An approach for implementing key performance indicators of a discrete manufacturing simulator based on the ISO 22400 standard

Usman Muhammad, Borja Ramis Ferrer, Wael M.Mohammed and Jose L.Martinez Lastra
FAST-Laboratory
Tampere University of Technology
PO Box 600, 33101 Tampere
{muhammad.2.usman, borja.ramisferrer, wael.mohammed, jose.lastra}@tut.fi

***Abstract*—Performance measurement tools and techniques have become very significant in today's industries for increasing the efficiency of their processes in order to face the competitive market. The first step towards performance measurement is the real-time monitoring and gathering of the data from the manufacturing system. Applying these performance measurement techniques on real- world industry in a way that is more general and efficient is the next challenge. This paper presents a methodology for implementing the key performance indicators defined in the ISO 22400 standard-Automation systems and integration, Key performance indicators (KPIs) for manufacturing operations management. The proposed methodology is implemented on a multi robot line simulator for measuring its performance at runtime. The approach implements a knowledge-based system within an ontology model which describes the environment, the system and the KPIs. In fact, the KPIs semantic descriptions are based on the data models presented in the Key Performance Indicators Markup Language (KPIML), which is an XML implementation of models developed by the Manufacturing Enterprise Solutions Association (MESA) international organization.**

Keywords—key performance indicators; ISO 22400 standard; manufacturing systems; knowledge-based system; ontology.

I. INTRODUCTION

Over the last decades, the monitoring of process performance has been one of the core focus in manufacturing enterprises in order to detect parts of industrial systems that may be improved for enhancing the overall production. This interest is largely due to growing competition, and to remain in competitive marketplace organizations must improve continuously. Such continuous improvement is bound to continuous monitoring and performance measurement at every level of organizations. The competitive markets forced the industry to adopt lean manufacturing shifting from the traditional mass production strategies, requiring high production efficiency and flexible production [1]. However, the focus of performance measurements remains yet on the business level of the organization that is more concerned with the financial status, marketing, and accounting processes [2]. But with the emergence of small and medium-sized enterprises these financial performance measurements are not enough, rather performance measurement at every level of the organization is required to compete in the market. Therefore, a shift in focus toward performance measurements of production level is observed [1]. Production level involves activities related to production scheduling, allocation of resources, maintenance operations, quality operations, inventory management and data collection [3], which can be done with technologies aligned with the fourth industrial revolution, or Industry 4.0, i.e., interconnectivity of production resources.

The manufacturing systems have become complex and distributed with time thus resulting in large amount of raw data, which must be collected and processed in real-time making the situation more complicated [4]. In result, manufacturing organizations have majorly focused on the key performance indicators that are real success drivers in today's world. Different organizations used different techniques and methodologies for identifying the KPIs for their system. One such methodology proposed in [3] is an 8-step iterative closed-loop model for defining and measuring KPIs in an industrial environment. However, these KPIs vary from one organization to another in terms of their terminologies used and interpretation thus creating a challenge of mutual communication and understanding. Moreover, different terminologies are used for same things at different hierarchical levels in the same organization. ISO 22400 standards solve this problem to the extent of Manufacturing Operations Management (MOM) level by defining a set of terminologies, concepts. And more importantly, the ISO 22400 standard identify 34 KPIs at MOM level, which can generally be used in any manufacturing system [5]. Implementation of these KPIs on a real-world manufacturing industry is another challenging task that the organizations face now. The Manufacturing Enterprise Solutions Association (MESA), an international organization developed a markup language by the name of KPIML, which presents the data model given for the KPIs in the ISO 22400 standards in an XML format [6].

This paper presents an implementation based on the ISO 22400 standard KPIs in a discrete manufacturing environment. The paper first presents an approach to implement and calculate KPIs for the manufacturing system and then implements the presented approach on a testbed for a set selected KPIs from the ISO 22400 standard. . A knowledge-based system has been designed and developed for modelling the manufacturing system as well as storing the information regarding the KPIs. Finally, a web-based simulator of a real industrial production line is used as a testbed of this research work.

The rest of the paper is structured as follows: Section II presents the related work and background knowledge used during the course of this research. Section III describes how the ontology model is created for this implementation and the overall architecture of the system. Section IV explains the testbed used in this research and describes how different KPIs are implemented on the testbed. Finally, Section V concludes the paper as well as presents the possibility for future research work.

