Development of an Event-Driven System Architecture for Smart Manufacturing

Maksymilian Piechota[1], Mikołaj Nowak[1], and Dariusz Król[2][0000−0002−2715−6000]

[1] Faculty of Information and Communication Technology, Wrocław University of Science and Technology, Poland
[2] Department of Applied Informatics, Wrocław University of Science and Technology, Poland
dariusz.krol@pwr.edu.pl

Abstract. This paper describes the automated production data acquisition and integration process in the architectural pattern Tweeting Factory. This concept allows the use of existing production equipment with PLCs and the use of industrial IoT prepared for Industry 4.0. The main purpose of the work is to propose an event-driven system architecture and to prove its correctness and efficiency. The proposed architecture is able to perform transformation operations on the collected data. The simulation tests were carried out using real data from the factory shop-floor, services prepared for production monitoring, allowing the calculation of KPIs. The correctness of the solution is confirmed on 20 production units by comparing its results with the blackboard architecture using SQL queries. Finally, the response time for calculating ISO 22400 performance indicators is examined and it was verified that the presented solution can be considered as a real-time system.

Keywords: Application case studies · Data integration · Engineering optimization and design · Event-driven system architecture · Tweeting Factory.

1 Introduction

The paper focuses on a concept called Tweeting Factory [3, 8]. Currently, in the manufacturing industry for data acquisition, the PLC controllers are mostly used [5]. However together with the development of the Industry 4.0 concept, intelligent sensors that are capable to communicate with various other systems [10] are getting more recognition. Additionally they can do some initial data processing. In the case of intelligent sensors, the whole process can be done by the sensor itself and the output values can be sent further over the protocols like AMQP or MQTT. Those new features open a variety of new capabilities to the production line systems. The Tweeting Factory is an architecture proposal for utilizing new features and providing the capabilities. It is an architecture pattern providing a communication between production units, data processing services, and output data subscribers [12]. The machines are responsible for messages dispatched

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_38

that are then processed by the services and the services send new messages. All messages in the system can be utilized by the subscribers. Every tenant in the system has access to all messages – the architecture implements the Publish-Subscribe pattern [6]. There are no constraints for the messages exchanged in the system. Only a few recommendations are made for the messages' metadata, such as the origin, subject, and publishing time. The services can read or send new messages, communicate with external systems and other data sources in order to leverage external information, data processing outside the system, or data recording.

The paper aims to validate the described architecture. Furthermore, the experiment designed for the verification purpose is described: based on the data acquisition process, the production environment project was designed that was the basis for the simulation environment. The simulation experiment was performed with the real production data provided by the cooperating company. The simulation results were compared with the results calculated by the blackboard style data acquisition process. Additionally, the Tweeting Factory performance was measured and analyzed. Finally, in the last section, the conclusions and possibilities for further research are discussed.

Before describing the methods in detail, we introduce the related work that influenced our research. Tweeting Factory term was used for the initial literature review. However, only four papers [3, 8, 12, 11] were found using the term. Additionally, an alternative name for the architectural model was discovered – LISA, i.e., Line Information System Architecture [11]. Taking the alternative name into account and the low number of papers found in initial survey, there was a reasonable doubt that the Tweeting Factory concept exists, but the different name is used. Another term that was found during the analysis is called Digital Twin. It seems that Digital Twin is the higher abstraction concept and Tweeting Factory might be one of its implementations.

During the analysis, no comparison with the blackboard style SQL data acquisition method was found. The authors find this subject very important to eventually prove the advantages of the Tweeting Factory over the existing solution. Therefore, following research gaps were determined, that aims to be the authors contribution:

- RQ1: Is the Tweeting Factory concept correct?
- RQ2: Does the Tweeting Factory concept fulfill the real-time system constraints?
- RQ3: How the Tweeting Factory concept can improve the data acquisition process?
- RQ4: Is the Tweeting Factory a correct replacement of the SQL method?

