Edge Computing in Industrial IoT Framework for Cloud-based Manufacturing Control

Silviu Raileanu, Theodor Borangiu, Octavian Morariu, Iulia Iacob
Dept. of Automation and Industrial Informatics, Politehnica University of Bucharest, Romania
{silviu.raileanu; theodor.borangiu; octavian.morariu; iulia.iacob}@cimr.pub.ro

Abstract– Edge computing is essential for the Industrial IoT as framework for data acquisition from shop floor devices; distributed intelligence will shift to the edge for speed reasons in real-time handling of big data. This research aims at developing a generic architecture for information and data collection, smart processing and aggregation at the edge of large-scale manufacturing control systems; the edge is represented by the set of shop floor entities (things) – resources and intelligent products that are agentified and communicate in multi-agent systems for decentralized MES tasks. The IIoT architecture integrates a private cloud platform with a network of IoT aggregation nodes composed of IoT gateways, sensors and PC-type workstations hosting the resource agents. Both networks form the distributed MES layer of a semi-heterarchical, cloud-based production control system. The implementing solution is given; experiments report communication with the cloud.

Keywords – Agent-based Systems, Networked Control, Internet of Things, Manufacturing Systems

I. INTRODUCTION

The digital transformation and smart integration of shop floor devices with control software caused an explosion in the data points available in large scale manufacturing systems. The degree at which enterprises are able to capture value from processing this data by extracting useful insights from it represents a differentiating factor on short and medium term development and optimization of the processes that drive the manufacturing operations. Data processing involves three important problems: i) aggregating at the right logical levels when data originates from multiple sources, ii) aligning the data streams in normalized time intervals and iii) extracting insights from real-time data streams. All these dimensions should be considered in the context of scale, which means that the processing of this data streams must be scalable linearly so that the overall processing time remains low [1], [2].

To generate and use knowledge, much of the value that big data retrieved from the shop floor and working environment and analytics at control (Manufacturing Execution System – MES) and supply chain levels can bring in manufacturing requires real-time access and differentiated use of data from multiple edge and end point sources throughout the shop-floor [3]. The data is constantly growing in quantity, diversity and complexity: its availability and significance depend on reaction levels (from micro-seconds to years); it is multi-dimensional (time, location, energy, usage etc.) and multi-spatial (inside and outside the enterprise). More, the data is concerned with limited reliability, limited accuracy and obsolescence, and also with different levels of priority (from on-the-fly reaction needs to time-independent and a posteriori analysis needs). This requires on one hand the use of extended digital models of manufacturing equipment, processes, sys-tems and products, and on the other hand the distribution of intelligence at shop floor level This is done by defining agents for all shop floor entities, providing them reasoning capabilities and processing power while letting them collaborate in Multi-Agent Systems (MAS) to attain common, global goals at batch production level [4].

The integration of IT and Operational Technology (OT) in the Industrial Internet of Things (IIoT) enables the "smart factory" concept. This concept consists of new production control paradigms and environment monitoring techniques using connected devices that are able to collect, process and transmit data. This data is used to extract knowledge, optimize scenarios and generate predictions to improve efficiency, accuracy, and cost. One of the potentially biggest benefits is intelligent decision-making due to the access to relevant and high-quality information extracted from big data gathered both at shop floor and business enterprise layers [5], [6].

The operational level of manufacturing control is related to the physical world which is put in evidence by the IoT with its smart devices. For efficiently monitoring resources and the work in progress it is necessary to process and analyse real-time data streams. The data acquired in real-time by the agents representing two types of shop floor entities, resources and products is processed in the MES with distributed intelligence (dMES) implemented by the agents' delegate MAS. The scope of real-time processing of the shop floor data is twofold:

- Tracking continuously the status of resources in terms of availability for assigned tasks and evaluation of the quality of services they provide for measured energy consumption; this processing tasks are performed by the resource agents and need rapid reaction for resource team reconfiguring at breakdown or major degradation of resource performances.
- Machine learning and deep learning schemes applied to shop floor data retrieved from processes, resources and products for process optimization, quality inspection and preventive maintenance which is done at supervisory level in the cloud [7], [8].

