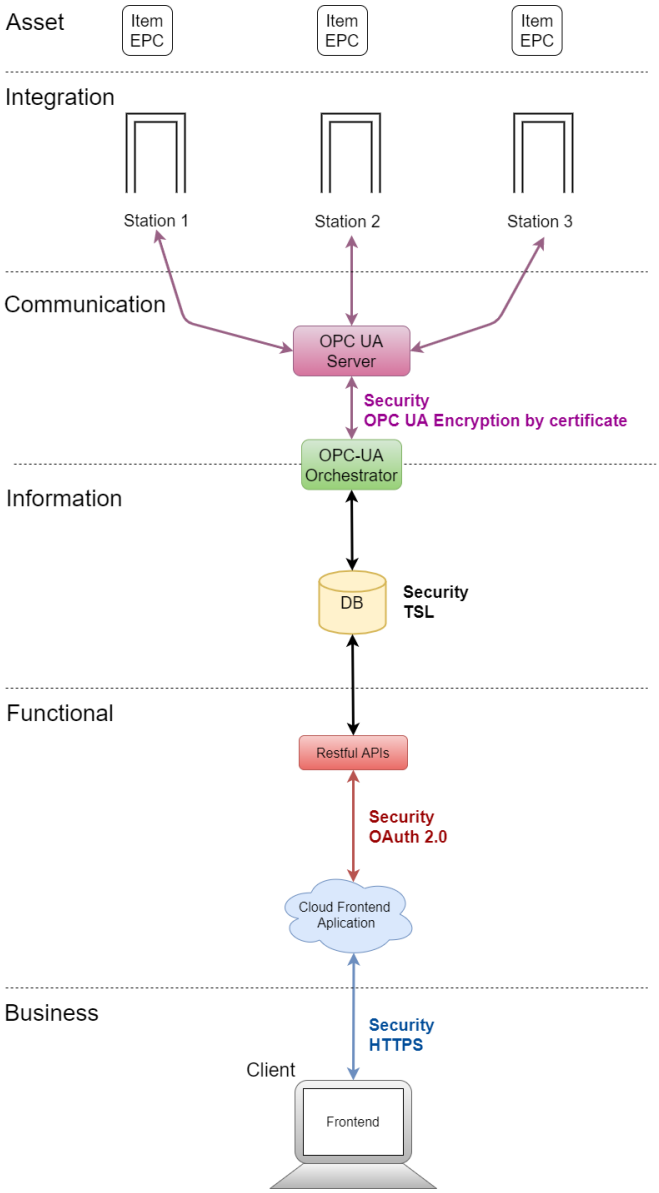

Fig. 2 System Architecture based on the RAMI 4.0 Layers.

The *Functional Layer* supports the business rules and the manufacturing process. The rules and decision-making logic are absorbed and processed in the *Functional Layer*,

consuming real-time data from the *Information Layer*, allowing quick view of order status and quick changes in production orders. The *Functional Layer* provides all data integrity that is inserted in the factory from the business side. In order to enable it to be quickly scalable and secure, the implementation chosen was Restful API with OAuth 2.0 protocol authorization. The choice of APIs and OAuth 2.0 focuses on client developer simplicity while providing specific authorization flows for web applications, desktop applications and mobile phones.

The *Business Layer* was designed to bring all important information about status of the production order, about production timeline and real-time status for every order or item for a respective client. This data allows a quick view and decision-making process. The client can orchestrate changes in the production order from IT system or laptop and the *Functional Layer* will figure out if that is possible or not and respond fast.

For example, a company we call shirt-retail has ordered ten thousand black t-shirt and ten thousand of white t-shirts from the factory we call shirt-maker, due to a change in market, shirt-retail wants to change the order to fifteen thousand black t-shirts and fifty thousand white t-shirts. Using the IT system, the shirt retail executive can ask using a front-end, which calls the API to change the order, passing the parameters for the order update. The API will look inside the production order and realize the update is possible based on real time information and responds if it is possible or not. It is invisible for shirt-maker if a production order has changed because it does not affect the process, it just changed to a new one. The layer ensures the integrity of functions in the value stream for the business, enabling a map to any business models and the results of the overall process for a single factory.