2 Overview of the Tweeting Factory

2.1 Data Acquisition

Data acquisition is a process of measuring the physical values and storing them in a digital form [7]. The following phases of the process can be highlighted:

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_38

1. Physical value extraction
2. The value measurement by a sensor
3. The measured values transmission to a registration device
4. The values conversion to a digital form by ADC
5. Storing of the digital values

Data can serve different purposes, for example, KPIs computation, historical data analysis. event reporting such as machines' condition monitoring, environment parameters control, failure reporting, and material fatigue. Evolution of the Industry 4.0 concept is strictly coupled to intelligent sensors. In comparison with the previous generation sensors, intelligent sensors are capable to communicate with the other environment's participants, follow precise production, leverage production ontology, and use machine learning algorithms for independent conversation [10]. Industry 4.0 compatible sensors are capable to address all phases of data acquisition, besides the last one – the data storage. For this purpose, we use the cloud computing infrastructure. The paper's authors consider Tweeting Factory concept as a good candidate solution to address this requirement.

2.2 From Tweets to Decisions

Tweeting Factory concept provides additional benefits for the manufacturing environments. Additional features of the architecture were found in the papers [8, 12]: KPI indexes calculation, hybrid systems optimization, energy consumption optimization, production process planning, quality assurance, production machines management, and monitoring. It is possible to use Tweeting Factory concept for data storing only [3], however additional services provided by the concept create its value. For example, for KPIs calculation, the stored data from a requested period can be used instead of single data coming directly from PLCs. All above-mentioned data are calculated in real-time, so it is possible to immediately generate a notification event when the specific index reaches a predefined threshold. A significant advantage of Tweeting Factory allowing to add services operating on available data is the ability to connect to external services. Based on that, the systems using the architecture are capable to easily connect and take part in the Shared Industry concept [14] or to be used in other applications outside the manufacturing industry [16]. Considering the Tweeting Factory as a data stream, it is also possible to apply machine learning algorithms.

2.3 Tweeting Factory Architecture

The base sources of the data are production units. Temperature or humidity are examples of the working environment measured values. Machine's work parameters such as state (idle/active), specific state time, or smaller component state are also measured. The machines can send that information themselves or to the connected PLCs. Devices other than production units can be found on the production line, for instance, a barcode reader. They can also send data themselves or to the PLC controllers. Another category of data sources are services. Services

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_38

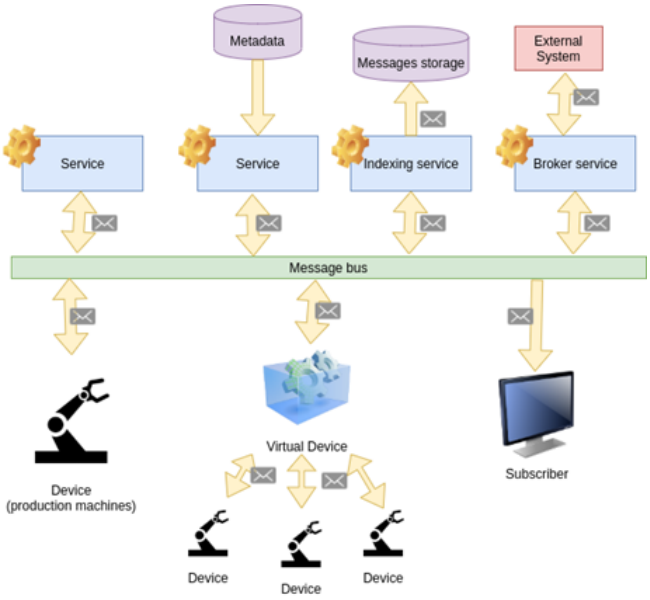


Fig. 1. The Tweeting Factory architecture model

can also subscribe by receiving various types of messages available on Tweeting Factory's bus. In most cases, the data published by the service is just a transformation of the data received. Tweeting Factory provides support to connect and leverage external systems. Metadata models proposed in [11] allow to decorate the messages with the context of the external systems. However, the metadata is not the only advantage. Tweeting Factory provides a way to emit events registered in external systems. It allows to gather context data, not limited to data generated by production machines, for example, information entered manually by an employee or imported from HR systems, such as work hours, holidays, employee availability, or workplace assignments.