978-1-5386-4444-7/18/$31.00 ©2018 IEEE

For large-scale manufacturing systems the coordination level is virtualized being implemented in private cloud models with IaaS model executing aggregated software applications based on high performance computing for capacity and demand matching and optimized decision-making, supported by big data analytics (DA).

An agent-based model with distributed intelligence for big data aggregation and analysis is presented in [9]. This model is organized on two layers of agents the capabilities of which can be configured. The data streaming and primary analysis is the responsibility of low-layer agents; they provide information about products, resources and production processes such as work-in-progress, resources' operational state and availability or quality inspection results. These agents collect data from shop-floor devices and analyse it in real-time; they are fixed (resource agents) or mobile (embedded on products travelling in the production structure). These agents cooperate to identify events and decide upon job allocation or rescheduling. Agents placed on the higher level process time-ordered data, track the covariance of multiple monitored metrics, and analyse big amounts of historical data available from shop floor actions, contextual or external aggregated information. They use this knowledge as support to decision: optimization of global cost functions, reconfiguring resource teams, predicting behaviours and events, evaluating preventive maintenance.

New MES designs aim at shifting in the real-time domain some of intensive computational tasks the result of which strongly influence the resource utilisation and production costs [10]. Such a global task at batch level is the optimization of product planning, scheduling and resource allocation based on distributing intelligence at shop floor's edge for data retrieval, pre-processing and high speed connectivity with private cloud platform. High performance computing tasks act in this case as global problem solver and return results in real-time to the dMES controls for product routing with continuous optimization of batch costs and resource usage.

The paper is organized as follows: Chapter 2 discusses the Edge Computing concept and issues for implementing in the Industrial IoT framework for shop floor data-driven manufacturing control systems. Chapter 3 presents the IIoT architecture for data acquisition and intelligent processing, composed from two networks of IoT gateway devices for fixed manufacturing resources and aggregation nodes for intelligent devices embedded on products carriers travelling between workstations. Experimental results and conclusions are reported in Chapter 4.

II. EDGE COMPUTING IN THE IIOT FRAMEWORK

Several concepts arose in the last years in order to bring together informational and physical objects. From a chronological point of view there was first the Internet which enabled applications interconnection facilitating efficient communication. Then, physical devices started to be connected to the Internet which made them smarter and more efficient by exchanging information with both cloud and peer systems. This second phase changed the way process control was conducted. The new concept is called Internet of Things (IoT) or Physical Internet (PI) and applies the technologies and methods of the digital Internet to the physical world. In this context, where multiple actors (both people and companies) interact in order to deliver products and services both time and cost effective while the process as a whole remains sustainable, requirements like real-time traceability, automatic collection of data from fixed and mobile objects, integration of this data into the cloud to analyse it, optimize processes and take decisions are important objectives.

OT and IT alignment in the IIoT framework improves data accessibility performed by a stable and fault-tolerant IT infrastructure for an OT environment. With edge computing a greater volume of high-quality data from the OT side can be obtained without impacting the current Supervisory Control And Data Acquisition system. With cloud and virtualization technologies, manufacturing-floor servers can be moved to the cloud, helping to reduce equipment and extending the MES functionalities (transferring the system scheduler, machine learning and prediction-generating technology in the MES, managing unexpected events, real-time resource reconfiguring based on digital models of processes and assets uncontaminated by functionalities, resource operating in today′s standard IT security protocols, a.o.) [11], [12].

Edge computing is essential for IIoT as framework for data acquisition from shop floor devices; distributed intelligence will shift to the edge for speed reasons in real-time handling of big data. Instead of transporting all data over the network and then processing it, for instance in the centralized cloud-based MES, some operations will be performed close to the IIoT device (endpoints: sensors and embedded devices) and application, hence at the edge of the network or the endpoint. A new perspective is brought to the industrial IoT space by integrating intelligence and computing capabilities directly into small-footprint edge devices – *IoT gateways* (Fig. 1 up). A software platform is needed to process data directly on distributed, small-footprint edge devices (or sensors) rather than sending all data to the private cloud for processing; this technology minimizes thus latency and simplifies the data exchange between the centralized part of the MES (the cloud IaaS) and the distributed part of the MES (the delegate MAS).

