

Flexibility Enhancements in Digital Manufacturing by means of Ontological Data Modeling

Richard Otrebski^a*, Johannes Rauer^a, Corinna Engelhardt-Nowitzki^a, Natalie Kryvinska^a, Mohamed Aburaia^a, Dominik Pospisil^a

^aUAS Technikum Wien – Department Industrial Engineering, Vienna, Austria, 1200 Vienna

Abstract

Cyberphysical production systems are important parts of today's manufacturing process. The ever-growing need of highly optimized, i.e. at the same time flexible and efficient systems, requires the use of not only appropriate machines, but as well a communication framework and data model that is manufacturer independent and scalable. This paper proposes a communication-framework based on OPC UA that employs an agent-based architecture. The proposed system has been implemented and tested in the Digital Factory of the UAS Technikum Wien. It shows promising behavior within distributed manufacturing systems.

Keywords: communication modeling, data modeling, digital manufacturing, OPC UA, CPPS

1. Introduction

The development of efficient and at the same time adaptive production systems is an important concern in industrial manufacturing [1]. The intention is to operate highperformance, robust, flexible and adaptable production systems that are at the same time user-friendly, environmentally compatible and cost-effective. From a business perspective the best possible combination of low cycle times and high throughput and utilization has to be achieved. A critical success factor for an innovative manufacturing environment in this sense is the digitization and interconnectedness of systems, machines, tools, work pieces, products and product components: A flexible production requires a fast and dense information flow in order to adapt the system. The communication of these system elements - cyberphysical systems (CPS) - is based on a large number of sensors that are exchanging messages via the internet, and the subsequent use of the retrieved sensor data for process adjustments [2]. However, the interoperability of the mentioned systems and components requires standardized protocols and interfaces and has to be based on a consistent data model.

This article uses the example of a 'Digital Factory' (DF) [3] [4] of the University of Applied Sciences Technikum Wien (UAS Technikum Wien) to illustrate how theory-driven, abstract data structuring of cyberphysical production systems (CPPS) can support industrial practice, especially process flexibility. The technical equipment is extremely heterogeneous to demonstrate typical industry 4.0 settings. For example, an assembly station

the company SMC is loaded by a Wittmann portal robot and subsequently delivers finished work pieces to a mobile robot. This requires the interaction of three different robot systems and corresponding system architectures. Another use case is intelligent routing, in which either the transport robot, the product or the central factory planning software decides between the use of two alternative milling robots. Again, communication between heterogeneous systems is needed. Accordingly, the objective of the present paper is to introduce an ontology-based integrated communication system based on the open platform communications unified architecture (OPC UA) protocol. OPC UA has been chosen as it is a versatile industrial protocol with applications in different levels of the automation pyramid [5]. In addition to continuous and flexible data exchange, the system incorporates new production devices as soon as they are connected and integrates those systems into the manufacturing process. This is accomplished with an agent-based approach [6] and the facilitation of OPC UA features such as the discovery service set.

consisting of traditional automation components provided by

Our contribution is the combination of different existing approaches into a combined framework for the creation of easily extensible systems. The Digital Factory corresponds to realistic company scenarios to the extent that not only seven different manufacturers, but also different robot generations are used for production. This enables a multitude of technical concepts and business models regarding production strategies and process planning implications (e.g., both, make-to-stock and make-to-order models can be represented). The technical DF environment consists of fourteen industrial robots and three autonomous mobile robots. The exemplary product to be

DOI: 10.5383/JUSPN.12.02.001

^{*} Corresponding author. Tel.: +43 1 333 40 77 7452

E-mail: <u>otrebski@technikum-wien.at</u>

 $[\]ensuremath{\mathbb{C}}$ 2020 International Association for Sharing Knowledge and Sustainability.

produced is an axle bearing collar. The workflow is deliberately designed to be logistically sub-optimal in order to represent typical material flow restrictions caused by e.g., unfavorable building architectures or terrain profiles, which are the norm in companies. Each robot performs defined production tasks and can be adapted to different tasks by quickly changing tools. The autonomous transport vehicles use a variety of procedures (e.g. odometry, indoor GPS navigation) to perform their transport tasks efficiently and collision-free within the scope of the respective production strategy.