The Tweeting Factory architecture is depicted in Fig. 1. Production units, including virtual ones, can measure and convert the physical values to a digital representation itself. However, they are not capable to store data. This responsibility belongs to indexing services and subscribers.

When dealing with messages from different sources, it is highly possible to get incompatible messages' formats. The incompatibility might be a reason for incorrect data handling. This problem is called the interoperability issue and is considered on a semantic level [13]. Production machines and external systems produce information with incompatible labels, what leads the subscriber service's data parsing issues. A well-known solution of the interoperability issue is an application of a domain ontology or an application of an existing information exchange standard, such as ISO 10303 [4]. Standardization of the system's message layout solves the interoperability issue on the semantic level [13]. The

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_38

literature provides different solutions [1], however, considering a variety of manufacturing industry applications, it is impossible to establish one unified standard. Different ontologies might be leveraged in providing a unified standard for the specific application. For example, in the paper, the authors decided to replace labels created by specific machines with URL labels. The interoperability issue resolved can be seen in Fig. 2. The external system's messages need to be translated by the broker service. For the production machines, if it is not possible to configure their messages' labels, translation services [15] should be introduced.

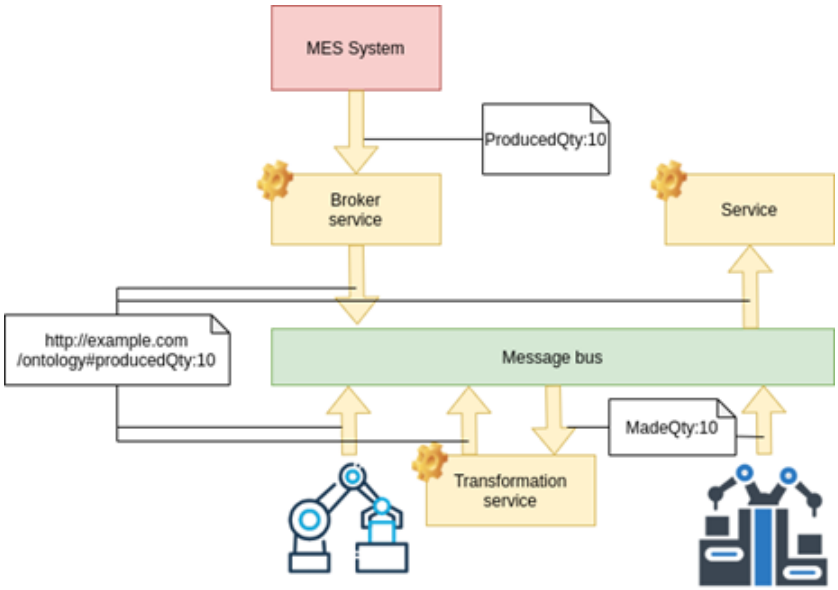


Fig. 2. Ontology based solution of the interoperability issue in the Tweeting Factory environment

3 Experiments

In this section, an experiment validating the Tweeting Factory concept and its results are presented. In the experiment, KPI metrics were calculated by Tweeting Factory simulation environment and by the blackboard style method. The analysis of the experiment's results allowed the authors to answer the presented paper's research questions. To prove the Tweeting Factory correctness (RQ1), the calculated metrics were compared with the metrics calculated by standard methods. The comparison proved also if the Tweeting Factory is a good replacement for the legacy SQL method (RQ4). Additionally, the performance of the system is measured and analyzed (RQ2).

The experiment was divided into the following steps:

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_38

1. Analyze source data.
2. Calculate KPI indexes by the standard method.
3. Design the new data acquisition method.
4. Complete the implementation.
5. Conduct a simulation run.
6. Analyze the results.

