Smart Shop-floor Monitoring via Manufacturing Blueprints and Complex-event Processing

Michalis Pingos[1], Amal Elgammal[2], Indika Kumara[3], Panayiotis Christodoulou[1], Mike P. Papazoglou[3] and Andreas S. Andreou[1]
[1]Department of Computer Engineering and Informatics, Cyprus University of Technology, Limassol, Cyprus
[2]Department of Computers & Information, Cairo University, Egypt
[3]Department of Economics and Management, Tilburg University, Netherlands

Keywords: Product-Service Systems, Smart Product Customization, Smart Shop-floor Monitoring, M2M, H2M.

Abstract: Nowadays, Product-Service-Systems (PSS) are being modernized into smart connected products that target to transform the industrial scenery and unlock unique prospects. This concept enforces a new technological heap and lifecycle models to support smart connected products. The intelligence that smart, connected products embed paves the way for more sophisticated data gathering and analytics capabilities ushering in tandem a new era of smarter supply and production chains, smarter production processes, and even end-to-end connected manufacturing ecosystems. The main contribution of this paper is a smart shop-floor monitoring framework and underpinning technological solutions, which enables the proactive identification and resolution of shop-floor distributions. The proposed monitoring framework is based on the synergy between the novel concept of Manufacturing Blueprints and Complex Event Processing (CEP) technologies, while it encompasses a middleware layer that enables loose coupling and adaptation in practice. The framework provides the basis for actionable PSS and production "intelligence" and facilitates a shift toward more fact-based manufacturing decisions. Implementation and validation of the proposed framework is performed through a real-world case study which demonstrates its applicability, and assesses the usability and efficiency of the proposed solutions.

1 INTRODUCTION

Nowadays, the trend in manufacturing calls for increased connectivity and more sophisticated data-gathering and analytics capabilities empowered by Cyber Physical Systems (CPS), big data technologies and the Internet of Things (IoT). These usher in tandem a new era of smarter supply and production chains, smarter production processes, and even end-to-end connected manufacturing ecosystems.

Manufacturing is trying to create a competitive advantage by not only offering products but accompanying them with services (Product-as-a-Service). Product-as-a-Service starts by sensor-based products that generate data in a continuous manner; these data can be utilized for delivering preventive and proactive maintenance. Product-as-a-Service often called Product/Service Systems (Bustinza et al., 2015).

However, the current state of practice of engineering PSSs still suffers from severe drawbacks (Elgammal et al., 2017; Papazoglou and Elgammal, 2017; Song, 2017). The most noticeable drawback is that PSS remains at conceptual level considering a marketing or business perspective and missing solid IT implementation. There is also complete lack of a common factory level vision to empower data sharing, monitoring and cross-correlation. In addition, PSS do not accommodate evolving user preferences or product differentiation features to enable effective customization. PSS are unable to capture a full view of products and services linking product structure with product quality, production processes and services. More importantly, they do not support analysis of product-related data gathered along product lifecycles to improve data-driven decision making.

This demands the use of novel lifecycle, techniques, and technologies to enable manufacturers to connect their data, processes, systems, personnel and equipment. The main contribution of this paper is a smart shop-floor monitoring framework with

Pingos, M., Elgammal, A., Kumara, I., Christodoulou, P., Papazoglou, M. and Andreou, A.
Smart Shop-floor Monitoring via Manufacturing Blueprints and Complex-event Processing.
DOI: 10.5220/0007720802290236
In *Proceedings of the 21st International Conference on Enterprise Information Systems (ICEIS 2019)*, pages 229-236
ISBN: 978-989-758-372-8

proactive capabilities, which enables the identification and resolution of execution disruptions. The proposed monitoring framework realizes the “Monitoring” process as a part of the novel smart PSS lifecycle previously introduced in (Papazoglou et al.,, 2018). The lifecycle provides a closed monitoring feedback loop that enables continuous product and service improvements on the basis of the novel concept of knowledge-intensive structures called Manufacturing Blueprints (Blueprints in short). The proposed framework is established on the basis of these tested structures, which semantically capture product-service and production-related knowledge (Papazoglou Elgammal, 2017; Papazoglou et al., 2015). Blueprints integrate dispersed manufacturing data from diverse sources and locations, which include and combine business transactional data and manufacturing operational data to gain full visibility and control, and provide the basis for production actionable “intelligence”. A a middleware layer is introduced that enables loose coupling with blueprints and adaptation of the proposed framework in practice. Furthermore, the framework utilizes CEP technology (Etzion and Niblett, 2010), the latter offering event processing which combines data from multiple sources to infer events or patterns that suggest more complicated circumstances. Implementation and validation of the proposed framework is demonstrated through a real-world case study sourcing from the H2020 ICP4Life EU Project, while the validation process assess the applicability, usability and efficacy of the proposed solutions.