Without a central 'communication data model', which on the one hand organizes the barrier-free data exchange between different production devices and on the other hand ensures that new machines can be integrated quickly and seamlessly, the adaptivity of a factory is limited to operative topics such as the above-mentioned tool change. Similarly, flexibility in the integration of new products and product mix flexibility are limited or at least cause disproportionately high integration costs.

Accordingly, this article illustrates how the contradiction between flexibility and efficiency of a production process, which can often be observed in conventional production scenarios, can be significantly mitigated by systematic ontology-based modeling, using the example of the development of a suchlike data model in the DF of the UAS Technikum Wien.

The remainder of this paper is structured as follows: Section 2 provides introductory considerations with regards to ontological modeling in heterogeneous manufacturing settings. Next, the developed data model is introduced and explained indepth (Section 3). After that, Section 4 discusses the needed considerations regarding the reliability of the production system. Subsequently, Section 5 exemplarily clarifies, how the data model supports the application of technical solutions in order to improve the business logic and performance. We conclude with further considerations as to how systematic data representation, and in particular ontology-based data modeling, enable the objective of efficient and resource-conscious, and at the same time flexible production even in unstable market fields that require a high degree of adaptability (Section 6).

2. Definitions and related work

Modern production systems (e.g., robots or computer numerical control (CNC) machines, here referred to as 'agents') must be capable to communicate with different and physically distributed systems and subsystems. Manifold contextual circumstances are to be taken into account, e.g. the geographical system position, or necessary interfaces with contract manufacturers or suppliers. This requires a holistic overall architecture of a CPS [7] and should be taken into account when designing and implementing agents. Besides, the need for redundancies and fault tolerance of the production system needs to be incorporated [8]. [9] even implemented a smart factory where the designed ontology lets users generate a custom product and a reasoner derives the required process steps.

These agents can be either closed systems such as a CNC machine, or another (subordinated) orchestrator, that is

responsible for an entire subsystem (i.e., an agent). Since the orchestrator and the agent provide the same interface, it is much easier to communicate the data in programming. For cross-system collaboration, an important concept is to abstract generic 'skills', i.e. capabilities that a system can offer, to the manufacturing process, from the specific technical system properties. Accordingly, [1] depicts an architectural approach for an agent based CPPS. Newly installed technical components only have to inform the CPPS about their skills to be easily integrated into the production process. In addition to the skill information, data on operation status (e.g., manufacturing progress, utilization, maintenance needs) and start/stop commands need to be exchanged. To match the skills offered by heterogeneous agents with production requirements, [10] propose the use of an ontology. This enables generating the production plan. In short, all agents have to register their skills, using the so-called 'skill tracker'. A production order is sent to the system and is coordinated via the 'production tracker'. The 'orchestrator' communicates with both, skill and production tracker in order to compile the scheduling of orders to agents with adequate skills and acceptable utilization status (also with regards to maintenance tasks). The orchestrator represents an order interface that only communicates with the skill tracker and the production tracker, not directly with the system components.

Comparable to [1], in the DF a collection of multiple agents provides different skills. Here, the orchestrator in combination with the skill tracker corresponds to the 'plant agent' proposed in [1]. Similarly, the production tracker corresponds to the 'coordination agent' in [1]. Similar to [11], who propose a communication architecture for CPPS, where the different applications are running on the same machine but in different containers to ensure process isolation and data integrity on a real-time operating system, also the DF has to serve flexibility requirements close to a real-time clock pulse. The chosen protocol, OPC UA is an industrial standard that provides a rich service set [5] and serves as a communication middleware for multi agent systems [12]. An aggregation server can be used for monitoring and integrating different information models [12][13][14].

An additional benefit is, that OPC UA can be easily incorporated into other software products like supervisory control and data acquisition systems (SCADA-Systems), manufacturing execution systems (MESs) and enterprise resource planning systems (ERP-Systems). This benefit has been examined in [15][16][17]. This so-called vertical integration is one of the goals of Industry 4.0 [18].

3. Communication- and data model

Figure 1 shows the simplified communication architecture for the DF. An order is composed of a bill of material (BOM) and a bill of processes (BOP). The BOM contains the list of necessary parts, semi-finished goods and other components required for production. The BOP is a list of process steps to be executed to create the product. Figure 5 shows the entire implemented data model of the Digital Factory as a class diagram. Also, the information flow is mapped.