Source data analysis (1) aims to check if the provided production data is correct for the experiment, what quality it is, and what time range and units should be chosen as an input to the simulation. The initial KPI calculation (2) was done by the legacy SQL method. The OEE-based metrics were chosen for the experiment, considering their universality [2]. The data acquisition method design (3) requires the specific production infrastructure analysis and leads to the Tweeting Factory architecture preparation for the specific environment. Simulation implementation based on the previous step's results. A set of programming tools was prepared that played the service role in the Tweeting Factory environment. Another tools were verifying the resulting data. The simulation run (5) was a simple run of the prepared tools. The results analysis (6) aimed to compare the resulting KPI metrics with the standard method's initial calculation from the previous step (2). Additionally, calculation times were analyzed for performance metrics.

Real-world data provided by the heating devices factory was used in the experiment. It provides such features as: production order reporting, machines' production data automated acquisition, gathered data analysis. In the experiment, the automatically collected data as well as the data provided by the employees was used. The provided database served as a source of the production data.

After data analysis, the following data was determined to be required for the experiment:

1. Products data: the data source production machine's name, the product completion time, the amount of the successfully produced and rejected products, estimated reference unit time of work. Described data was available in various tables in provided database.
2. Production machine's work time: the machine's code, the event's start timestamp, the event's end timestamp, the event's type, the event's value.
   The event's value corresponding to work time is either Work or PW. This type of event can take value 0 (stopped working) or 1 (started working). These records were emitted every time a machine started or stopped to work and also at the top of every hour.
   Based on determined event types it is possible to determine KPI metrics to calculate:
   - PQ – Produced Quantity – based on products' completion events,
   - RQ – Rejected Quantity – based on products' completion events,
   - PRI – Planned Run Time – based on products' completion events,
   - APT – Actual Production Time – based on a machine's work time events,
   - PBT – Planned Busy Time – based on a machine's work time events.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_38

In order to calculate the OEE index [2], the following component metrics are required:

$$A = \frac{APT}{PBT},\ E = \frac{PRI * PQ}{APT},\ Q = \frac{PQ - RQ}{PQ} \tag{1}$$

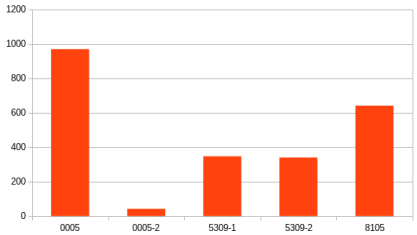
None of the above indexes can't reach a value greater than 1, however it can happen as a result of false PRI estimation. Then the performance index is limited to 1. The OEE index is a product of three indexes:

$$OEE = A * E * Q = \frac{PRI * (PQ - RQ)}{BPT} \tag{2}$$

Assuming the available experiment's data is sufficient for the OEE index calculation.

The last step of the source data analysis is to determine the event data subset for the simulation. Every event data contains the code of a machine and its timestamp. It allows to connect the machine's event data with a product's events. The number of products' completion events versus production unit is presented in Fig. 3(a). To achieve better readability, the machines with the number of such events less than 20 were filtered out. From the plot, it can be noticed that four machines emit more events: 0005, 5309-1, 5309-2, 8105. The number of a production machine's work time events is presented in Fig. 3(b). It can be noticed that the machines highlighted above, except for 0005, have also a lot of work time events.

(a) *Products' completion events*



(b) *Work time events*

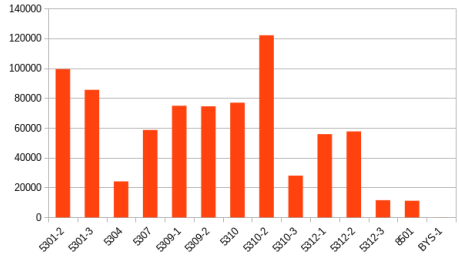


Fig. 3. Distribution versus production unit

Assuming the three machines highlighted above (5309-1, 5309-2, 8105) were chosen for the simulation, as those were the only ones that had a high number of events of both types. To determine the time range of events for the simulation, the number of both types of events generated by the chosen machines was analyzed versus months. The results are presented in Fig. 4(a) and Fig. 4(b).

