Implementation of a Reference Architecture for Cyber Physical Systems to support Condition Based Maintenance

Felix Larrinaga[1], Javier Fernandez[1], Ekhi Zugasti[1], Iñaki Garitano[1], Urko Zurutuza[1], Mikel Anasagasti[2] and Mikel Mondragon[2]

***Abstract*— This paper presents the implementation of a reference architecture for Cyber Physical Systems (CPS) to support Condition Based Maintenance (CBM) of industrial assets. The article focuses on describing how the MANTIS Reference Architecture is implemented to support predictive maintenance of clutch-brake assets fleet, and includes the data analysis techniques and algorithms implemented at platform level to facilitate predictive maintenance activities. These technologies are (1) Root Cause Analysis powered by Attribute Oriented Induction Clustering and (2) Remaining Useful Life powered by Time Series Forecasting. The work has been conducted in a real use case within the EU project MANTIS.**

***Index Terms*— Keywords-Industry 4.0; Reference Architecture; Condition Based Maintenance; Cyber Physical Systems**

I. Introduction

The proliferation of CPS, Internet of Things (IoT) and cloud technologies are opening new opportunities for collaboration between systems, platforms and applications. This trend affects all aspects of life and numerous domains (Smart Cities, Energy, Health, etc.). Industry is not alien to these changes and opportunities. Manufacturers, production lines, solution integrators and engineering companies are investing to update their machinery and systems to the new situation.

Maintenance is essential for improved performance of industrial assets and processes. While reactive maintenance focuses on repairing an asset once failure occurs, proactive maintenance focuses on avoiding repairs and asset failure through preventive and predictive methods. CBM is a predictive maintenance strategy that is based on the continuous monitoring of various parameters of an asset to evaluate its health level and future development.

The availability of large quantities of data through IoT and CPS triggers the implementation of advanced monitoring strategies for asset management while facilitating the adoption of policies and strategies for maintenance measures, such as CBM [1]. However, this raises higher level issues that might have not been considered previously for condition monitoring scenarios: 1) How to transmit these data from the physical system and to where? 2) How to create interoperable data representation and semantics? 3) What can be the backend that processes this inbound data streams in a scalable manner? 4) How can we still maintain real-time restrictions and abide by communicational constraints? Even though data might be available with great time and value resolution, it is often not practical to be transmitted "as is" from the device or machine for communicational constraints.

To address these issues the ECSEL Project MANTIS [2] was born in 2014. This paper presents the implementation of the MANTIS reference architecture for one of the use cases in the project (GOIZPER use case). First, the background of the project and the use case are presented (section II). Section III introduces the MANTIS reference architecture. Section IV outlines the technologies and tools used to implement the reference architecture. Section V presents the algorithms for CBM implemented in the use case. Finally, results and conclusions are presented in Sections VI and VII.

II. Background

The main objective of MANTIS is to develop a Cyber Physical System based Proactive Maintenance Service Platform Architecture enabling Collaborative Maintenance Ecosystems. This reference architecture has to fulfil the requirements established by several industrial use cases demanding solutions for CBM that include sensors and SW at CPS level as well as a platform and tools for data analysis.

Intensive work has been conducted in the project to conceptualize, define and design the reference architecture and to identify the components, technologies and tools necessary to implement such solution. Those components, technologies and tools depend in great extent on the requirements established by the use cases in the project.

One of those use cases in MANTIS is concerned with analysing the clutch brake system and its components in press machines to detect the most important failure sources and be able to perform predictive maintenance in those press machines. The use case is led by GOIZPER. GOIZPER is one of the market leaders of power transmission components for metal forming machine tools like clutches, brakes or cams. Their final customers come from highly demanding sectors such as all world's top automotive manufacturers, stampers, home appliances or metallic furniture and are demanding products with high levels of quality and availability seeking a drastic reduction of high cost caused by production downtimes with required maintenance-repair operations and a better delivery times' compliance. That is why GOIZPER considers critical to increase machines and components reliability. To meet this challenge GOIZPER wishes to incorporate cutting-edge technologies in their products as a means

*This work was not supported by any organization

[1]Electronics and Computing Department, Mondragon University, Loramendi 4, 20500 Mondragon, Spain `flarrinaga, jfernandeza, ezugasti, igaritano, uzurutuza@mondragon.edu`

[2]Goizper S.Coop., Barrio Antigua, 4, 20577 Antzuola, Guipúzcoa `masasagasti,mmondragon@goizper.com`

of enhancing products robustness and functionality in order to facilitate proactive-predictive maintenance activities.

In MANTIS, the use case considers a test bench containing a Clutch Brake component. The use case has mainly incorporated dedicated smart sensors, pre-processing and data acquisition, communications in harsh environments, a platform for data gathering, treatment and analysis as well as tools and algorithms to support decision-making and to identify problematic situations.

Fig. 1. GOIZPER use case test bench.