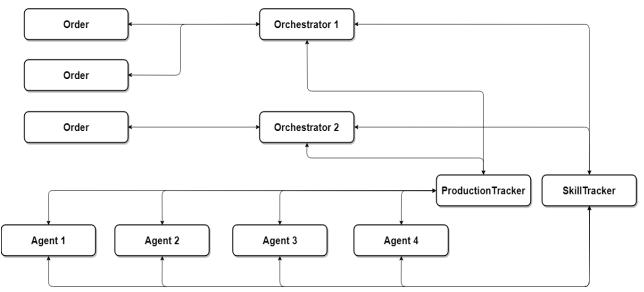


Fig. 1. Simplified communication architecture of the agents in the CPPS with only one skill tracker and only one production tracker.

The architectural approach is generic, insofar it allows for the application of various planning approaches: For instance, a scheduling algorithm could be implemented at different system levels – as well centralized (e.g., controlling the production tracker or the orchestrator) as de-centralized, (e.g., leaving the scheduling decisions to the negotiation of singular agents among each other). Thus, the generic communication path by which the orchestrator handles an incoming order is an inquiry to the skill tracker whether the necessary capabilities are available in the production system. The tracker responds with an adapted BOP in which the possible agent identifiers for each process step are enclosed. In case there is no possible agent for a process step, the order cannot be processed. Otherwise, it is forwarded to the production tracker.

Figure 2 shows the relationship between the orchestrator and the agent. Both are derived from the same base class and provide basic functions for starting a process (execution), aborting the process, checking whether a process step is executable, and estimating the duration of a process step. In addition, the agent has a function to update the data of the underlying hardware. Important information required by each component are the pose of the repository, the system status, the elapsed time, the time until the next service interval, and the participant's skill set. This information is visible to each participant in the network. Moreover, each participant needs to know which discovery servers to register with. Discovery servers offer the possibility to manage a list of servers. These servers report periodically to their configured discovery servers. This means that only the address of a discovery server needs to be known to see all registered servers in the network, since all registered servers can be read out. Multiple discovery servers are one way to increase the reliability of the system. These considerations are explored in Section 4.

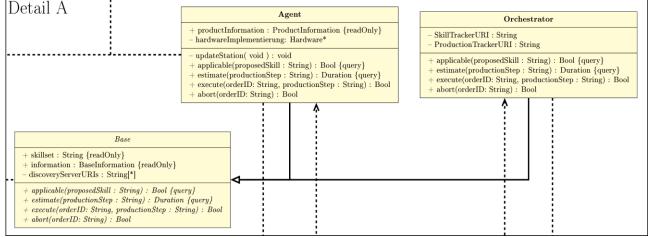


Fig. 2. Detailed view of the relationship between the agent and the orchestrator.

The pose captures a point and an orientation in threedimensional space. In addition, a frame is specified. Through the frame it is even possible to integrate different production lines (assuming that the physical transportation issue is solvable). The derived classes need additional information. The agent needs to know the available functionality, the product information of the current product to be manufactured, and a reference to the actual hardware implementation. The product information includes the current OrderID, a universally unique identifier (UUID) type 4 for unambiguous identification

of orders, the start time of a production order and the expected end. The hardware implementation is an abstract base class from which each station must be derived. The orchestrator only requires the universal resource identifier (URI) of the associated skill tracker and production tracker of the monitoring subsystem.

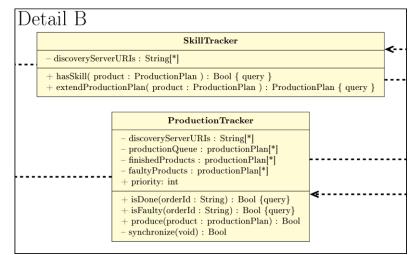


Fig. 3. Detailed view of the skill tracker and the production tracker.