To use the largest unit of data, the following time range was selected from 2018-09-17 to 2019-09-16. It can be noticed that there are months when no

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_38

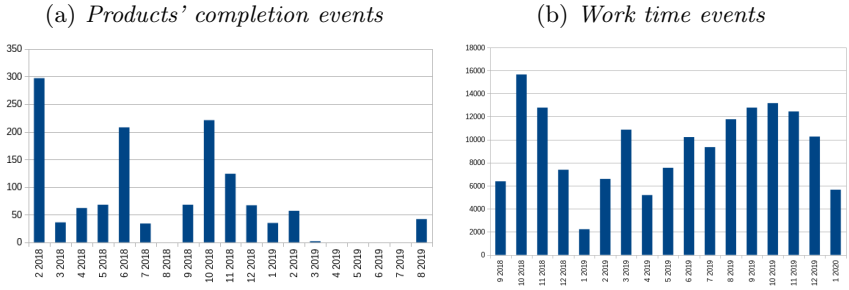


Fig. 4. Distribution versus months

products' completions were registered. It may be because employees refrain from registering the events in the system terminals. In such cases, it is impossible to calculate the quality, performance and the OEE indexes, but the availability index is still possible to calculate. Additionally, considering a low number of products' completion events in general, the indexes will be calculated versus days.

3.1 KPI Indexes Calculation by the Legacy Method

To verify the Tweeting Factory concept, the KPI metrics were calculated offline using SQL query. The well-known considerations and limitations were applied to the legacy method.

3.2 A New Data Acquisition Method

Considering that the available data set does already contain the production information of units and entered by employees, only OEE index acquisition is considered. The available production data is further considered as production events (employee or machine). Following Tweeting Factory architecture, the events are transmitted by the message bus between the machines and the services that subscribe and publish the events. For the needs of the simulation, the assumption was made that the system is publishing the data on the message bus instead of storing it. The following services were introduced to the designed environment:

- a service transforming the machines' events,
- a service calculating the KPI indexes in real time,
- a service storing the KPI indexes into a database.

The designed architecture is presented in Fig. 5.
Having the Tweeting Factory not specifying the message formats [12], JSON format is applied in the architecture. The base message format includes: start, end, machine, e_type, and payload. The e_type and payload field values are specific to the registered event. The e_type field is the key value

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_38

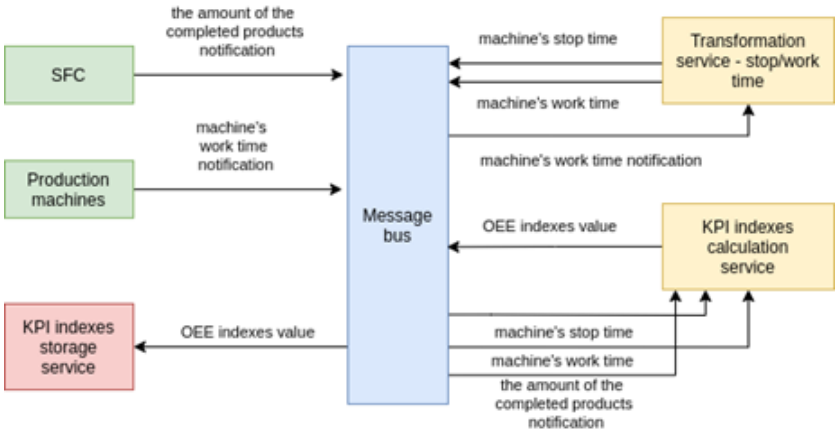


Fig. 5. The Tweeting Factory architecture proposal

of the registered event. The payload field contains the data specific for the event:

- the number of completed product notifications: QtyAll – an amount of produced products, QtyRejected – an amount of rejected products, RefTime – an estimated reference unit time for a product,
- work time notification: work – 0 for stopped, 1 for running
- work time: time – work time in seconds
- stop time: time – stop time in seconds
- OEE indexes value: pq – an amount of all produced products, rq – an amount of all rejected products, pri_pq - the product of of manufactured products and planned working time per unit, in seconds, apt – machine's work time in seconds, pbt – machine's estimated work time in seconds, quality – quality index value, - efficiency – efficiency index value, availability – availability index value, oee – OEE index value.