II. STATE OF THE ART

A. Existing KPI based Solutions

As performance measurement in industry grasped the attention of managers and researchers, many performance measurement frameworks and systems can be found. For example, [7] discusses a Process Performance Measurement Systems (PPMS) that divides performance measurement systems into two different levels. Firstly, the individual performance measures level is focused on measuring the performance separately in relation to time, cost, and flexibility. On the other hand, the performance measurement system level, as an entity, consists of a set of performance indicators that contribute towards the effectiveness and efficiency of an industry as a whole.

In addition, efforts have been done to design a general framework that helps in identifying a set of key performance indicators for the industrial environment. In [3], Rakar et al. designed an 8-step closed-loop iterative model based on the finding of Bennett in [8] for deriving key performance indicators for the process industry. This iterative closed-loop model helps in continuously monitoring the performance of the industry with the help of the selected indicators and at the end of each cycle, it drops or adds new performance indicators depending on the importance and usability of a specific performance indicator. Figure 1 shows the eight steps involved in the closed loop iterative model.

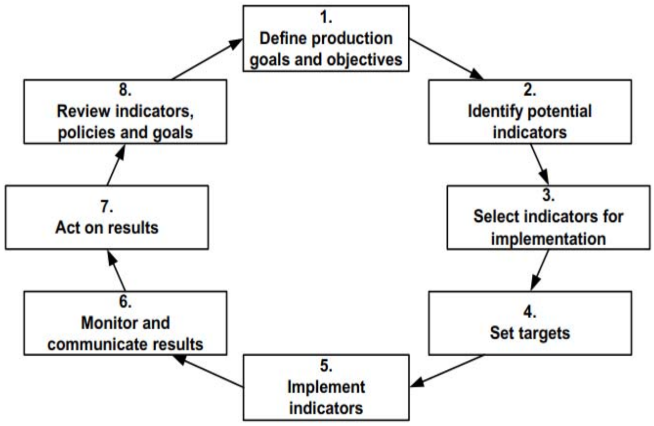

Figure 1: 8-step iterative closed loop model for deriving KPIs [3]

Furthermore, the authors in [3], managed to identify five categories of KPIs with the help of the above-mentioned closed loop iterative model. These indicators are i) Safety and environment, ii) Efficiency, iii) Quality, iv) Production plan tracking, and v) Issues related to employees. The major focus of the research in this field remains in the process of identifying and deriving key performance indicators for the industry rather than their implementation on the real-world industry.

B. ISO 22400 standard and KPIML

Performance measuring indicators keeps varying from one industry to another. Therefore, it becomes challenging for any research organization to develop a generic set of key performance indicators that support different kinds of manufacturing industries. The ISO 22400 standards, "Automation systems and integration — Key performance indicators (KPIs) for manufacturing operations management" identifies and defines up to 34 KPIs. These KPIs are calculated at MOM level of the industry, it requires some parameters from the lower levels such as control systems. After calculation, these KPIs are sent to business planning and logistics level for further usage and decision-making.

The ISO 22400 standards consist of two parts:

1. ISO 22400-1: Overview, concepts and terminology
2. ISO 22400-2: Definitions and descriptions

The ISO-22400-1, consist of the overview and basic concepts of the KPIs framework in the industry as well defines the specific terminology that is used in designing a KPI. The ISO-22400-2, compiles a list of 34 KPIs that are can be used in manufacturing industry with their definitions, scopes, formulas and a complete description for each KPI.

C. Web Services and Hypermedia

Due to the advances in the Internet and Computer Technologies (ICT) domain, the implementation of the web applications became more mature. In general, web applications require several parts to work hand in hand to gather, e.g., web services. For instance, they allow a standardized communication between applications [9]. As an example, SOA (Service Oriented Architecture) is a paradigm for implementing communication definitions within the implementation of three elements; service broker, service provider and service consumer. Based on this, several protocols have been designed such as HTTP (Hypertext Transfer Protocol), MQTT and OPC-UA (Open platform Communication-Unified Architecture) among others [10], [11]. In addition, some implementations are built using these protocols, e.g., SOAP (Simple Object Access Protocol) or WS (WebSocket) protocol. Besides, although it is not a standard, RESTful web services can be considered as a main example of implementation of web services based on HTTP protocol. In this context, RESTful web services represent a stateless request-response communication approach. In fact, RESTful web services employ the concept of Create, Read, Update and Delete (CRUD) by using different HTTP request methods, mapped to GET, POST, PUT and DELETE [12]. In regard of usage in industrial cases, RESTful web services have been included in many approaches as discussed in [13]–[17].