The remainder of the paper is structured as follows: Section 2 discusses related work efforts in the areas of servitization and shop-floor monitoring. Section 3 presents the smart manufacturing framework and discusses its main components. This is followed by presenting the current implementation efforts in section 0. Finally, section 5 concludes the paper and highlights future work directions.

2 RELATED WORK

Related work efforts found in the literature are mostly focused on separately addressing aspects of the two converging research directions in this paper, namely Servitization and Shop-floor Monitoring. These are discussed in the next two sub-sections.

2.1 Servitization

Servitization is the innovation of an organization’s capabilities and processes to shift from selling products to selling integrated products and services that deliver value in use (Howard et al., 2013; Tim et al., 2017). Different approaches in the literature build on a distinction between products and services, and demonstrate how a change in the balance between these can result in different levels of servitization (Tim and Howard, 2013).

An approach with focus on value proposition that distinguishes between "base", "intermediate" and "advanced" services is proposed in (Howard et al., 2013; Tim and Howard, 2013). The base services focus on the product provision; intermediate services are based on exploitation of production competences to also maintain the condition of products; finally, the advanced services concentrate on the capability delivered through performance of the product (Howard et al., 2013; Tim and Howard, 2013).

Another frequently addressed approach for PSS classification proposes distinguishing between three main categories (Baines et al., 2007): (i) product-oriented, (ii) use-oriented, and (iii) result-oriented. In the product-oriented PSS, the product is offered in a traditional sale model, but also includes the sale of additional services (Baines et al., 2007). In the use-oriented and the result-oriented PSS, customer satisfaction is achieved by the functions provided by the products or the result of services rather than the product ownership (Chou et al., 2015).

Unfortunately, existing literature provides little or no guidance on how to successfully tackle the servitization challenges. Baines et al. (2009) discuss the scarcity of previous studies “that provide guidance, tools or techniques, that can be used by companies to servitize”, pointing out that “guidance in the literature on how to approach organizational strategy (for servitization) is largely limited to anecdotal evidence from case studies that suggest good practices and processes for implementation”. Sai et al. (2016) add to the discussion that most of the existing servitization studies remain at a theoretical level, limiting the applicability of the findings.

Unlike efforts in the literature, which lack concrete IT solutions to realize the vision of servitization (Howard et al., 2013; Tim et al., 2017), the proposed monitoring framework in this paper provides a native support to couple products and services for their continuous monitoring and improvement through a closed feedback loop.

2.2 Shop-floor Monitoring

Papazoglou et al. (2018) presented a novel PSS customized lifecycle approach that includes technological solutions aiming to enable PSS customization. The proposed methodology utilizes blueprints (Papazoglou and Elgammal, 2017; Papazoglou et al., 2015), which provide the root for actionable PSS and production intelligence. As previously mentioned, the proposed monitoring framework relies on blueprints as the basis of manufacturing intelligence; blueprints are briefly discussed in Section 3.

In addition, manufactures today are moving into a different direction that targets in fulfilling orders on demand by negotiating value-adding processes in real-time, taking at the same time into consideration quality, time, price etc. The growing demand of customized production results is considered a major challenge to traditional manufacturing businesses (Zhang et al., 2017). Wan et al. (2018) proposed a customized version of a Smart Factory for pharmaceutical manufacturing that was tested on a demand-based drug packing production. That work also introduced a Manufacturing's Semantics Ontology knowledge in the perception layer that aimed to plan the scheduling of the pharmaceutical production, thus the new plans are directly created from the production demand and the data collected from machines. Similarly, the work in (Zhang et al., 2017) outlined a framework of an intelligent shop-floor to allocate resources based on the production requirements. The proposed structure consists of three models: (i) the smart machine agent model, (ii) the self-organizing model and (iii) the self-adaptive model. A prototype cyber-physical system that includes the aforementioned models was developed to test the proposed methodology and assess the flexibility of configuring resources to deal disturbances.