4. ARCHITECTURE IMPLEMENTATION

The *Asset Layer* of the architecture was based on products or parts identified using RFID, but could work with barcode, QR Code, Bluetooth beacon, and any other object identification technology. The system consists of one RFID tag stored with an Electronic Product Code (EPC) for unique identifiers assigned to physical objects. The RFID reader is assigned on each station to identify nearby tags.

The EPC is a binary code, usually 96 bits in size, which is recorded in the TAG RFID memory. This code can contain any information, such as a serial number, but there are market standards and the most accepted is from the global organization GS1 (Repec et al., 2017).

The OPC UA uses a client/server or publish/subscribe communication following the design paradigm of service-oriented architecture (SOA). Each station communicates directly with an endpoint. The pub/sub model decouples the publisher that sends a message from the subscriber (or subscribers) that receives the messages.

The OPC UA Orchestrator is an important part of the architecture in which all the factory and production rules are set. In the presented paper, the rules consumed from the Orchestrator are used to define the product roadmap inside the manufacturing production line.

For example, a product identified by EPC ABC1234 must go through station 1, 2, 3, 4, 7, 14 and 22. The OPC UA Orchestrator uses Database knowledge to identify the product route and guide it (*Integration Layer*) inside the production line using RFID and actuators. In this way, when the product is done in station 1, the Orchestrator will guide to station 2, then 3, until station 22, the final point. It is all based on the product Part Number, the Orchestration will respond based on the known route. The sequence diagram is shown on Fig. 3.

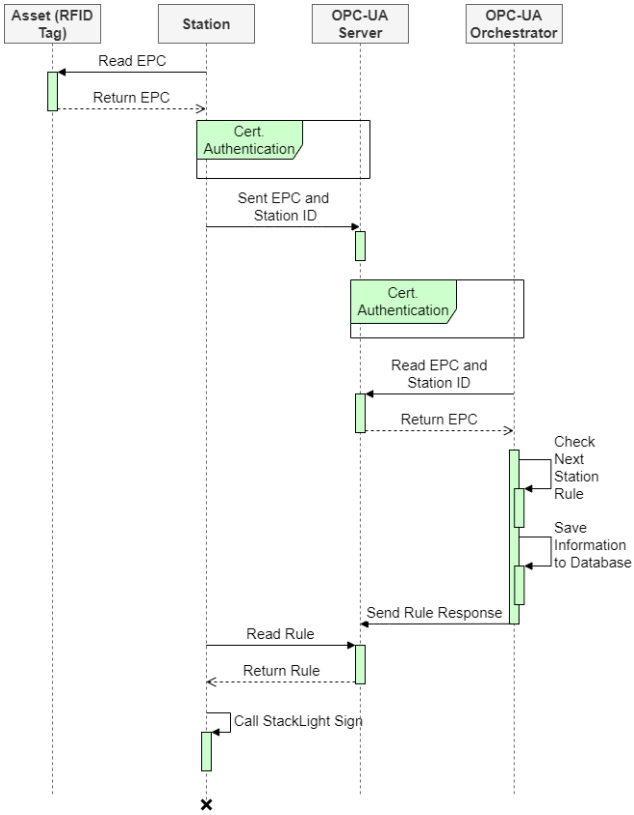

Fig. 3. Architecture Integration within the *Asset, Integration and Communication Layers.*

The system has three possible answers when the item reaches a station:

- If the item should not be in this station, it returns '-1' and this is interpreted by a route error.
- If the item is in the correct route, a number corresponding to the next station is returned, so the system can make decisions.
- If the item is in the correct route, but this is its last station, it returns '0'.

The *Information Layer* is where all information comes together. From the factory perspective, all inserted EPC from the RFID will be stored in itemRoute and verified by the Orchestrator using the KnowledgeRoute table. The ongoing status of products will affect the table Item and,

consecutively, the table Order. The table Client was designed to make the system more plug and play, which does not rely on the client, it focuses at the products roadmap. The Database can be seen in Fig. 4.