Besides web services, another important part regarding the web applications includes interfaces and hypermedia protocols. In this context, several standards, APIs (Application Programming Interface) and programming languages have been developed and refined in order to provide an interactive interface of web applications. For instance, HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), SVG (Scalable Vector Graphics) and JavaScript. Moreover, organizations create higher-level of frameworks that provide APIs which are built on the top of the mentioned standards like BootStarp3 and AngularJS [18].

D. Knowledge Based Systems

In the previous sections' context, i.e., the emergence of ICT and web-based technologies, the design and deployment of knowledge-based solutions in the industry is increasing [19]. Commonly, a Knowledge Base (KB) is employed as a centric engineering artifact that contains descriptions of the system to be controlled and/or monitored and the surrounding environment [20].

Knowledge repositories can be implemented within ontologies [21]. There are many options for designing and implementing ontologies [22], such as the Ontology Web Language (OWL) [23], which is a based on the RDF[1]. In any case, both humans and machines can access the semantic model at runtime.

Ontologies that are written using OWL syntax can be queried through RDF-based queries. These can be of different types depending on the purpose of the user. For example, knowledge descriptions can be retrieved within ASK, SELECT or CONSTRUCT SPARQL queries [24] or even updated by executing INSERTE, DELETE or MODIFY SPARQL Update queries [25].

Furthermore, the explicit statements included in ontologies can lead to implicit knowledge within the use of reasoning engines, or reasoners. These entities evaluate and validate the links between semantic resources in order to infer new relationships, not envisioned at the design phase and/or possible during runtime. To support reasoners to knowledge inference, semantic rules, e.g., Semantic Web Rule Language (SWRL) rules may be added on top of the ontological statements [26].

III. Approach

This section will present the methodology adopted along the course of this paper to implement the KPIs. The section will describe the knowledge base designed for this implementation, and other components used in the architecture followed in this implementation.

A. Architectural view and components of the approach

The architectural view diagram presented in **Figure 2**, illustrates all the components involved in the approach of this research work. The architecture is formed by five basic building blocks: the Knowledge based System, the Manufacturing Plant, the Orchestration Engine, the KPI Implementation Component and the KPI Visualization [27]. Each component is described below.

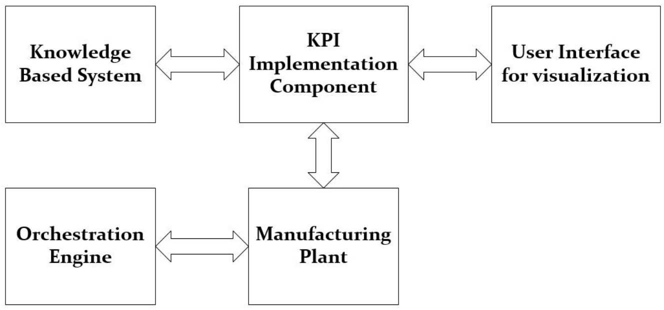

Figure 2: Main components of the approach [27]

1. Knowledge based System:

The Knowledge based System is built by two main components, i.e., an ontology model and a web server, which hosts the ontology and provides access to it through specific interface. More precisely, the ontology is hosted by a *Jena Fuseki*[2] server that permits the retrieval and update of ontological knowledge within its REST interface.

To illustrate the model that could be implemented for describing manufacturing systems and the KPIs to be used for monitoring their performance, following Figure 3 presents the main classes of the ontology designed for this research work.

[1] https://www.w3.org/RDF/

[2] https://jena.apache.org/documentation/fuseki2/

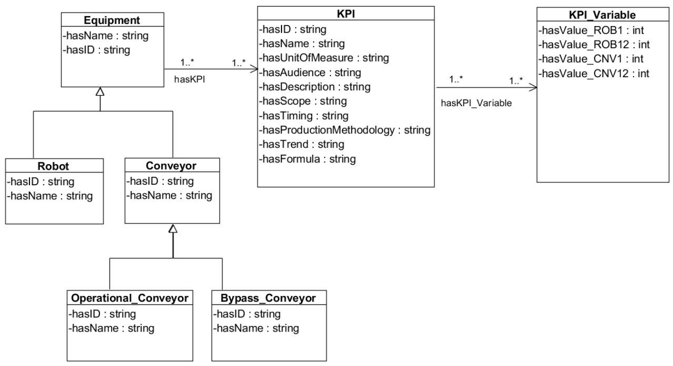

Figure 3: UML class diagram of the ontology [27]