According to (Theorin et al., 2015) future manufacturing systems must be flexible in order to adapt easily in the continuously changing market demands, but at the same time they need to make a better use of source data, thus low-level data should be transformed to real-time information for decision-making support. (Theorin et al., 2015) presented a Line Information System Architecture, called LISA, to enable factory integration and data exploitation. LISA is an event-driven framework with a prototype-oriented model which combines international standards and well-known off-the-shelf technologies aiming to be mechanically applicable. The work of (Christ, et al., 2016) introduced a different methodology based on Complex Event Processing (CEP), a technology to analyse event streams. The limitation of traditional CEP is that it cannot consider events that have not taken place yet, thus this paper introduces the concept of Conditional Event Occurrence Density Estimation (CEODE) to CEP. Christ et al. (2016) outline a structure for merging CEP engines with predictive analytics using CEODEs and demonstrates how CEP can change from a waiting process to predictive and prescriptive, to be able to deal with the production line challenges.

In a nutshell, this paper proposes a monitoring framework based on manufacturing blueprints and Complex Event Processing (CEP) technologies; to the best of our knowledge, it constitutes one of the very few, if any, initial studies that combine the aforementioned approaches under the same structure. The proposed framework aims to enable the proactive identification and resolution of shop-floor distributions in order to help businesses to connect their data, systems, equipment and personnel, developing at the same time a user-friendly environment for customers to customize products where resources are allocated based on the production requirements.

3 METHODOLOGY AND APPROACH

The approach of the present paper relies on a production planning/engineering middleware which is placed between the process of product design customization and the actual execution of the production steps performed at the shop-floor, following the novel smart product lifecycle introduced in (Papazoglou et al., 2018). The aim of the middleware is to provide the missing link connecting a conceptualized co-creation process based on which a user/customer designs the desired product using a graphical environment - for the purposes of this paper we assume that this design definition (product request) is already available and has been performed in the Unreal Engine environment (https://www.unrealengine.com) - with the processes and actions executed by machines during the actual making of the product.

In this context a series of steps are followed which start by transforming the product customization information inserted by the user into a standardized representation to facilitate comparison with existing knowledge stored in a repository as regards properties of the product to be developed and the sequence of operations at the machine level to actually construct it. To this end, the concept of Blueprints is adopted, to take advantage of their formal, standardized representation of product properties, events and workflows for the conceptual description of the details for building a product as analyzed earlier.

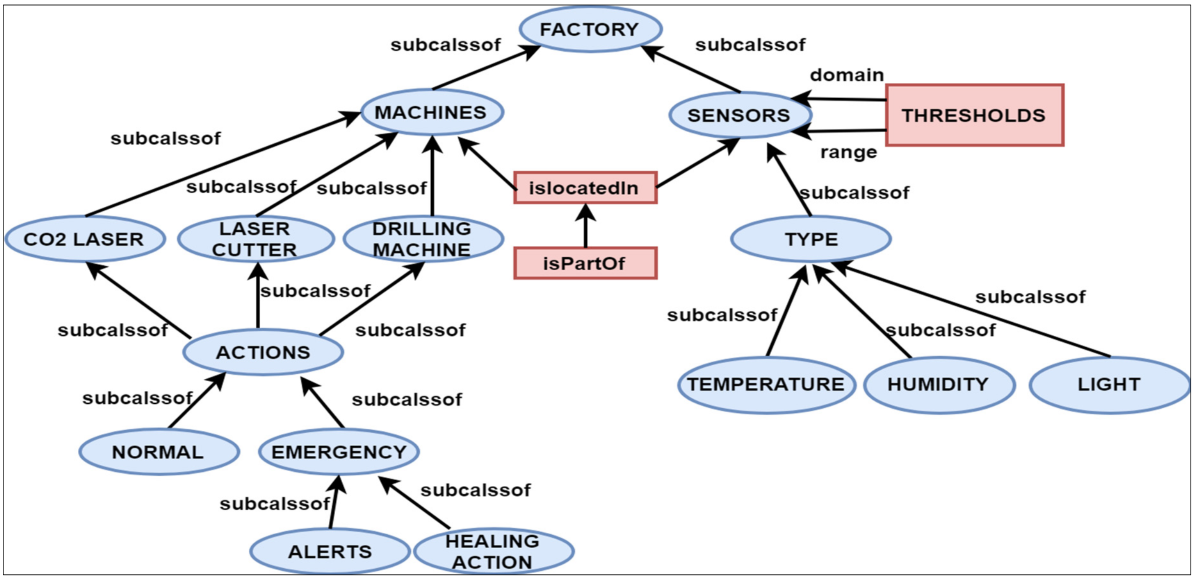

Figure 1: UML Meta-model OWL ontology in Protégé.