3.3 Simulation Implementation

To adjust the design to the simulation conditions, the following extensions were made:

- the source events generated originally by the system and the machines are emitted by the simulation program,
- in order to shorten the simulation time (365 days of the chosen time range), the simulation clock event is introduced,
- start and end events indicating the whole simulation start and end time are introduced,
- in order to measure a KPI calculation time, the message's metadata were extended by a new parameter trace_id, being an unique message's id and a new service calculating the time.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_38

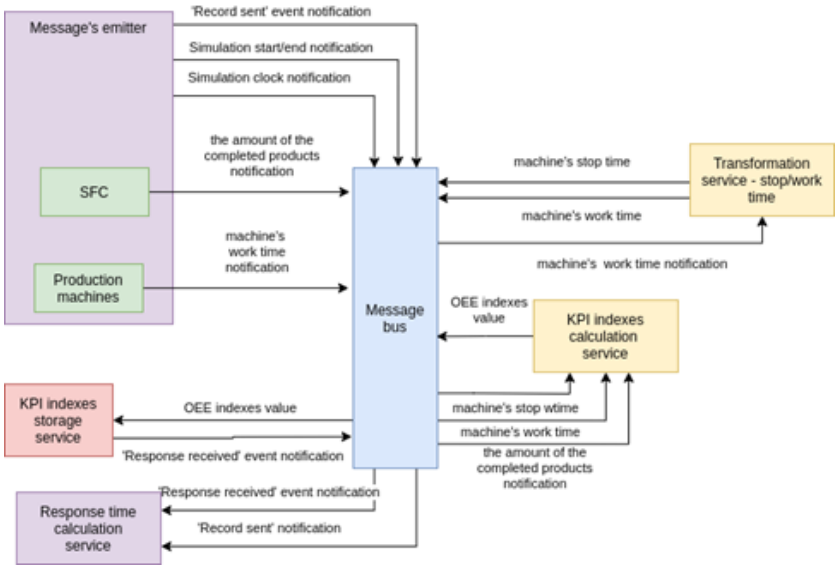


Fig. 6. Simulation environment's extended architecture

The extensions implemented on the architecture are presented in Fig. 6. The Tweeting Factory concept does not specify any software for the message bus [12] implementation. The authors decided to use RabbitMQ [3]. The main advantage of the software over the Apache ActiveMQ [4] is ease of use and an extensive documentation. All services are implemented using Python3 and the Pika library [5] that enables a communication with the RabbitMQ server over the AMQP 0.9.1 protocol. Additionally, a CSV file with the input data for the simulation messages emitter was prepared. The CSV contains an array of source events, sorted by the publication time. A new type of messages were added in the extended simulation architecture:

- 'record sent' event notification: real_time – the event's timestamp, id – id for time measurement,
- 'response received' event notification: real_time – the event's timestamp, id – id for time measurement,
- simulation clock notification: clock – simulation timestamp,
- simulation start notification,
- simulation end notification.

3.4 Simulation Run

The run was performed on a PC equipped with AMD Ryzen 5 3600 processor unit and 16GB 3200MHz RAM memory. The message bus component was

[3] https://www.rabbitmq.com/
[4] https://activemq.apache.org/
[5] https://pypi.org/project/pika/

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_38

implemented in the Podman container [6] and docker.io/library/rabbitmq:3.8.4-management container image was used.
The average KPI indexes for the machines are presented in Table 1.