As it can be seen in previous diagram, there are three main classes: KPI, KPI_Variable and Equipment which, in turn, is a higher class of industrial equipment. First, KPI class describes the set of KPIs that are used for monitoring the performance of discrete processes. Further, the KPI_Variable includes all the variables that are needed for each KPI equations. Finally, the equipment class and subclasses describe the components of the manufacturing system to perform physical operations.

It should be noted that as this model has been used for describing a specific assembly line, described in following section, the subclasses of Equipment are based on such use case. Nevertheless, for different scenario, this part of the model can be changed accordingly. In fact, this is one of the advantages of the ontology as it can be reused and adapted to the system to be described.

2. Manufacturing Plant:

Manufacturing plant is the main building block in this architecture diagram as this whole implementation will revolve around computing KPIs for the desired manufacturing plant. Manufacturing plants used raw materials or components and convert them in to useful finished goods with help of certain resources by performing a sequence of operations and tasks on them. The manufacturing plant will serve as the source of generating events and data in runtime, that will be utilized in the calculation of KPIs and populating the knowledge base. The manufacturing plant used as a test bed in this implementation is a multi-robot line for manufacturing mobile phones structures of different kind. The functionality and working of the testbed is demonstrated in the Implementation section.

3. Orchestration Engine:

The orchestration engine is used in this architecture to execute the production orders on the manufacturing plant. The user or production planner in the manufacturing plant specifies the production order in accordance with requirements from the customer. The production order contains the required recipe, quantity, and color of the product. The orchestrator engine executes such kind of production orders on the manufacturing plant as per requirements and specification by orchestrating the process flow. The design of orchestration engine can vary with the communication protocols of manufacturing system (i.e., SOAP vs REST) [28]. The orchestrator designed for this research work subscribes to all the events on the production line and thereafter, it executes all the needed tasks to be performed on each product.

4. KPI Implementation and calculation:

The KPI implementation component contains the formulas for the KPIs defined in the ISO 22400-2. It implements these formulas on the incoming data from the knowledge-based system. The KPI implementation component receives events notifications from the manufacturing plant in run time and based on those notifications it extracts useful information and data regarding the production line. The extracted data is then sent to knowledge-based system for updating the knowledge base. The KPI formulas are then applied in this component whenever the data is updated.

The updated KPI values are then sent to the user interface for visualization. The visualization of the results obtained from the implementing and calculating the KPIs in form of different visual and descriptive graphs will be demonstrated in further research work.

IV. Implementation

This section will present the implementation of the proposed approach in the previous section. In this section, a use case for the implementation will be introduced along with the complete description of how the applicable KPIs are implemented and visualized.

A. FASTory Simulator as Use Case

The use case of this research work for this implementation of KPIs is based on a multi-robot production line known as the FASTory line that is used for research purposes. This system is located at the Factory Automation Systems and Technology laboratory (FAST-Lab.), which belongs to the Tampere University of Technology. The FASTory line was used for assembling electronic devices and it was later modified for performing drawing operations in order to draw three different parts of mobile phones: frame, screen and keyboard. This is shown in Figure 4.

Figure 4: FASTory production line

The FASTory production line consists of 12 workstations out of which 10 are identical workstations that have a SCARA robot for drawing the three parts of the mobile phone. On the other hand, Workstation 1 (WS1) is used for loading/storing empty pallets and another workstation (WS7) for loading and unloading paper on the pallets. The different colored drawing on the paper is the representation of different components of the mobile phone. The line is capable of making 3 different recipes for each component in 3 different colors that makes a total of 729 different product variations. Each drawing workstation have two set of conveyors, a main conveyor that leads the pallet to the robot for drawing operations and one bypass conveyor that is used in case the workstation is busy or there is no operation to be performed. Besides, there are three types of sensors on each work station, a presence sensor to detect the presence of the pallet, an RFID reader at the entrance of each workstation for pallet recognition and a stopper at each zone to stop the pallet in that zone. The FASTory line has S1000 controllers installed at each workstation, which send notifications about any change on the line, these events notifications are exposed as web services.