The existing knowledge, as expressed through Blueprints, essentially describes product and production related information, which consists of product events knowledge and product emergency events knowledge.

The latter is based on extending certain Blueprint types, namely Production Process Blueprint (PPB) and Production Service Blueprint (PSB), something which constitutes part of the novelty of the present paper. This extension to abovementioned Blueprints (see figure 1) provides refined details for the machines at the shop-floor involved in the specific product creation, and more specifically for the type, frequency of sampling (timing) and thresholds of sensors these machines include, as well as a list of actions to handle each emergency event according to sensor values (thresholds). Thresholds are defined by the control room operator or the shift manager, and in this work we consider these values as given.

In the next step the new knowledge is pushed to the shop-floor to complement and extend the normal workflow of machine actions throughout the process of building the product with actions that handle undesired cases (i.e., when alerts are raised). During this step the product request is compared with the existing knowledge stored and the closest match is used as the backbone to define refinements and revise the sequence of events for building the product and perform emergency actions. This revision creates a new Blueprint instance. The dedicated middleware receives this Blueprint instance and follows the cycle of normal execution and emergency actions described therein by translating them into a series of events that will take place at the shop-floor. To do so it parses and queries the RDF/OWL (https://www.w3.org/OWL/) images of the Blueprints involved to retrieve this information.

Focusing on the emergency actions, the middleware supports the monitoring and control of the execution of normal tasks by the machines and their actuators. The “product monitoring and actuating knowledge” is a process that is invoked in parallel with normal execution and monitors the threshold values in contrast with sensor readings so that in cases where anomalies are detected the proper actions are initiated, again in the form of events, as these were previously defined by the user (e.g. the shop-floor manager). This initiation process is handled by a tool called WSO2 (https://wso2.com/), which is essentially an event-driven framework that supports event-driven systems. Therefore, our approach follows the integration of WSO2 and Blueprints in terms of translating the conceptual Blueprint actions into actual steps/events executed through WSO2.

WSO2 is an open-source enterprise platform that enables integration of application programming interfaces (APIs), applications and web services both at local level and across the Internet. WSO2 offers a platform of middleware products for agile integration, API management, identity and access management, and smart analytics.

WSO2 essentially monitors in real-time the execution steps at the shop-floor by receiving real-time events. The detection of a violation of any of the defined monitoring rules (and/or thresholds), apart