In this context the goal of this research is to develop a generic architecture for information and data collection, processing and aggregation at the edge of large-scale manufacturing control systems; the edge is represented by the set of shop floor entities (things) that are agentified and to which intelligence is added:

- *resources*: sets of sensors continuously generate data about: operating parameters (vibrations of machine tool axes, forces at robot grippers, electrical parameters of actuators, drives and processing boards, image parameters of virtual cameras, a.o.), tool parameters, resource status (errors at program execution, programs enabled / disabled, power off at collisions, calibration errors, a.o.), quality of operations performed (duration, part recognition and locating by vision, a.o.), and energy consumption (continuously, per product and operation);
- *products*: human-machine interfaces with access to the cloud provide information and data about the product's desired recipe while intelligent devices embedded on the

physical products (single-chip processors with WiFi connectivity are placed on the pallet carriers that move products on the conveyor belt between resources which successively act upon them) collect information and data about the products' current execution status, quality control results (geometry measurements, shape finishing, alignment of subassemblies), and events that occurred during execution (power off and machining retrace at recovery;

- *orders*: intelligent embedded devices on product carriers aggregate information and data about the way the product's recipe is transposed in a dedicated batch entry for execution, sequence of operations with precedencies and resources assigned for each operation, about fulfilment of the product's execution order and eventual unexpected events, about the product's location in the shop floor, operations already performed, timeliness and delays relative to the current schedule, and delivery time;
- *environment*: the environment is continuously monitored in plants where special products are manufactured (e.g. radiopharmaceuticals; networked sensors collect, weigh, and pre-process temperature, relative humidity, pressure, radioactivity, a.o. data) or in workstations where vision systems are used (sensors report lighting variations).

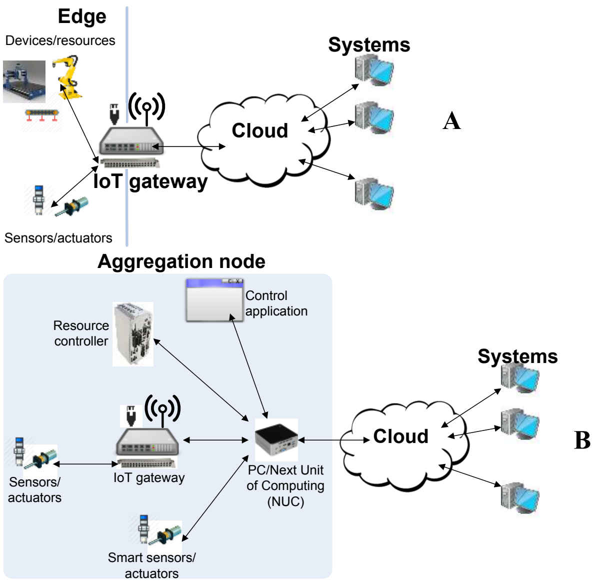

Fig. 1. IoT gateway (A) and Aggregation node (B) concepts for large-scale manufacturing control systems

In the proposed IIoT framework for complex production control, order agentification is realized with IoT gateways implemented with small Overo AirStorm system boards from Gumstix [13] and embed intelligence on product carriers.

In order to integrate and pre-process a higher volume of information generated by multiple resource sensors, energy measuring devices and environment monitoring sensors in a particular shop floor area (a workstation or a resource area), the concept of IoT gateway was extended to the *IoT aggregation node* one (Fig. 1 down). This type of edge system supports multiple communication protocols (inter-agents and agent-cloud), aggregates various types of data collected with different timings from a large number of sensors and adapts data processing to user requirements (evaluate resource behaviours and QoS, detect anomalies, predict unexpected events) [17], [18]. An aggregation node is an extension of IoT gateways allowing multiple devices to connect to the cloud using a centralized point (usually a PC); these devices can be IoT gateways, smart sensors/actuators or industrial controllers. While IoT gateways have usually limited communication capabilities and processing power, the aggregation node, being PC-based, supports multiple communication protocols and is flexible enough to run customized software. The resources are agentified in the IIoT framework for manufacturing control with IoT aggregation nodes.