To connect the factory *Information Layer* to the Business side, the *Functional Layer* is responsible for creating a path between the factory database and the business requests, as can be seen in System Architecture in Fig. 5. To achieve that, several Restful APIs were implemented to provide the client all the factory services remotely.

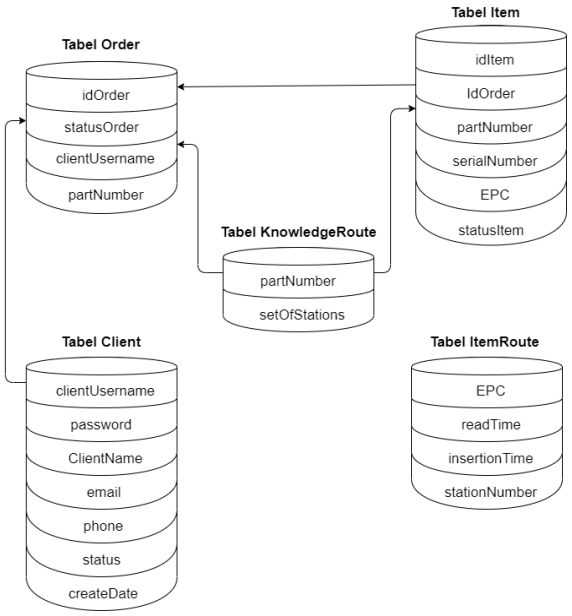

Fig. 4 Database: *Information Layer* Model.

The available APIs allow to:

- Post Order: this API is used to request an order. The Order is associated with a specific client by the clientUsername and a valid PartNumber, associated with the table ItemRoute. That means, that only a client associated in the table Client can insert an order in the Database and only a PartNumber registered in the ItemRoute can be produced. After the order is created, the table Item is populated.

- Get Order: this API is used to consult orders. If no value is passed, the API returns all orders in the database from a specific client. If an idOrder is passed, the API returns the order and all items associated with the order. If the Order does not exist, it returns: {"error":"not found"}

- Get Route: this API is used to consult a Product Route given a Product Part Number. This may be used by the client to understand a route and consult a status of Items using GET Orders and implement changes in the order course. If the Part Number does not exist it returns: {"error":"not found"}

- Put Order: this API is used to update a current order. If the KnowledgeRoute for that Part Number and the table Item perceived that a change in the order is achievable, the systems will update the PartNumber for that item, allowing it to follow a different path and become a different product. This allows the manufacturer to be more flexible and adaptive to market changes.

- Delete Order: The API is used to delete a current order if the systems have not started the fabrication process. This information can be consulted in the table Order in Status.

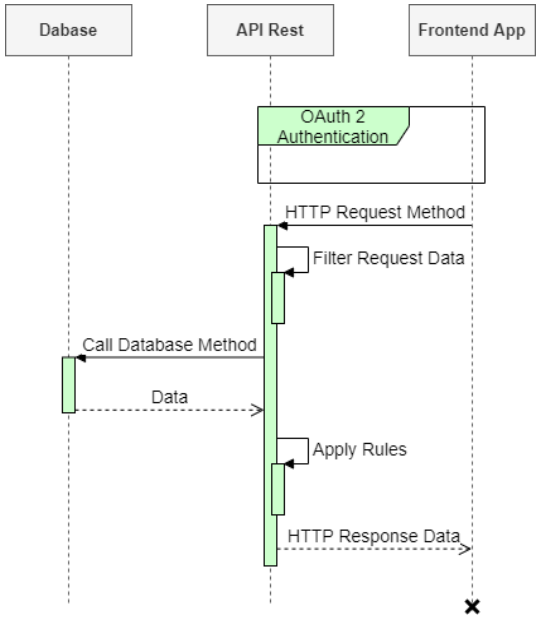

Fig. 5 System Architecture: *Functional Layer*.

The *Business Layer* was thought to be adaptive, so the client can create a front-end to use the APIs in the best way to adapt to his business needs. The client can orchestrate the APIs in different ways by application or by a middleware, which combines different APIs at the same time.