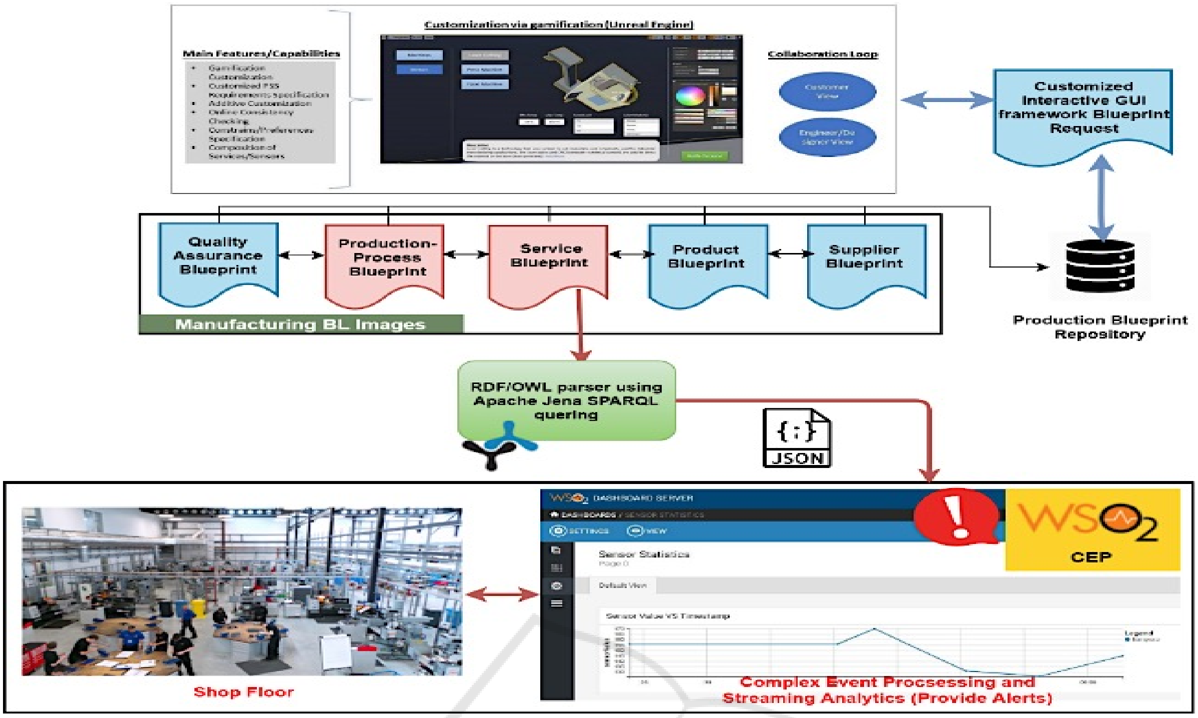

Figure 2: Functional representation of the proposed middleware

from the alerting process, fires the appropriate response action(s) defined in the corresponding rules of the Blueprints. An action in its simplest form could be the generation of an alert sent to the shop-floor manager, and in a more advanced case, the initiation of a series of signals/actions/controls passed directly to the shop-floor to instruct and drive the actuators of the machines to execute a process. For example, a possible action as a response to the detection of a rise in temperature for a welding machine (as compared to the predefined threshold), is to send a signal to the actuators at the shop-floor to turn on specific ventilation or air-conditioning machinery to cool the place and lower the temperature.

Figure 1 shows the extension of the service Blueprint in order to reveal alerts in WSO2 CEP tool by parsing the new RDF file of the extended Blueprint. Machine and sensors are part of a factory. Sensors is located in machines and are part of machines such as CO2 LASER, Laser Cutter and Drilling Machine. Machines perform normal actions in order to produce a requested product. If an abnormal scenario occurs, then emergency actions, such as alerts and healing actions, are executed. Certain types of sensors, such as for temperature, pressure, humidity etc., offer the ability to a set-up threshold values (e.g. min, max).

Figure 2 shows the whole concept of the workflow with the combination of the existing tools.

When a customer builds a product request in a GUI environment (this step is beyond the scope of the present paper and it is assumed to be available), the request is transformed into a Blueprint image. This request-image is then pushed to the middleware in order to be compared with the existing production Blueprint monitoring repository images. Shop-floor managers or engineers have already defined the thresholds of sensors on the machines which produce the requested product (e.g., Min Temperature: 50°C and Max Temperature: 250 °C). In order to push events and details of the whole production, normal operation and emergency cases, the latter being based on threshold values of the sensors at the shop-floor, PPB and PSB must first be parsed and then queried in order to obtain this information (see manufacturing Blueprint - BL images with light red color in figure 1 – second and third from left). PPB consists of Production Workflow (solution), Process Event and Data Collection, and Resources Devices and Equipment. PSB consists of Service Type, Service Sensors, Service Metrics and Service Schedule (Papazoglou and Elgammal 2017). What is actually executed is parsing and querying the images of these blueprints (PPB, PSB) expressed in RDF/OWL form.

An RDF data model is similar to classical conceptual modelling approaches (such as entity–relationship or class diagrams), but it provides a semantic support. It is based on the idea of making