III. Distributing Intelligence at the Edge in Cloud Manufacturing

The proposed semi-heterarchical manufacturing control architecture performs the following tasks:

1. Configuration of the resource set; planning products in the batch, scheduling operations for products and assigning resources to operations; prediction resource behaviours and energy consumption, detecting anomalies based on machine learning; cell and production monitoring. These tasks will be done for batch orders received and accepted on the upper MES level implemented in a private cloud platform.
2. Automatic control of product routing and execution of batch orders; on line rescheduling at disturbance occurrence through collaborative decisions taken by order agents at de-centralized dMES level. The information flow from the lower dMES control level to the System Scheduler in the cloud concerns: i) the resources' status, behaviour, energy consumption and QoS performed at termination of any operation on products in current execution; ii) occurrence of predicted (operation / product termination) and unexpected (resource breakdown, rush order, storage depletion) events.

The overall control of the batch production is semi-heterarchical, with hierarchical, centralized optimization of mixed batch planning and product scheduling coexisting with heterarchical, decentralized control of product execution. The initially computed optimal schedule is taken as a recommend-dation, being applied as long as the resources' capabilities are not altered. This dual control topology with on line switching modes and real-time rescheduling is feasible by running in the cloud optimization programs on differentiated time horizons up to the farthest (batch) one. A proper timing must be selected to: i) check continuously the resources' behaviour and evolution of their capabilities, to identify the degradation of these performances, to predict unexpected events and energy consumptions, and ii) to update optimal schedules [16]. No significant event should be lost (not detected), while the global control system should not become too nervous. A suitable timing scheme would be to (eventually) update the optimal production schedule each time a product's execution is finish-ed, while updating the resources' status, KPIs and energy consumption each time an operation on a product is finished. In this timing, the cloud System Scheduler updates the optimal sequence for the remaining products to be executed, and the dMES order agents reschedule in cooperation the products in current execution. This rule diminishes the system's myopia.

Fig. 2 shows how is distributed the intelligence at the edge of the shop floor in the heterarchical agent-based dMES.

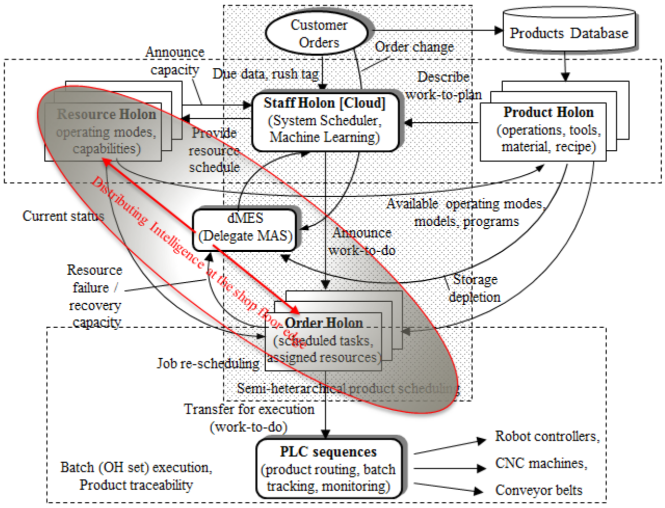

Fig. 2. The holonic model for a large-scale manufacturing control system

The model uses the three types of basic agents to which their physical counterparts are related: equipment, goods and actions; three basic holon classes interact in real-time to put in practice and update optimized execution schedules: *resource holons*, *product holons* and *order holons*. A supervisor class – the *staff holon* – optimizes production schedules at batch horizon in real-time. This model uses the principles defined in the PROSA holonic reference architecture [17].