To avoid the potential risks, mechanical and electrical problems related to production line and more specifically to reduce the setup time as well running cost, a simulation of the real production line was developed that imitates the real process occurring on the line. The FASTory simulator was developed as a part of eScop project[3]. The simulator is a webserver that provides interaction via RESTful services, moreover it provides a user interface for interaction as well as runtime view of the status of the simulator. The Figure 7 shows the simulator of the real production line. In this implementation, the FASTory simulator will be used as the test bed of this research work in order to avoid the aforementioned problems with the real production line.

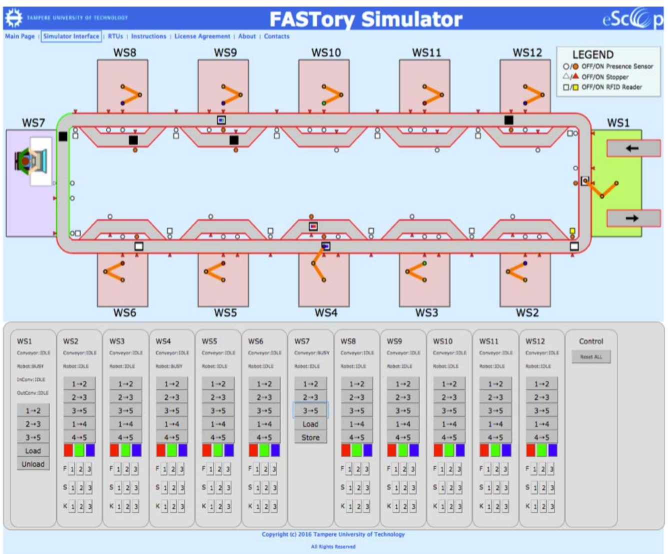

Figure 5: FASTory Simulator[4]

[3] http://escop-project.eu/

[4] http://escop.rd.tut.fi:3000/

B. *Model manipulation and KPI formulas calculation*

The approach described in the previous section is brought in to implementation to monitor the performance of the FASTory line. One of the principal reasons of selecting the FASTory simulator as the Manufacturing Plant block shown in Figure 2 is because it mimics a real manufacturing plant [29]. In other words, the simulator can be seen as a digital twin of the assembly line. For the knowledge-based system, an ontology model is used that act as an RDF store and stores the incoming data. The components for this proof of concept following the architectural view of Figure 2 are shown in Figure 6.

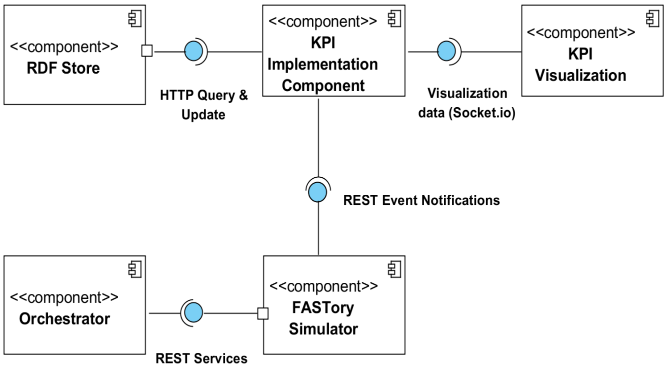

Figure 6: Interaction of components in an Architecture [27]

The KPI component in the architecture is the major component, which has three major tasks to perform. Firstly, it listens to the incoming event notification from the simulator interface. The communication between them is based on REST. More precisely, the simulator interface sends HTTP POST request whenever there is some change in the state of the simulator. These notifications include events related to the execution of robot drawing and conveyor transfer pallet operations. WS7 is a manual workstation, the operator loads and unloads pallet on it. The industrial controller assigned to WS7 sends Pallet Loaded or unloaded notification whenever any of these operations is performed. On the other hand, WS1 is dedicated for paper loading and unloading and it sends notification whenever it loads or unload a paper from the pallet. Furthermore, all other workstations send robot start or stop drawing, conveyor start or stop transferring and pen changed notifications depending on the operation that is performed by each workstation.