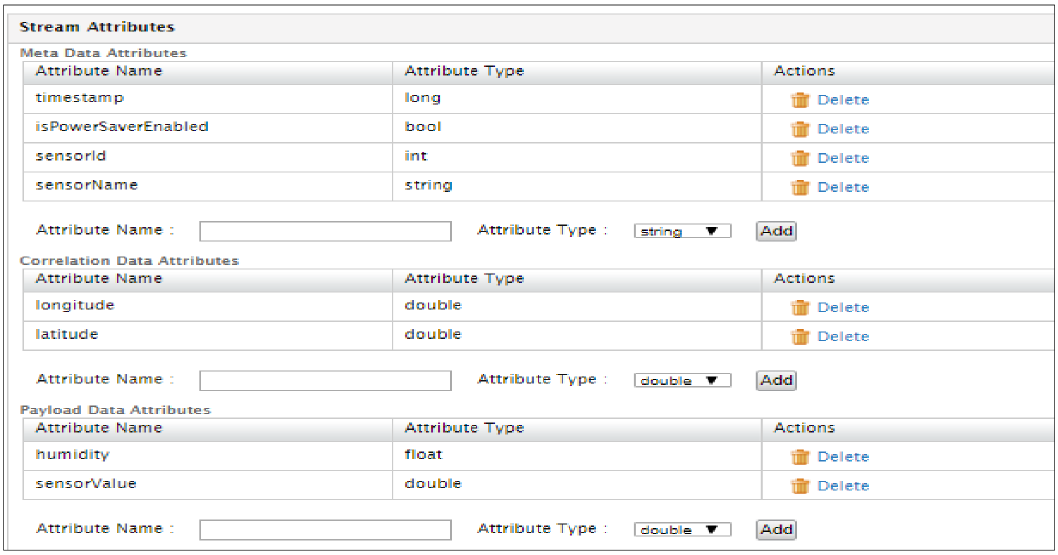

Figure 3: Stream example of the attributes of sensors.

statements about resources (in particular web resources) in expressions of the form *subject–predicate–object*, known as *triples*. The *subject* denotes the resource, while the *predicate* denotes traits or aspects of the resource, and expresses a relationship between the *subject* and the *object*. RDF is an abstract model with several serialization formats (i.e. file formats), so the particular encoding for resources or triples varies from format to format. A collection of RDF statements intrinsically represents a labelled, directed multi-graph. This in theory makes an RDF data model better suited to certain kinds of knowledge representation than other relational or ontological models. However, in practice, RDF data is often stored in relational database or native representations. OWL is a computational logic-based language such that knowledge expressed in OWL can be exploited by computer programs, e.g., to verify the consistency of that knowledge, or to make implicit knowledge explicit. OWL documents, known as ontologies, can be published in the World Wide Web and may refer to or be referred from other OWL ontologies. OWL is part of the W3C's Semantic Web technology stack, which includes RDF, RDFS, SPARQL, etc. (https://jena.apache.org/documentation/query/).

The current paper is not involved with data visualization; it deals with rules and implements the definition of the actions to be taken when emergency cases arise according to some threshold values set, while it assumes that the latter are already available as their definition is the subject of another research work by the authors (reference omitted for blind review and will be given upon acceptance).

To sum-up, when a request for building a product is received (see figure 2), the middleware, according to the request, firstly produces an extended production profile, which includes both the normal events that must be executed to produce the desired customized product and the emergency events, along with the threshold values of the sensors in each machine at the shop-floor that, when exceeded, trigger the execution of emergency actions. Secondly, it converts this profile into a series of events using WSO2 and transferring these events to the machines. At this stage it is also assumed that the details of the machines required for production, as well as the number and type of the sensors each machine includes, have already been defined and are available prior to the request sent to the middleware

4 IMPLEMENTATION AND DEMONSTRATION

This section presents a demonstration example where the methodology described in section 3 is applied on a real-world use-case. Firstly, an extension in the Blueprints is performed to refine details for the machines and the sensors at the shop-floor, as well as a list of alerts and actions (see table 1) that are used to handle each emergency event according to sensor values (thresholds). As presented in figure 3, in our demonstration example we used three types of machines, a CO2 Laser, a Laser Cutter and a Drilling Machine. In addition, to handle the emergency events, we have utilized various types of sensors that have been considered as an integral part of the machines, such as Temperature, Humidity and Light. Sensors were defined as presented in figure 3 and their real-time readings were presented as shown in figure 4.

As already mentioned in the methodology, upon occurrence of an abnormal event, emergency actions, alerts and/or healing actions that were defined by the shift manager upfront are executed/produced automatically.

Table 1: Demonstration alerts and actions.