The overall objective seek by GOIZPER is to early detect internal wear of a GOIZPER clutch-brake. To do that, the moving parts of the clutch-brake need to be sensorized. By continuously monitoring the system conditions proper operation of the clutch-brake can be ensured. Moreover, the most critical operating variables can be registered in the platform in order to analyse the working process and prevent misuses.

III. MANTIS Reference Architecture Model

MANTIS Architecture Reference Model (MANTIS-ARM) has been created to provide the cornerstone for designing, developing and deploying Predictive and Preventive Maintenance MANTIS-enabled architectures and solutions. The MANTIS-ARM consists of five main elements:

- Reference Model: a reference model is an abstract framework for understanding significant relationships among the entities of some environment [3].
- Reference Architecture: provide a template solution for the architecture (aka. architectural blueprint) for a particular domain [4].
- Feature model: Introduces key concepts to characterize common and varying aspects in the architectures to be derived from the reference architecture.
- Guidelines: discusses how the provided models, views and perspectives are used.
- Reference applications: show the diversity of the included solution variants, and thus illustrate architecture signification features and related design decisions.

The mission of the platform architecting activities in MANTIS includes to devise the overall architecture of the

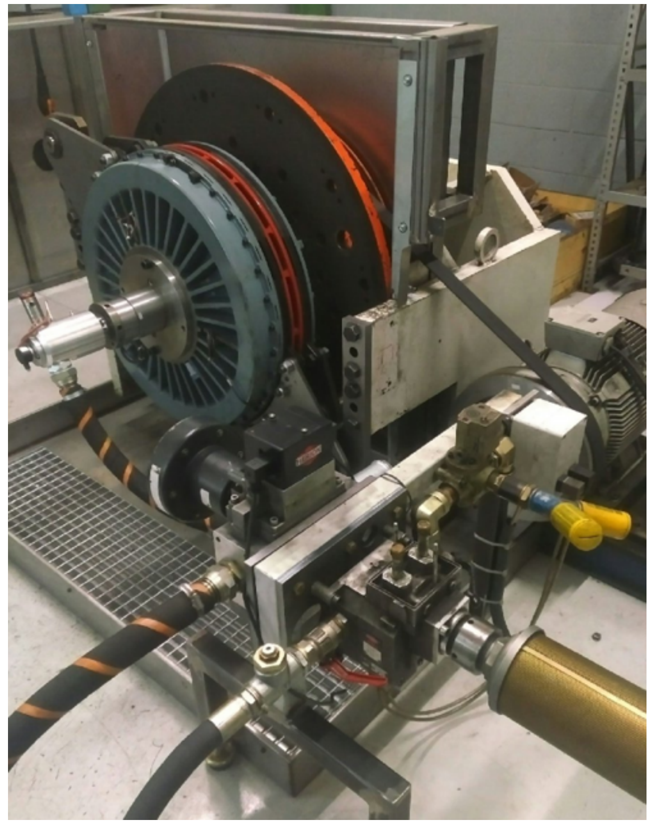

Fig. 2. GOIZPER use case clutch-brake in operating mode

MANTIS distributed system for proactive maintenance, to address issues that have an impact on several steps in the chain of turning raw data into information usable for distributed decision-making and to consider key aspects: interoperability, consistency, availability, reliability, robustness, safety and security of the system as a whole.

A reference service platform architecture shall allow for industries participating in MANTIS to take advantage of progress on proactive maintenance in related but different industries. In addition, it allows for less mature industries in the project to reuse experiences from industries in the forefront of proactive maintenance. They can thereby ensure that improvements in maintenance can be achieved gradually and consistently with future plans and best practice. Important aspects that the architecture should address towards the use cases are:

- Interface, protocol and functional interoperability
- Data validation ensuring that data analyses are made on data that give clean, correct and useful data information about the system.
- Distributed data, and information processing and decision-making ensuring consistent behaviour and avoid contradicting actions, e.g. between local and distributed data analysis and decision-making.
- Information validation ensuring that created information still is relevant for the system analysed in particular for CBM.
- Safety and fault tolerance ensuring that critical infor-

mation remains available and following decisions can be taken or proposed although partial system failure.

- System and service level security ensuring that the system incorporates means to hinder misconfiguration and can be protected from wire-tapping and various attacks.
- System engineering and reusability of defined and existing services.
- System verification and validation of the service platform architecture and overall design, covering both functional and non-functional properties.

A. Architecture and Interoperability levels

The Mantis architecture described in [5], considers a number of components separated in three tiers. The identification of those levels for interoperability is inspired by the IIC-RA three-tier architecture pattern [6] that comprises edge, platform and enterprise tiers (see Figure 2). These tiers play specific roles in processing the data flows and control flows involved in usage activities.