Figure 3 shows the most important auxiliary components in detail. The skill tracker regularly checks whether new participants are registered on the discovery servers. From these, the skill sets are periodically queried and cached. Hence, the skill tracker offers two functions: One function is checking whether a current process step can be fulfilled. The other function adds the addresses of all agents which can execute a process step to the original production plan. The production tracker triggers the production start of a product, executes a status check to determine whether a manufacturing step has been finished, or the product is faulty. Altogether, the

production tracker is responsible for the coordination of the actual production and distributes the work orders to the stations that are not occupied or have a short waiting queue. In addition to logging in which production stages the various products are, the production tracker also triggers the transport framework for transport between the stations. Additionally, the synchronization function communicates with other production trackers and exchanges information regarding the state of the production. In the DF, the communication channels are still quite simple at this stage in order to later serve for different planning strategies at the business process level.

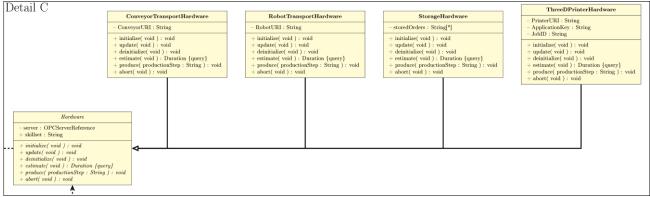


Fig. 4. Detailed view of the currently implemented hardware.

Figure 4 depicts the currently implemented hardware modules. Both transport classes require a network address to communicate over a proprietary protocol. Both transport mechanisms operate on the same principle. The source and target destination are queried and then the possibility of transport is evaluated. While the mobile robot can reach every station, it is much slower than the conventional transport with a conveyor belt. As the estimated completion time can be significantly different, this information needs to be considered in the scheduling process. The storage hardware keeps a list of the different queued orders. The storage unit is required to not only store finished products, but also semi-finished products. The printer class interacts with a fused deposit filament (FDF) printer. The required parameters for the current implementation have need of the network address of the printer, an application key and a job identifier. This is necessary since the current implementation interfaces are realized with a preexisting open source framework running on the printer.

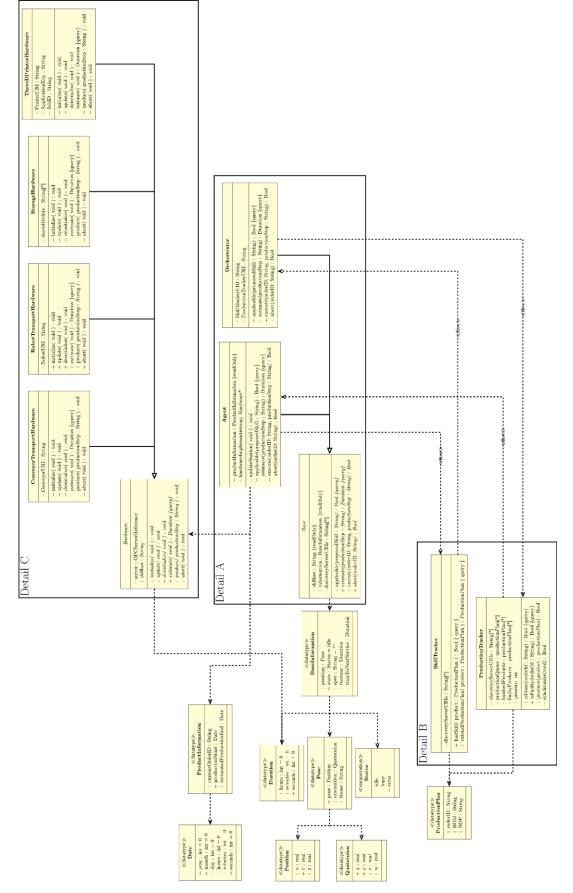


Fig. 5. Communication data model of the agents.

4. Reliability-Considerations

When designing a production system, it is important to consider the aspect of reliability. This concern is met with synchronization mechanisms in selected elements of the derived system. All participants work with a list of discovery servers. Therefore, the risk of a lost network is reduced. Multiple agents governing the same hardware are not planned. Additionally, the discovery servers register with each other.

The 'production tracker' is fitted with a synchronization between different trackers. Now multiple production trackers are responsible in the network. Adding a new production order to one of the trackers makes it available on all trackers. The responsibility is moved to the tracker with the highest priority. If one of the trackers were to stop working, the tracker with the next highest priority takes over. Updates to the production list are fanned out to all other trackers.