By collecting in real-time the resources' status, the QoS performed and the energy consumed, an optimization model initially computed in the cloud can be updated and re-run with certain timing in order to maintain the best global batch cost functions (e.g., execution time, energy consumption, balanced resource usage, a.o.). In this research, the cost is composed of the total production time (makespan) and the energy consumed; energy is influenced by the makespan (energy = power × time). Tests were carried out in the experimenting stage for the update processes of instantaneous power and energy. The events that can alter the computed best schedule fall into two categories: *hard change* of the resource's state (e.g., resource/operation failure), and *soft change* of the state (degraded resource parameters causing increase of utilisation costs, e.g.: increased execution time or energy consumed).

The events are configured initially in the cloud based on experiments and history; they are identified during production by the resource holons that communicate with the order holons. Hard changes may be detected anytime and cause: i) automated reconfiguring of the resource team and operations rescheduling in the cloud for the not yet executed products in the batch, and ii) operations rescheduling in the dMES by collaborative decisions of the MAS of order agents. Soft changes are evaluated whenever an operation is finished for any of the products (order holons) in simultaneous execution, and may cause centralized operations rescheduling in the cloud for the products not yet executed whenever a product is finished and the increase in the soft change parameter chosen (execution time or energy consumption) exceeds a predefined threshold `thr` (experimentally computed).

The proposed IIoT architecture for data acquisition and intelligent processing integrates in the distributed MES layer two networks of devices:

- A *network of IoT gateway devices* in which data is handled by the mobile order agents residing on intelligent devices) embedded on product carriers. The lifecycle of an order agent corresponds to the execution time of the product it represents; when a product is finished and exits the shop floor, the IoT gateway device located on the pallet carrier will receive from the PLC supervising product routing the data to be used by the order agent of the next product which will be progressively created on the available pallet carrier. The number of IoT gateway devices (n) corresponds to the number of products simultaneously executed; the residing order agents communicate with the resource agents at operations rescheduling for the n products in current execution imposed by a hard change in the status of one resource (breakdown or strong performance degradation). This rescheduling is made in collaboration by all order agents that are integrated in a MAS framework with WiFi communication; order agents send data to the cloud concerning the work-in-progress (Fig. 3).
- A *network of aggregation nodes* composed of Arduino ETH boards (IoT gateways), sensors and PC-type workstations host the resource agents. Data is collected directly from resources, and processed in real-time by the computer, for the tasks defined at point 1 above (Fig. 4).

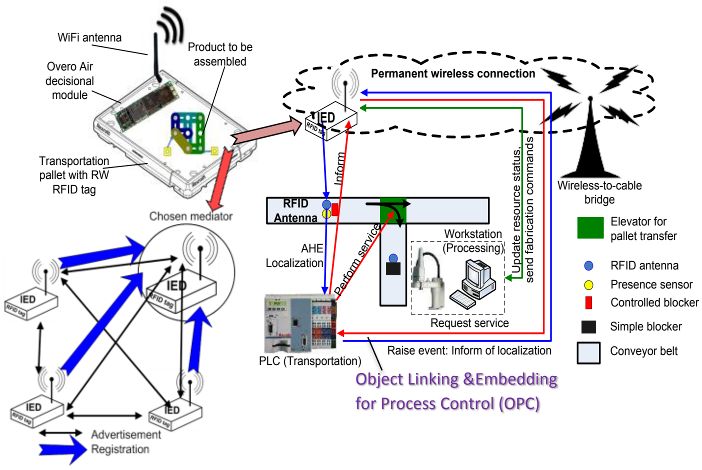

Fig. 3. Order agents running on IoT gateway devices embed intelligence on products, negotiate resources and take collective decision on "next operation assignment" at shop floor disturbances (e.g., resource breakdown, high power consumption, low QoS)

For example, if resource allocation for product operations is based on energy consumption, real-time data is collected and locally processed to obtain the instantaneous power which is fed to the cloud database in order to update the power record (instantaneous power in time) and to the respective resource agent residing on the PC to calculate the energy consumed for a performed operation.