An example of one such notification is given below in Figure 7.


```
{ MSG: ‘RobotStartDrawing’,
  WS: ‘8’,
  PalletID: 1508488758200,
  Recipe: ‘1’,
  ServiceID: 1508488778004}
```


Figure 7: Event Notification received form FASTory Simulator

The *MSG* elements in the event notification received in JSON format, shows the type of operation that is occurring at the moment, the *WS* shows the workstation at which that operation is occurring. The *PalletID* and *ServiceID* are ids of the pallet at which the task is performed and of the service respectively. Lastly, *Recipe* element shows the recipe variation that is executed for that product.

Once the KPI component receives the event notification, it starts calculating the values for the set of variables involved in KPIs formulas. Some of major variables involved in this implementation are: Actual production time (APT), Actual unit busy time (AUBT), Produced quantity (PQ), Good quantity (GQ) and Scrap quantity (SQ). Beside these variables, which are calculated form the production line, some other variables are also involved that are planned by the production manager at the start of executing the production order. These planned variables are: Planned busy time (PBT), Planned operation time (POT) and Planned order execution time (POET). The following list of variables obtained from the ISO 22400-2 standard [30], shows how these variables are calculated.

- *Actual production time (APT)*: It is the actual time in which the workstation is adding some value to the final order. It is calculated as the time difference between *Robot start drawing* and *Robot stop drawing* events.
- *Actual unit busy time (AUBT)*: Actual unit busy time is the time when a workstation is busy. It includes the time when the robot is drawing as well as the time when the conveyor is transferring the pallet towards the drawing zone.
- *Produced quantity*: The produced quantity is the amount of quantity produced until that moment in time.
- *Good quantity*: The good quantity is considered as the quantity that meets the quality criteria.
- *Scrap quantity*: The scrap quantity is considered as the quantity that falls below the quality criteria.

Thereafter, the values of these KPI variables are updated in the RDF Store. The RDF store is populated via HTTP POST request that contains the data of these KPI variables. These KPI variables are stored in the RDF store with a timestamp, which helps in retrieving them by the order. After each update, the KPI component retrieve the new data from the RDF store with a query in a HTTP GET request. Following Figure 10 and Figure 11 are the examples of both update and retrieve data SPARQL query for one KPI variable respectively from the RDF store in this implementation.

```
PREFIX kpis:<http://www.semanticweb.org/KPI#>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
insert
{[]
   kpis:Kpi_Variable ?VariableName;
   kpis:hasValue_ROB1 ?Production_Time;
   kpis:hasTimestamp ?time
}
where
{
   values (?VariableName ?Production_Time)
       {(kpis:Actual_Production_Time
Working_Time)}
   bind (now() as ?time)
}
```

Figure 8: Update Query

```
PREFIX
kpis:<http://www.semanticweb.org/KPI#>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
SELECT ?VariableName ?Production_Time ?time
{[]
    kpis:Kpi_Variable ?VariableName;
    kpis:hasValue_ROB1 ?Production_Time;
    kpis:hasTimestamp ?time.
FILTER(?VariableName=kpis:Actual_Production
_Time)
}
order by ?time
```

Figure 9: Select Query to retrieve data from the RDF store

The KPI component after querying the updated data from the RDF store start executing the formulas for each of the KPIs. In this implementation only five KPIs are computed that are related to our production line and monitoring them will add a value to our system. The following five KPIs are calculated with the help of the given formulas in the ISO 22400-2 standards [30].

- Allocation Efficiency: The allocation efficiency is the ratio between the actual time a work unit is busy to the planned busy time estimated at the start of production shift.

$$Allocation\ Efficiency = \frac{Actual\ Unit\ Busy\ Time}{Planned\ Busy\ Time} \quad (1)$$

- Utilization efficiency: It is calculated as the ratio between the productive time a work unit is working to the actual time a work unit is busy.

$$Utilization\ Efficiency = \frac{Actual\ Production\ Time}{Actual\ Unit\ Busy\ Time} \quad (2)$$

- Availability: Availability shows the fraction of time a work unit is adding value with respect to the initially planned busy time for a work unit.

$$Availability = \frac{Actual\ Production\ Time}{Planned\ Busy\ Time} \quad (3)$$

- Quality Ratio: Quality Ratio is calculated as the ratio between good quantity that meets the quality criteria and total produced quantity.

$$Quality\ Ratio = \frac{Good\ Quantity}{Produced\ Quantity} \quad (4)$$

- Scrap Ratio: Scrap ratio is the inverse of quality ratio, it is the computed as the ratio between scrap quantity that didn't fulfil the quality criteria and total produced quantity.

$$Scrap\ Ratio = \frac{Scrap\ Quantity}{Produced\ Quantity} \quad (5)$$

Finally, the implementation presented in this paper will be tested and demonstrated for computing the value for each of the KPI with help of the above-mentioned formulas and then can be visualized in the future research work. Following is the sequence diagram of how and when each component interacts with each other to implement the proposed solution.