Alerts	Actions
Temperature value overcame upper threshold	Turn on A/C
Humidity value overcame upper threshold	Stop the execution Switch-off machine

Following our methodology, the new knowledge acquired from the previous step is pushed to the shop-floor. This extends the usual workflow of machines and aids in the building of a customized product by creating a new Blueprint instance.


```
"event": {
    "metaData": {
        "timestamp": 4354643,
        "isPowerSaverEnabled": false,
        "sensorId": 701,
        "sensorName": temperature
    },
    "correlationData": {
        "longitude": 4.504343,
        "latitude": 20.44345
    },
    "payloadData": {
        "temperature": 55.8,
        "sensorValue": 57.2
    }
}
```


Figure 4: Real-time snapshot of sensors' readings.

```
Import java.util.Iterator;
import com.hp.hpl.jena.query .Query; .QueryFactory; .QueryExecution; .QueryExecutionFactory; // IMPORT APACHE JENA PACKAGE
import com.hp.hpl.jena.query .QuerySolution; .ResultSet; .ResultSetFormated; // IMPORT LIBRARY FOR SELECT QUERIES
import com.hp.hpl.jena.rdf.model .Literal; .Model; //IMPORT LIBRARY FOR RDF FILES
import com.hp.hpl.jena.util.FileManager; //IMPORT LIBRARY FOR RDF FILES
public class Main{public static void main(String args[]) {sparlqlQuering();}} // MAIN CLASS RESULT OF sparlqlQuering()
static void sparlqlQuering(){
        FileManager.get().addLocatorClassLoader(Main.class.getClassLoader());
        Model.model = FileManager.get().loadModel(services.rdf)
        Model.model = FileManager.get().loadModel(3rocess.rdf) // HANDLE THE BLUEPRINT IMAGES
        String queryString=
      " SELECT * WHERE MACHINE IS ACTIVE {
                - SENSORS THRESHOLDS OF ACTIVE FACTORY MACHINE
                - NORMAL EXECUTION OF ACTIVE FACTORY MACHINES
      - EMERGENCY ACTIONS OF ACTIVE FACTORY MACHINES(ALERTS, HEALING ACTIONS)
                }"
        SAVE RESULTS OF THE QUERY IN A JSON FILE
    } // QUERING AND SAVE RESULTS IN JSON FILE
```

Figure 5: JAVA pseudocode for handling and querying Blueprint images.


```
{
  "head": { "vars": [ "activemachine" , "sensor1","sensor2","sensor3" ]
  } ,
  "results": {
    "bindings": [
      {
        "activemachine": { "type": "uri" ,
        "value": "http://example.org/machines/machine6" } ,
        "sensor1": { "type": "literal" , "upper": "40",
        "lower": "0","emergency":"Turn ON A/C"},
        "sensor3": { "type": "literal" , "upper": "10",
        "lower": "5","emergency":"Turn ON A/C"}
      } ,
```


Figure 6: JSON results for the proposed example.

The proposed methodology utilizes Apache Jena to handle the Blueprint images and SPARQL for accessing Blueprint images. SPARQL takes the description in the form of a query, and returns that information, in the form of a set of bindings or an RDF graph. The JAVA code/pseudocode for handling and querying the Blueprint images of the demonstrated example is shown in figure 5 and the outcomes of the JSON file that were imported into WSO2 are presented in figure 6.

In summary, the proposed methodology supported the identification of the most meaningful events and patterns from multiple data sources, the analysis of their impact, and the decisions to be taken for resolving any problems occurred in real-time.

5 CONCLUSIONS AND FUTURE WORK

The present paper introduces a smart shop-floor monitoring framework that supports the identification and resolution of execution disruptions with proactive capabilities. The framework adopts a smart PSS lifecycle to offer a monitoring feedback loop that enables continuous product and service improvements based on knowledge-intensive structures called Manufacturing Blueprints. The latter integrates data from diverse sources and locations in the manufacturing environment and facilitate production actionable "intelligence". The middleware layer of the proposed framework connects it with the Blueprints offering at the same time adaptation capabilities in order to be fully operational in practice. In addition, the framework utilizes Complex Event Processing technology to combines data from the multiple sources present at the shop floor and infer events or patterns according to the circumstances. The proposed framework is being implemented and validated using a real-life case study demonstrating its applicability, usability and efficiency.