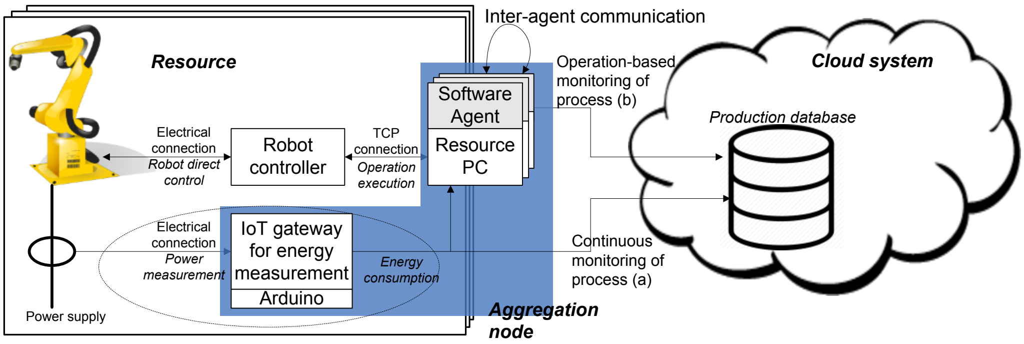

Fig. 4. Architecture of an IoT aggregation node and integration with the Cloud scheduler for a) continuous and b) operation-based data tasks

In case a) the instantaneous power $P_I = I_{RMS} \times U_{RMS}$ is sent directly to the cloud using a HTTP POST request to a PHP / MySQL application located on the cloud. In the second case b) the sequence of messages is: the resource agent receives from an order agent (in charge of product routing and control of job execution) an operation execution request; the current consumed energy is read from the Arduino board and a start operation signal is issued to the resource; the resource executes the operation and at the end signals its completion to the resource agent; the agent reads again the current consumed energy and by subtracting the first energy value from the second one calculates the energy consumed for that operation (Fig. 5). The resource agent can monitor resource parameters like temperature, torque, power supply of drives and CPU boards, positioning error, success of object recognition, a.o. This information is written to the cloud database using ODBC.

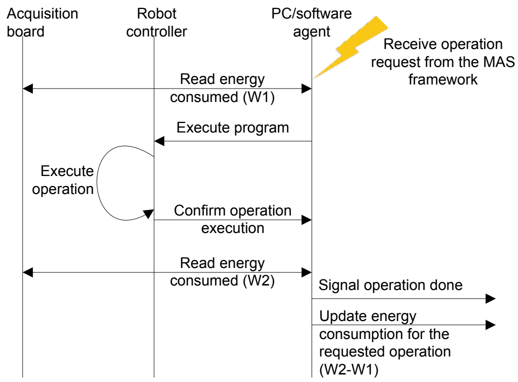

Fig. 5. Sequence of messages in the operation-based monitoring process of the resource agent executing on an IoT aggregation node

Both IoT device networks communicate between them and with the Cloud IaaS to update process execution knowledge and apply decisions taken at hierarchical (Cloud) and heterarchical (delegate MAS) levels in production.

IV. Experimental Results and Conclusions

The types of IoT device networks have been designed and implemented for a 6-workstation (4 robotized material assembly workstations from which 2 include CNC machines for material processing, 1 robotized pallet for product I/O workstation, and 1 robotized part storage workstation) shop floor. The robots work with 2D vision systems.

Experiments have been carried out to test several solutions for collecting parameters and internal variables from manufacturing resources – the Omron industrial robot Adept eCobra [18]: a) using the proprietary programming software Adept ACE; b) using a library offered by Adept to read variables from external programs; and c) using a set of application programs running both on the robot and on the aggregation PC node which communicate over TCP. This last solution is part of the aggregation node: an application running on the PC that is able to connect to the industrial equipment to read data, and that can also connect to the database located on the centralized cloud MES to write data. Using these methods the data collected from robot resources (Fig.6 i, ii) includes:

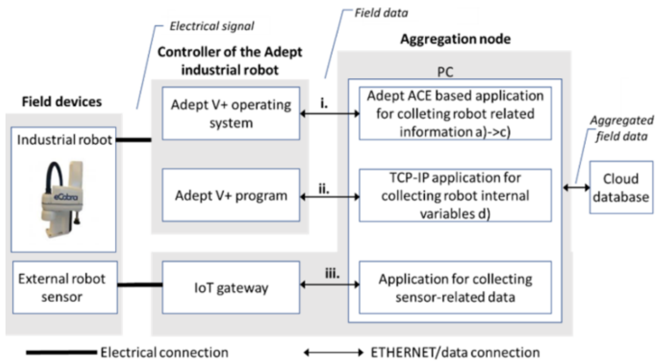

Fig. 6. Data gathering from resources and aggregation at dMES level

a) Information gathered from the robot: *input voltages; temperatures of processing board, encoders*, and amplifiers; *joint motor torques*; and *positioning errors*;
b) Information obtained from belts the encoders of which are connected to the robot system: *belt velocity, instance count, instances per minute* or *faults*;
c) Information about the process: *processing time*, *idle time*, *parts per minute*, *parts processed* or *not processed*;
d) Information about the monitored variables which represent the Cartesian locations and joint, status of digital inputs and outputs, robot parameters (e.g.: speed) and state of programs.

Additionally, information from the robot environment sensors is collected using the Arduino IoT gateway and sent for aggregation at the PC-based aggregation node (Fig.6, iii). All the data gathered from the robot through the previous sources (ACE application/library and TCP/IP protocol) is sent as clear text to the aggregation application in charge of connecting and writing records to the database.

The gathered data consisting of production parameters is forwarded for storage to a private IBM CloudBurst cloud platform. The validating scenario consists in experiments with different methods of gathering information from an embedded device using low and high-level communication protocols. The chosen communication protocols range from transport layer (UDP and TCP) to application layer (HTTP requests]) and publish / subscribe messaging protocols on top of the TCP/IP stack like MQTT. To evaluate the database update frequency, communication latency and loss of information a scenario consisting of 100 messages containing one single timestamp written at source based on the information gathered from a time server (time.nist.gov) was performed, see Table 1.

TABLE 1. COMPARATIVE ANALYSIS OF FOR IOT GATEWAY DEVICES - CLOUD COMMUNICATION PROTOCOLS

Communication type	*Sent messages*	*Received messages*	*Average delay [sec]*	*Average frequency*	*Observations*
UDP	100	100	0.68	18msg/sec	point-to-point
TCP	100	100	0.65	13 msg/sec	point-to-point
HTTP	100	100	1.2	8 msg/sec	point-to-point
MTTQ	100	100	0.54	18 msg/sec	many-to-many

The order agents residing on the aggregate nodes run JADE agents. The communication is achieved at two levels: hardware and software interoperability. For the first level the devices on which the order JADE agents execute must access the local network (agents from different classes must have direct IP visibility to join the same platform and communicate directly) and must be mobile (wireless communication). The embedded system Overo AirStorm selected in this case offers wireless connectivity and uses an operating system able to run the virtual machine from Oracle needed for JADE. For software interoperability, JADE implements FIPA specifications (www.fipa.org) to assure compatibility at inter-platform level: Message Transport System (MTS) responsible with message delivery between agents, Agent Management System and Directory Facilitator. JADE implements the standard FIPA MTP (a set of transport protocols and MTS associated encoding schemas), in order to sustain interoperability with other platforms, the default ones the IoT gateway network runs at start-up being a standardized HTTP-based MTP used for inter-platform communication and a proprietary MTP (JADE Internal MTP - IMTP) used for communication between agents running on the same platform.

The edge computing solution for data collection, smart processing and aggregation from shop floor resources and products with embedded intelligence was successfully tested with two IoT device networks. The key element of the solution is the aggregation node which concentrates information from different sources and writes records on a database located on a private cloud. From the experiments performed, it resulted that the MQTT protocol is the best choice for sending data from embedded systems to cloud. This is due to the following characteristics: it is a publish/subscribe solution allowing the realization of infrastructures which can communicate in the many-to-many case.

As a conclusion the paper presents a solution for shop-floor device interoperability for resources and intelligent products within a production control system. This solution is based on aggregation nodes which collect data from multiple directly connected devices and forward this data to the cloud control platform hosting the high level MES control system in charge of operation optimization, execution and monitoring.