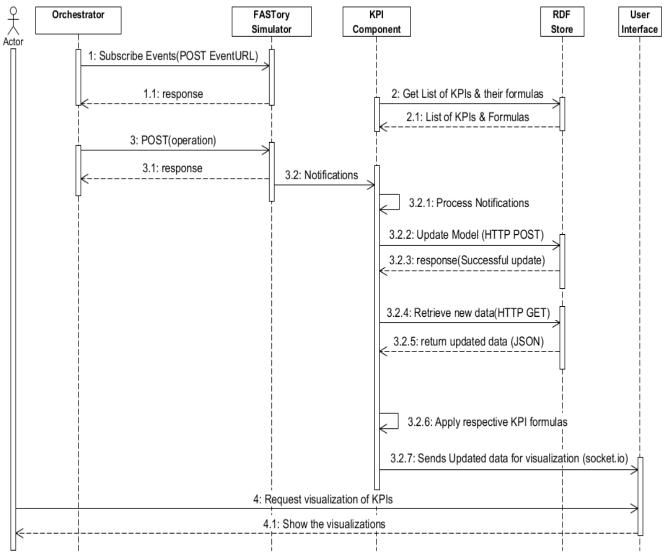

Figure 10: Sequence Diagram for components interaction [27]

The sequence diagram in Figure 10 is the complete depiction of the components interaction between each other. It shows that the process starts by the orchestrator after a production order is placed at it. As shown in the sequence diagram, the KPI component on reception of each event notification repeats the process of updating and querying the model, executing the KPI formulas and sending the updated data to the frontend for visualization.

V. Conclusion

This article presents an approach for implementing of a set of the recently designed KPIs in the ISO 22400-1 & 2 standards [5][30] on a real-world use case. In this context, the paper presents an architectural view that includes all the necessary components to calculate KPIs for manufacturing system. Then, the suggested architecture can be used as a blueprint for developing a system that permits KPI implementation and visualization tasks. The latter will be explored and demonstrated in further research work. On the other hand, this paper presents the FASTory simulator as the use case of the described approach. The paper make use of KPIML[6] to design an ontology for describing such KPIs. The authors claim that this research work serves as a paving step towards the implementation and visualization of the ISO 22400 standard KPIs in production industry for monitoring and decision-making.

Acknowledgment

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement n° 636909, correspondent to the project shortly entitled C2NET, Cloud Collaborative Manufacturing Networks.

References

[1] K. T. Gama and V. Cavenaghi, "Measuring performance and Lean Production: a review of literature and a proposal for a performance measurement system," in *Proceedings of the Production and Operation Management Society (POMS) 20th Annual Conference*, 2009.

[2] E. Ankrah and C. C. Y. Mensah, "Measuring Performance in Small and Medium Scale Enterprises in the Manufacturing Industry in Ghana," *Int. J.*, vol. 34, 2015.

[3] A. Rakar, S. Zorzut, and V. Jovan, "Assessment of Production Performance by Means of KPI," in *Control 2004*, University of Bath, UK, 2004.

[4] B. Zhang, C. Postelnicu, and J. L. M. Lastra, "Key Performance Indicators for energy efficient asset management in a factory automation testbed," in *IEEE 10th International Conference on Industrial Informatics*, 2012, pp. 391–396.

[5] "International Standard ISO 22400–1. Automation Systems and Integration – Key Performance Indicators (KPIs) for Manufacturing Operations Management - Part 1: Overview, Concepts and Terminology." Geneva: International Standard Organization (ISO), 2014.

[6] "Key Performance Indicator Markup Language, Version 01." MESA International, May-2015.

[7] L. Milanović Glavan, "Understanding Process Performance Measurement Systems," *Bus. Syst. Res.*, vol. 2, pp. 25–38, Jan. 2011.

[8] M. Bennett, P. James, and with Leon Klinkers, "Sustainable Measures - Evaluation and Reporting of Environmental and Social Performance," *Int. J. Sustain. High. Educ.*, vol. 1, Apr. 2000.