The 'skill tracker' can be started multiple times in the same network. As these trackers are only querying information from the agents or orchestrator, the additional trackers are ideal to decrease the risk of a network error and to distribute the information available. These simple modifications to the system are able to increase the reliability of the system.

5. Application

Currently the DF includes 15 different stations and is able to produce two different products. The first product, an axle bearing block, consists of four parts. The manufacturing process consists of seven steps and involves in total eight manufacturing devices. With regard to flexibility, the proposed communication data model allows for flexible scheduling. For instance, one of the available (and differently equipped) mobile robots has to pick up the pallet with the four parts from station 1 at production start. Intelligent path planning now takes the product to the next working step: It is transported to a milling station (a Kuka 6-axis robot together with a Pocket NC milling machine or alternatively an ABB robot). Subsequently, the pallet is brought to the assembly station, equipped with a portal robot from Wittmann and several actuators and sensors from SMC as well as a programmable logic controller from Siemens to assemble the four individual parts.

The second product, an object with several components either 3d printed or machined, consists of seven parts. Three of those components are created when they are demanded. The other parts are screws which are required to hold the components together.

Further process steps (quality inspection, packing) are to follow. As meanwhile the mentioned data model helps to coordinate the process and to send the necessary commands for the steps, a flexible path or tool change has been enabled. Integration times for new or alternative machines could be remarkably reduced. Since this progress has only recently been implemented, analyses that quantify the time savings achieved are still pending.

Additionally, all agents in the system can be integrated into a SCADA system. To reduce the needed server URIs, an accompanying aggregating server is used. The server queries the defined discovery servers and integrates all information points exposed by the registered servers. The aggregating server can be configured to allow not only supervisory access, but also direct control access. This aggregating server also reduces the complexity when subsystems need to be exposed

from one network to another. Currently, the aggregating server exposes the state of the agents in the DF to a SCADA system in a different network.

6. Conclusion

With digitalization, networking is becoming increasingly important, so a higher-level data model is an advisable means to simplify, further automate and accelerate the information exchange between devices from different manufacturers. Besides, a precise and semantically rich process monitoring opens up attractive future options for use cases with regards to predictive maintenance, virtual or augmented reality and many other digital technology applications. Not only operational flexibility could be enhanced, but also the structure of the production process could achieve remarkable adaptivity enhancements: as soon as new tools and systems can be integrated seamlessly, the whole production topology could be re-organized within short notice. In the case of the DF, e.g., the milling robot could do physically different tasks, e.g., packing or loading. Similarly, the robot in station 1 (currently the interface to the receiving warehouse and production start) could execute other process steps, only limited by physical restrictions such as weight and working space.

The proposed and implemented system connects different parts of already available components and combines them in a new way. The implementation of the communication framework has proven to work smoothly within the examined use case. We have shown, that the relatively simple framework is able to cope with a distributed manufacturing system. First steps regarding the evaluation have been started with the integration of additional hardware and a new product. The validation and evaluation of the implemented framework can be shown by integration of new agents and manufacturing of more products. The current inclusion of additive manufacturing techniques extends the flexibility of the production system. Even though the framework has been designed with reliability in mind, redundant hardware is still required. Further steps are the implementation of smarter planning procedures, such as flexible priority rules, and an adaptive split between central and de-central control paradigms.

Acknowledgments

We thank all sponsors of our Digital Factory for their support, in particular ABB, KUKA, FANUC, SMC, FESTO, SIEMENS, WITTMANN, BECKHOFF, Pilz, Sick.

The city of Vienna (MA23 – Economic Affairs, Labour and Statistics) deserves thanks for their kind support of the research project ENGINE (MA 23 Project 22-04).

References

- [1] Vogel-Heuser, B., Diedrich, C., Pantforder, D., Gohner, P., 2014. Coupling heterogeneous production systems by a multi-agent based cyberphysical production system, in: 2014 12th IEEE International Conference on Industrial Informatics (INDIN), IEEE. doi:10.1109/indin.2014. https://doi.org/10.1109/INDIN.2014.6945601
- Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., Ueda, K., 2016. Cyber-physical systems in manufacturing. CIRP Annals 65, 621–641. https://doi.org/10.1016/j.cirp.2016.06.005