All the data is invisible to the factory-side, and fully validated in the *Functional Layer* to guarantee the business real-time information and quick responses to orders. This allows an executive to implement quick changes in a production order based on market needs like insert, update, and delete orders quickly, without having to contact the fabrication unit.

Nowadays, the factory perspective requires constant changes to adapt to new products, new brands and new machines. We have proposed micro services based on architecture of RAMI 4.0 Layers allowing them to be fabricated only by doing a setup, so the OPC UA can guide the product alongside the factory. Another benefit is being able to demand new orders and modify the ones already being produced, allowing quick response in an increasingly volatile market, all being done using RESTful Web Services.

This proposal, from the Information to *Business Layer* uses Service Oriented Architecture and has a standards-based architecture for the creation of an IT infrastructure, aiming to simplify the relationships between different systems,

improving its operation and facilitating the incorporation of new elements. Therefore, if there are changes in business needs, these factors allow the company to respond to them quickly.

A strongest point of this architecture is the adoption of a data model based on JSON through the entire architecture, since *Integration* to B*usiness Layer* of RAMI 4.0. Normally, equipment from vendors has a different data format to communicate between the Asset and Integration layers. In consequence, this architecture can collect those pieces of information from any devices and normalize it for the upper layers transforming into JSON data format on the *Integration Layer*.

Another important point featured is the security connections implemented on layers from *Integration* to *Business Layer*. Which means in a good way, that all interactions between layers are encrypted with credentials. However, the *Asset Layer* could need to integrate devices with different specifications. Thus, it could lead to an extra effort to make legacy or custom devices compatible with the Integration layer.

The OPC UA server presented is the key software component in the *Communication Layer* and for the proposed architecture as well. Its features are essentials to collect and send data to lower layers and delivery information to the upper layers. Therefore, in this way, the server must have to be inside the factory to attend the requirement of real-time communication with devices located on the *Asset* and *Integration Layers*. That scenario requires a good IT infrastructure inside the factory to provide high availability and load balance for all connections need.

The proposed architecture for *Functional* and *Business Layers* are ready to be deployed on any commercial cloud system nowadays, such as AWS, Azure or IBM. It helps to decentralize the IT infrastructure, gain scalability, and save financial resources.

5. CONCLUSION

This paper presented and described the development of architecture for flexible manufacturing in Industry 4.0 using all the Information Technology layers of RAMI 4.0. The main components of the architecture are the RFID with EPC codes on the *Asset Layer*, the OPC UA communication and Orchestrator on the *Communication Layer*, a database-oriented information model on the Information layer and the use of REST APIs on the *Functional Layer*.

The Asset and *Integration Layer* orchestrated by the *Communication Layer* brings all information from the factory to the database allowing production decisions from the server to the factory floor. This process improves system models over the value chain and optimizes system-wide operations. The *Information Layer* includes a data management mechanism for virtual representation and stable integration of different data in a structured way.

The *Functional Layer* describes most of the application related functionalities. It implements domain-specific applications, the integration of control strategies of field devices to provide system operation planning and asset and service management. In terms of information exchange, the *Functional Layer* provides the integration for the control strategies of the entire system operation down to the field layer receives data flows obtained during the manufacturing process and integrates this information with the decision-making rules from the B*usiness Layer* perspective.

The *Business Layer* describes most of the business-related functionalities related to the business rules of the corporation. It also offers the system operator decision making support, provides system operation planning and asset and service management. This layer also provides interfaces for external users to access them, which can be achieved through APIs and human interface applications such as middleware or fronted. In terms of information exchange, the *Business Layer* provides the foundation for the control strategies of the entire system operation down to the field layer and receives data flows obtained during the manufacturing process.

ACKNOWLEDGMENT

Research supported by grant 2018/19984-4, São Paulo Research Foundation (FAPESP).