[9] W. M. Mohammed, A. Lobov, B. R. Ferrer, S. Iarovyi, and J. L. M. Lastra, "A web-based simulator for a discrete manufacturing system," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 6583–6589.

[10] C. Martin *et al.*, "Integrated design of Human-Machine Interfaces for production plants," in *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, 2015, pp. 1–6.

[11] H. Derhamy, J. Rönnholm, J. Delsing, J. Eliasson, and J. van Deventer, "Protocol interoperability of OPC UA in service oriented architectures," in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, 2017, pp. 44–50.

[12] Z. Yun, L. Junjie, C. Ji, and W. Hua, "A framework research of power distribution equipment condition monitoring cloud platform based on RESTful web service," in *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*, 2017, pp. 1–6.

[13] L. E. Nugroho, A. Azis, I. W. Mustika, and Selo, "Development of RESTful API to support the oil palm plantation monitoring system," in *2017 7th International Annual Engineering Seminar (InAES)*, 2017, pp. 1–5.

[14] B. D. Martino, A. Esposito, S. A. Maisto, and S. Nacchia, "A semantic IoT framework to support RESTful devices' API interoperability," in *2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)*, 2017, pp. 78–83.

[15] B. R. Ferrer, W. M. Mohammed, E. Chen, and J. L. M. Lastra, "Connecting web-based IoT devices to a cloud-based manufacturing platform," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017, pp. 8628–8633.

[16] S. Iarovyi, W. M. Mohammed, A. Lobov, B. R. Ferrer, and J. L. M. Lastra, "Cyber #x2013;Physical Systems for Open-Knowledge-Driven Manufacturing Execution Systems," *Proc. IEEE*, vol. 104, no. 5, pp. 1142–1154, May 2016.

[17] W. M. Mohammed *et al.*, “Generic platform for manufacturing execution system functions in knowledge-driven manufacturing systems,” *Int. J. Comput. Integr. Manuf.*, vol. 31, no. 3, pp. 262–274, Mar. 2018.
[18] B. R. Ferrer, W. M. Mohammed, A. Lobov, A. M. Galera, and J. L. M. Lastra, “Including human tasks as semantic resources in manufacturing ontology models,” in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017, pp. 3466–3473.
[19] B. R. Ferrer, W. M. Mohammed, and J. L. M. Lastra, “A solution for processing supply chain events within ontology-based descriptions,” in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 4877–4883.
[20] B. Ramis *et al.*, “Knowledge-based web service integration for industrial automation,” in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, 2014, pp. 733–739.
[21] T. R. Gruber, “Toward principles for the design of ontologies used for knowledge sharing,” in *International Journal of Human-Computer Studies*, 1993, pp. 907–928.
[22] D. Kalibatiene and O. Vasilecas, “Survey on Ontology Languages,” in *Perspectives in Business Informatics Research*, 2011, pp. 124–141.
[23] “OWL Web Ontology Language Reference.” [Online]. Available: https://www.w3.org/TR/owl-ref/. [Accessed: 01-Jan-2018].
[24] “SPARQL Query Language for RDF.” [Online]. Available: https://www.w3.org/TR/rdf-sparql-query/. [Accessed: 01-Jun-2017].
[25] “SPARQL 1.1 Update.” [Online]. Available: https://www.w3.org/TR/sparql11-update/. [Accessed: 01-Apr-2017].
[26] “SWRL: A Semantic Web Rule Language Combining OWL and RuleML.” [Online]. Available: https://www.w3.org/Submission/SWRL/. [Accessed: 01-Nov-2017].
[27] M. Usman, “An Implementation of KPI-ML to a Multi-Robot Line Simulator,” Tampere University of Technology, 2017.
[28] W. M. Mohammed, “Encapsulation of MES Functionalities as RESTful Web Services for Knowledge-Driven Manufacturing Systems,” 2017.
[29] W. M. Mohammed, A. Lobov, B. R. Ferrer, S. Iarovyi, and J. L. M. Lastra, “A web-based simulator for a discrete manufacturing system,” in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 6583–6589.
[30] “International Standard ISO 22400–2. Automation Systems and Integration – Key Performance Indicators (KPIs) for Manufacturing Operations Management - Part 2: Definitions and descriptions.” Geneva: International Standard Organization (ISO), 2014.