Table 1. The average KPI indexes for the production units

Unit	Quality	Efficiency	Availability	OEE
(a)	(b)	(c)	(d)	(e)
5309-1	0.9988	0.4625	0.4644	0.2657
5309-1	0.9949	0.5431	0.4087	0.2543
8105	0.9884	0.7952	0.6920	0.5857

Column (a) shows the symbolic codes of the selected production units. In column (b) the quality value is placed, while in the next two columns there are efficiency (c) and availability values (d). The last column (e) lists the total value of OEE.
The correctness verification was performed by the comparison of the Tweeting Factory approach results with the results of the legacy method. The following metrics were compared: PQ, RQ, PRI*PQ, APT, PBT, Quality, Efficiency, Availability, OEE. The comparisons have occurred 1095 times (365 days, 3 machines). No significant differences between the output results were found. The performance verification was done by the analysis of the data generated by the simulation indicating the waiting time for the KPI index calculation. The time was measured 113542 times. Average time was 0.012520 seconds, i.e., 12.52 milliseconds. In the case of about 99% of the results, the calculation time was less than 25 milliseconds. Fig. 7(a) presents the box plot of the data. Large number of outliers can be generated by the temporary high load of the KPI calculation service or the temporary spike of the processor demand of the operating system.
Anderson-Darling test [9] was performed to determine whether a normal distribution adequately describes a set of data. Significance level 0.05 was assumed. The following hypotheses were tested:

- H0: the results are normally distributed,
- H1: the results are not normally distributed.

The output p-value was $3.7x10^{-24}$, so the alternative hypothesis was accepted. Shapiro-Wilk test [17] was also performed to confirm the previous test output. The input data for the test was limited to 2000 records to keep the test's power. Significance level 0.05 was assumed. The following hypotheses were tested:

[6] https://podman.io/

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_38

– H0: the results are normally distributed.
– H1: the results are not normally distributed.

The output p-value was $6.923531x10^{-57}$, so the alternative hypothesis was accepted. Quantile distribution of the KPI indexed calculation time is presented in Fig. 7(b). The distribution is not linear, which means the distribution is not normal.

(a) *Box plot* (b) *Quantile normal distribution*

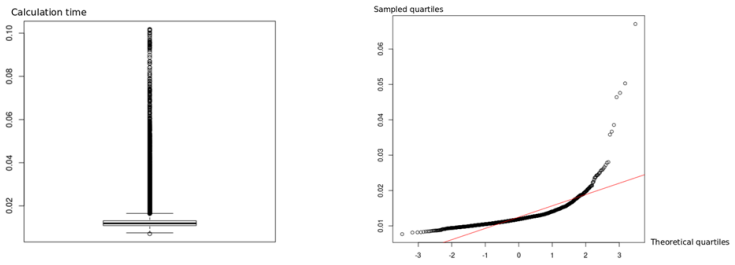


Fig. 7. KPI indexes calculation

To check if the calculated average value is a good representation of the random variable, the Wilcoxon test [37] was performed for one attempt. Significance level 0.05 was assumed. The following hypotheses were tested: - H0 – random variable's value's distribution is symmetric around = 0.012520. - H1 – random variable's value's distribution is not symmetric around = 0.012520. The output p-value was 0, so the alternative hypothesis was accepted. The median and percentile values are considered in further analysis. Statistical analysis showed no coincidence with the normal distribution of the computation time KPI indicators and the asymmetry of the time distribution with respect to the mean.

4 Conclusions

No differences between KPI index values calculated by the Tweeting Factory and the blackboard style method prove the Tweeting Factory concept correctness (RQ1). The response time of the proposed system that was shorter than 17ms in more than 95% cases puts the system among the real-time systems (RQ2). Considering the above two conclusions, the Tweeting Factory becomes a great candidate for a replacement of the SQL legacy method (RQ4), providing a lot of advantages (RQ3):

– ease of new production units and service integration,
– capability to add and remove new components without a need for reconfiguration,

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_38

– control over the messages flow,
– ESB controls data exchange security,
– complex metrics calculation in real-time,
– support to integrate the system with external systems.