REFERENCES

[1] Helo, P., Suorsa, M., Hao, Y., Anussornnitisarn, P. (2014). Toward a cloud-based manufacturing execution system for distributed manufacturing, J. Computers in Industry, no. 65, p. 646-656

[2] Thomas, A., Trentesaux, D., Valckenaers, P. (2011). Intelligent Distributed Production Control, Journal of Intelligent Manufacturing, 23(6), doi: 10.1007/s10845-011-0601-x

[3] Zhang, Y. et al. (2017). Game Theory Based Real-Time Shop Floor Scheduling Strategy and Method for Cloud Manufacturing, Int. J. of Intelligent Systems, Vol. 32, Issue 4, p. 437–46, doi: 10.1002/int.21868

[4] Governor, J. (2017). IBM Watson IoT and the digital twin. Industry 4.0, Published: February 24, 2017, RedMonk, http://redmonk.com/jgovernor/2017/02/24/ibm-watson-iot-and-the-digital-twin-industry-4-0/

[5] i-SCOOP (2017). IT and OT convergence – two worlds converging in Industrial IoT, https://www.i-scoop.eu/internet-of-things-guide/industrial-internet-things-it-ot/

[6] Industrial Ethernet Book (2017). IIoT: Combining the Best of OT and IT, IEB Issue 95 / 14

[7] Valckenaers, P., Van Brussel, H., Bruyninckx, H., Saint Germain, B., Van Belle, J., Phillips, J. (2011). Predicting the unexpected, J. Computers in Industry, 62(6), p. 623-637, Elsevier

[8] Morariu, O., Morariu, C., Borangiu, T., Raileanu, S. (2018). Manufacturing Systems at Scale with Big Data Streaming and Online Machine Learning. In Service Orientation in Holonic and Multi-Agent Manufacturing, Springer book series Studies in Computational Intelligence, Vol. 762, chapter 19, p. 253-264

[9] Queiroz, J., Leitão, P., Oliveira, E. (2017). Industrial Cyber Physical Systems Supported by Distributed Advanced Data Analytics. In SOHOMA 2016, Studies in Comput. Intelligence, Vol. 694, p. 47-59

[10] Zhang, Y. et al. (2017). Game Theory Based Real-Time Shop Floor Scheduling Strategy and Method for Cloud Manufacturing, Int. J. of Intelligent Systems, Vol. 32, Issue 4, p. 437–46, doi: 10.1002/int.21868

[11] Valckenaers, P., Van Brussel, H., Bruyninckx, H., Saint Germain, B., Van Belle, J., Phillips, J. (2011). Predicting the unexpected, J. Computers in Industry, 62(6), p. 623-637, Elsevier

[12] IIC (2017). Industrial Internet Consortium Quarterly Progress Reports 2015-2017, http://www.iiconsortium.org/quarterly-reports.htm

[13] Gumstix (2018), www.gumstix.com/

[14] Raileanu, S., Anton, F., Borangiu, T. (2017). High Availability Cloud Manufacturing System Integrating Distributed MES Agents. In Proceedings of SOHOMA 2016, Springer book series Studies in Computational Intelligence, Vol. 694, Chapter 2, p. 11-23

[15] Gloria, A., Cercasa, F., Souto, N. (2017). Design and implementation of an IoT gateway to create smart environments, Proc. of the 8th Int. Conference on Ambient Systems, Networks and Technologies, https://doi.org/10.1016/j.procs.2017.05.343

[16] Novas, J.M., Bahtiar, R., Van Belle, J, Valckenaers, P. (2012). An approach for the integration of a scheduling system and a multi-agent manufacturing execution system. Towards a collaborative framework, in Proc. of the 14th IFAC Symposium INCOM'12, Bucharest, IFAC PapersOnLine, p. 728-733

[17] Van Brussel, H.V., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P. (1998). Reference architecture for holonic manufacturing systems: PROSA, Computers in Industry, Special Issue on IMS 37, p. 255–274

[18] eCobra SCARA Robot, https://industrial.omron.us/en/products/ecobra, consulted in March 2018