- [3] Engelhardt-Nowitzki, C., Aburaia, M., Markl, E., 2017. Smart manufacturing in the digital factory–a practical case study of an industrie 4.0 implementation, in: EUROCAST, pp. 180–181.
- [4] Rauer, J.N., Wöber, W., Aburaia, M., 2019. An autonomous mobile handling robot using object recognition, in: Proceedings of the ARW & OAGM Workshop 2019, TU Graz. doi:10.3217/978-3-85125-663-5-14.
- [5] Mahnke, W., Leitner, S.H., Damm, M., 2009. OPC Unified Architecture. Springer Berlin Heidelberg. <u>https://doi.org/10.1007/978-3-540-68899-0</u>.
- [6] Monostori, L., Váncza, J., Kumara, S.R.T., 2006. Agentbased systems for manufacturing. CIRP Annals 55, 697– 720. <u>https://doi.org/10.1016/j.cirp.2006.10.004</u>.
- [7] Lee, J., Bagheri, B., Kao, H.A., 2015. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manufacturing Letters 3, 18–23. <u>https://doi.org/10.1016/j.mfglet.2014.12.001</u>.
- [8] Tomiyama, T., Moyen, F., 2018. Resilient architecture for cyber-physical production systems. CIRP Annals 67, 161– 164. <u>https://doi.org/10.1016/j.cirp.2018.04.021</u>.
- [9] Wang, S., Wan, J., Li, D., Liu, C., 2018. Knowledge reasoning with semantic data for real-time data processing in smart factory. Sensors 18, 471. <u>https://doi.org/10.3390/s18020471</u>.
- [10] Engel, G., Greiner, T., Seifert, S., 2018. Ontology-assisted engineering of cyber–physical production systems in the field of process technology. IEEE Transactions on Industrial Informatics 14, 2792–2802. <u>https://doi.org/10.1109/TII.2018.2805320</u>
- [11] Kaestner, F., Kuschnerus, D., Spiegel, C., Janssen, B., Huebner, M., 2018. Design of an efficient communication architecture for cyber-physical production systems, in: 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE), IEEE. pp. 829–835. <u>https://doi.org/10.1109/COASE.2018.8560563</u>

- [12] Großmann, D., Bregulla, M., Banerjee, S., Schulz, D., Braun, R., 2014. OPC UA server aggregation – The foundation for an internet of portals, in: Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA), IEEE. pp. 1–6. <u>https://doi.org/10.1109/ETFA.2014.7005354</u>.
- [13] Banerjee, S., Großmann, D., 2017. Aggregation of information models—An OPC UA based approach to a holistic model of models, in: 2017 4th International Conference on Industrial Engineering and Applications (ICIEA), IEEE. pp. 296–299. https://doi.org/10.1109/IEA.2017.7939225.
- [14] Seilonen, I., Tuovinen, T., Elovaara, J., Tuomi, I., Oksanen, T., 2016. Aggregating OPC UA servers for monitoring manufacturing systems and mobile work machines, in: 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE. pp. 1–4. https://doi.org/10.1109/ETFA.2016.7733739
- [15] Rohjans, S., Uslar, M., Appelrath, H.J., 2010. OPC UA and CIM: Semantics for the smart grid, in: IEEE PES T&D 2010, IEEE. <u>https://doi.org/10.1109/TDC.2010.5484299</u>.
- [16] Nguyen, V.H., Besanger, Y., Tran, Q.T., 2016. CIM and OPC UA for interoperability of micro-grid platforms, in: 2016 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), IEEE. <u>https://doi.org/10.1109/ISGT.2016.7781235</u>
- [17] Hoffmann, M., Meisen, T., Jeschke, S., 2017. Opc ua based erp agents: Enabling scalable communication solutions in heterogeneous automation environments, in: Demazeau, Y., Davidsson, P., Bajo, J., Vale, Z. (Eds.), Advances in Practical Applications of Cyber-Physical Multi-Agent Systems: The PAAMS Collection, Springer International Publishing, Cham. pp. 120–131. https://doi.org/10.1007/978-3-319-59930-4_10
- [18] Lara, M., Saucedo, J., Marmolejo, J., Salais, T., Vasant, P., 2018. Vertical and horizontal integration systems in industry 4.0. Wireless Networks <u>https://doi.org/10.1007/s11276-018-1873-2</u>