REFERENCES

Ausberger, T., and Stetina, M. (2019). General methodology for building of OPC UA gateways. 16th IFAC Conference on Programmable Devices and Embedded Systems PDES 2019: High Tatras, Slovakia, 29–31 October 2019. Pages 317-322. https://doi.org/10.1016/j.ifacol.2019.12.680

Dorst, W., Glohr, C., Han, T., Knafla, F., Loewen, U., Rosen, R., Schiemann, T., Vollmar, F., and Winterhalter, C. (2016). Implementation Strategy Industrie 4.0: Report on the results of the Industrie 4.0 Platform. URL: https://www.bitkom.org/Bitkom/Publikationen/ImplementationStrategy-Industrie-40Report-on-the-results-of-the-Industrie-40- Platform.html [Accessed 06 Jun 2020]

Eruvankai, S; Muthukrishan, M and Mysore, A, K (2017). Accelerating IIoT adoption with OPC UA. In: Internetworking Indonesia Journal, v.9, n1, p.3-8.

Gotze, J. (2016). Reference architecture for industry 4.0. QualiwareCenter of Excellence, 2016. Available at: https://coe.qualiware.com/reference-architectures-for-industry-4-0 [Accessed 17 Dez 2018].

Hassanzadeh, A., Modi, S. and Mulchandani, S. (2015). Towards Effective Security Control Assignment in the Industrial Internet of Things 2nd IEEE World Forum on Internet of Things (WF-IoT), 10.1109/WF-IoT.2015.7389155, 6 pages.

Lee, I. and Lee, K. (2015). The internet of things (IoT): Applications, investments, and challenges for enterprises Business Horizons, 58 (2015), pp. 431-440

Lehnhoff, S., Mahnke, W and Rohjans, S (2011). IEC 61850 based OPC UA communication - the future of smart grid. In 17th Power Systems Computation Conference, Stockholm Sweden, p.1-9.

Lydon, B. (2019). Automation IT: RAMI 4.0 reference architectural model for industrie 4.0. ISA", Isa.org, 2019. URL: https://www.isa.org/intech/20190405/. [Accessed: 06- Jun- 2020].

Mathias, S., Schmied, S. and Grossmann, D. (2020). An investigation on database connections in OPC UA applications. The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) Pages 602-609. https://doi.org/10.1016/j.procs.2020.03.132

Melo, P. and Godoy, E. (2018). Análise comparativa de implementações de software de código aberto do OPC UA. 10.20906/CPS/CBA2018-0292.

Melo, P. and Godoy, E. (2019) Controller interface for industry 4.0 based on RAMI 4.0 and OPC UA," 2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0&IoT), Naples, Italy, 2019, pp. 229-234, doi: 10.1109/METROI4.2019.8792837.

Mourtzis, D., Gargallisa, A., and Zogopoulosa, V. (2019). Modelling of customer-oriented applications in product lifecycle using RAMI 4.0. 7th International conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV2018). Volume 28, 2019, Pages 31-36. https://doi.org/10.1016/j.promfg.2018.12.006

Pisching, A., Pessoa, O., Junqueira, F., Santos, D.J., Miyagi, E., (2018). An architecture based on RAMI 4.0 to discover equipment to process operations required by products. Computers & Industrial Engineering, Volume 125, November 2018, Pages 574-591 doi: https://doi.org/10.1016/j.cie.2017.12.029

Repec, C., A., Traub, K., Corthell, J., El-Leithy, H., Fisher, R., Graff, H., Mullen, D., Oosterhof, R., Tröger, R. (2017). EPC Tag Data Standard. URL: https://www.gs1.org/sites/default/files/docs/epc/GS1_EPC_TDS_i1_10.pdf [Accessed: 20- Jun- 2020].

Steinegger, Roland & Giessler, Pascal & Hippchen, Benjamin & Abeck, Sebastian. (2017). Overview of a Domain-Driven Design Approach to Build Microservice-Based Applications.

Wang, Y.; Towara, T.; Anderl, R. (2017). Topological approach for mapping technologies in reference architectural model industrie 4.0 (RAMI 4.0). Proceedings of the World Congress on Engineering and Computer Science 2017 Vol II WCECS 2017, October 25-27, 2017, San Francisco, USA.

Zezulka, F., Marcon, P., Vesely, I. and Sajdl, O. (2016) Industry 4.0 - An Introduction in the phenomenon. 14th IFAC Conference on Programmable Devices and Embedded Systems PDES 2016IFAC-PapersOnLine, 49(25), pp. 8-12.