There are also cons that need to be taken into account when considering the concept:

– data transformations required for the components using unsupported protocols,
– single point of failure – ESB,
– security of data exchange under the sole control of the message broker.

The literature analysis demonstrated a low number of concept applications in the industry. Taking the experiment's promising results into account the authors state that the lack of real life application studies is a major gap that may be a subject for the further research. Considering the literature survey conclusion that the Tweeting Factory concept can be used as a Digital Twin core component, the Digital Twin with Tweeting Factory application in Shared Industry [14]. Leveraging the real-time feature of the concept, another interesting research subject could be managed and optimized using machine learning algorithms.

Acknowledgment

Part of the work presented in this paper was received financial support from the statutory funds at the Wrocław University of Science and Technology and DSR Ltd.

References

1. Çetiner, G., Ismail, A., Hassan, A.: Ontology of manufacturing engineering. In: 5th International Advanced Technologies Symposium. p. 6 (2009)
2. De Ron, A., Rooda, J.: Equipment effectiveness: OEE revisited. IEEE Transactions on Semiconductor Manuf. **18**(1), 190–196 (2005)
3. Dressler, N.: Towards The Tweeting Factory. Master's thesis, KTH Industrial Engineering and Management, SE-100 44 Stockholm (2015)
4. Feeney, A.: The step modular architecture. Journal of Computing and Information Science in Engineering **2**(2), 132–135 (2002)
5. Gittler, T., Gontarz, A., Weiss, L., Wegener, K.: A fundamental approach for data acquisition on machine tools as enabler for analytical industrie 4.0 applications. Procedia CIRP **79**, 586–591 (2019)
6. Hoffmann, M.: Smart Agents for the Industry 4.0. Springer Vieweg, 1 edn. (2019)
7. Kos, T., Kosar, T., Mernik, M.: Development of data acquisition systems by using a domain-specific modeling language. Comput. Ind. **63**(3), 181–192 (Apr 2012)
8. Lennartson, B., Bengtsson, K., Wigström, O., Riazi, S.: Modeling and optimization of hybrid systems for the tweeting factory. IEEE Transactions on Automation Science and Engineering **13**(1), 191–205 (2016)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_38

9. Nelson, L.: The Anderson-Darling test for normality. Journal of Quality Technology **30**(3), 298–299 (1998)
10. Schütze, A., Helwig, N., Schneider, T.: Sensors 4.0 – smart sensors and measurement technology enable industry 4.0. Journal of Sensors and Sensor Systems **7**(1), 359–371 (2018)
11. Theorin, A., Bengtsson, K., Provost, J., Lieder, M., Johnsson, C., Lundholm, T., Lennartson, B.: An event-driven manufacturing information system architecture. IFAC-PapersOnLine **48**(3), 547–554 (2015)
12. Theorin, A., Bengtsson, K., Provost, J., Lieder, M., Johnsson, C., Lundholm, T., Lennartson, B.: An event-driven manufacturing information system architecture for industry 4.0. International Journal of Production Research **55**(5), 1297–1311 (2017)
13. Tursi, A.: Ontology-Based approach for Product-Driven interoperability of enterprise production systems. Phd thesis, Université Henri Poincaré - Nancy 1, Politecnico di Bari (2009)
14. Wang, G., Zhang, G., Guo, X., Zhang, Y.: Digital twin-driven service model and optimal allocation of manufacturing resources in shared manufacturing. Journal of Manufacturing Systems **59**, 165–179 (2021)
15. Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging. Addison-Wesley, 1 edn. (2003)
16. Yan, X.: Knowledge Acquisition from Streaming Data through a Novel Dynamic Clustering Algorithm. Phd thesis, North Carolina Agricultural and Technical State University (2018)
17. Zhao, L., Chuang, Z., Ke-Fu, X., Meng-Meng, C.: A computing model for real-time stream processing. In: 2014 International Conference on Cloud Computing and Big Data. pp. 134–137 (2014)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08757-8_38