Future research is ongoing in a number of parallel and complementary directions, which include: (i) design and development of a user-friendly graphical domain-specific language to enable the product engineer/designer to specify and interpret monitoring rules in a user-friendly and intuitive manner, (ii) (semi-) automating recovery actions by seeding self-adaptiveness and self-healing capabilities, moving towards the vision of self-autonomous smart factory, and (iii) augmenting the dashboard with sophisticated visualization features by supporting augmented and virtual reality.

ACKNOWLEDGEMENTS

This paper is part of the outcomes of the Twinning project Dossier-Cloud. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 692251.

REFERENCES

Baines, T., Lightfoot, H. W., Evans, S., Neely, A., Greenough, R., Peppard, J., Wilson, H., 2007. State-of-the-art in product-service systems. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 221*(10).

Baines, T. S., Lightfoot, H. W., Benedettini, O., & Kay, J. M., 2009. The servitization of manufacturing: A review of literature and reflection on future challenges. *Journal of Manufacturing Technology Management.*

Bustinza, O. F., Bigdeli, A. Z., Baines, T., & Elliot, C., 2015. Servitization and Competitive Advantage: The Importance of Organizational Structure and Value Chain Position. *Research-Technology Management.*

Chou, C.-J., Chen, C.-W., & Conley, C., 2015. An approach to assessing sustainable product-service systems. *Journal of Cleaner Production, 86*, 277-284.

Christ, M., Krumeich, J., & Kempa-Liehr, A. W., 2016. *Integrating Predictive Analytics into Complex Event Processing by Using Conditional Density Estimations.* Paper presented at the 2016 IEEE 20th International Enterprise Distributed Object Computing Workshop (EDOCW).

Elgammal, A., Papazoglou, M., Krämer, B., & Constantinescu, C., 2017. *Design for Customization: A New Paradigm for Product-Service System Development.* Paper presented at the The 9th CIRP IPSS Conference: Circular Perspectives on Product/Service-Systems, Denmark.

Etzion, O., & Niblett, P., 2010. *Event Processing in Action*: Manning Publications Co.

Howard, L., Tim, B., & Palie, S., 2013. The servitization of manufacturing: A systematic literature review of interdependent trends. *International Journal of Operations & Production Management, 33*(11/12).

Papazoglou, M., & Elgammal, A., 2017. *The Manufacturing Blueprint Environment: Bringing Intelligence into Manufacturing.* Paper presented at the IEEE International Conference on Engineering, Technology and Innovation (ICE), Portugal.

Papazoglou, M., Heuvel, W. J. v. d., & Mascolo, J., 2015. Reference Architecture and Knowledge-based Structures for Smart Manufacturing Networks. *IEEE Software, PP*(99), 1-1.

Papazoglou, M. P., Elgammal, A., & Krämer, B. J., 2018. Collaborative on-demand Product-Service Systems customization lifecycle. *CIRP Journal of Manufacturing Science and Technology*.

Sai, S. N., David, L., Gillian, W., & Nick, Y., 2016. Eight challenges of servitisation for the configuration, measurement and management of organisations. *Journal of Service Theory and Practice, 26*(6).

Song, W. 2017., Requirement management for product-service systems: Status review and future trends. *Computers in Industry, 85*, 11-22.

Theorin, A., Bengtsson, K., Provost, J., Lieder, M., Johnsson, C., Lundholm, T., & Lennartson, B. 2015. ,An Event-Driven Manufacturing Information System Architecture. *IFAC-PapersOnLine, 48*(3), 547-554.

Tim, B., Ali, Z. B., F., B. O., Guang, S. V., James, B., & Keith, R., 2017. Servitization: revisiting the state-of-the-art and research priorities. *International Journal of Operations & Production Management, 37*(2).

Tim, B., & Howard, W. L., 2013. Servitization of the manufacturing firm: Exploring the operations practices and technologies that deliver advanced services. *International Journal of Operations & Production Management, 34*(1), 2-35.

Wan, J., Tang, S., Li, D., Imran, M., Zhang, C., Liu, C., & Pang, Z., 2018. Reconfigurable Smart Factory for Drug Packing in Healthcare Industry 4.0. *IEEE Transactions on Industrial Informatics*, 1-1.

Zhang, Y., Qian, C., Lv, J., & Liu, Y., 2017. Agent and Cyber-Physical System Based Self-Organizing and Self-Adaptive Intelligent Shopfloor. *IEEE Transactions on Industrial Informatics, 13*(2), 737-747.