The edge level comprises physical entities that belong to the same local network and or functional area. It implies the virtualization of the physical entities into CPS (called component level interoperability) and the provision of the data extracted from the CPS to the platform level. The platform level receives processes and forwards commands from/to the edge level. It provides more complex and resource consuming data analytics and knowledge generation functionalities. It is also concern on how to represent the knowledge models generated by data analytics digital artefacts. The enterprise level is concerned with the applications that integrate information from one/several sites to enhance the global decision-making process using monitoring through Human Machine Interfaces (HMI) and data aggregation and analysis.

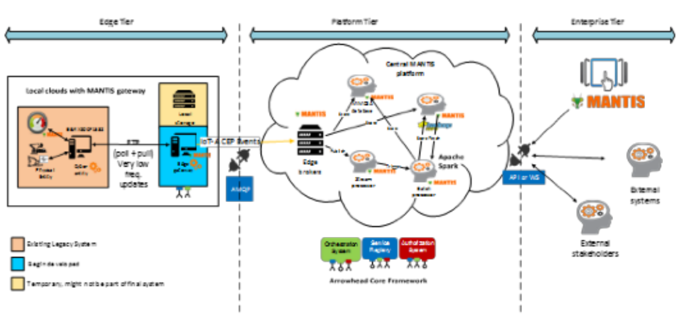

Fig. 3. Mantis Architecture levels.

B. Methodology

The methodology followed to build the GOIZPER architecture, components and its data models is also specified in Mantis. The approach for designing MANTIS use cases follows the principle of architecting for concrete stakeholder concerns (Architecture Drivers). These stakeholder concerns will drive the actual architecture design, which is based in the approach follow by the SPES consortium [7]. This approach suggest to start by delineating system and its context. Continue with the functional decomposition of the system. The next step is the software realization of software. The final steps considers the hardware realization of functions and the deployment of software entities.

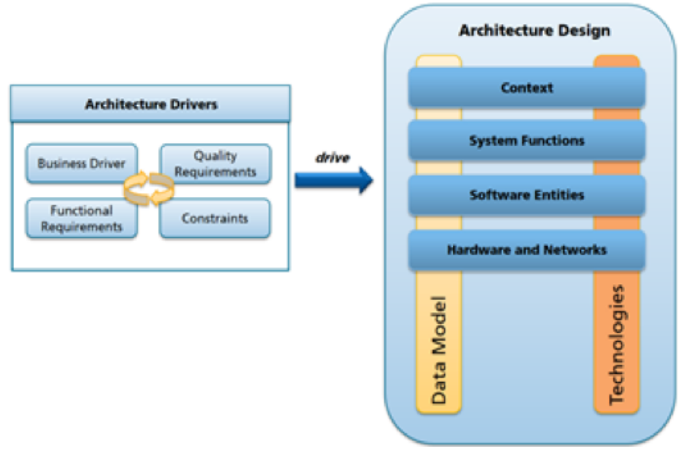

Fig. 4. Mantis architecture construction approach.

IV. Platform Technological Implementation

The implementation of the Mantis reference architecture in the GOIZPER use case constituted the technological platform shown in Figure 3, which implements the following blocks:

1) Data Access and Ingestion through the Edge Broker: Provides the access to the platform and the adaptation of the messages coming from the CPS to the Information Models and data structures provided at platform level. The Edge Broker is composed of:
   a) Publish-Subscribe servers: manage messages from/to CPS and internal components of the platform using queues and exchanges. The solution has been implemented using RabbitMQ [8] and Advanced Message Queuing Protocol (AMQP) [9].
   b) Translator/Converters: convert or translate input data formats into output data formats and perform protocol mapping. CPS messages are generated according to the MANTIS Event Information Model (based on the IoT-A event information model) and converted into the storage formats presented in the next paragraph. The converters have been implemented using an Enterprise Service Bus (ESB) named WSO2[10].
2) Data Storage systems: store information coming from CPS and results of data analysis maintenance actions. Two storage systems:
   a) MIMOSA DB[11]: is a database compliant with the ISO-13374 Standard (Condition Monitoring and Diagnostic of Machines). According to this standard, a CBM system should be composed of various functional blocks: Data Acquisition (DA), Data Manipulation (DM), State Detection

(SD), Health Assessment (HA), Prognostics Assessment (PA) and Advisory Generator (AG)[12]. One of the main objectives of the MIMOSA CBM architecture is to standardize the information flow between the various blocks, so that equipment from different vendors could be interoperable. The MIMOSA database is deployed in SQL Server and API REST is used to access data from applications.

b) Hadoop Distributed File System (HDFS): is a distributed file system designed to run on commodity hardware. Designed to be deployed on low-cost hardware, HDFS is highly fault-tolerant and provides high throughput access to application data, which makes suitable for applications that have large data sets.

3) Batch Processor: data analysis and processor mechanisms to enable the management of large volume of data, fetched from storage systems and process on demand. Implemented using Apache Spark [13]. See next section for details on data analysis techniques employed in the use case.
4) HMI: Describe purpose and technological solution.

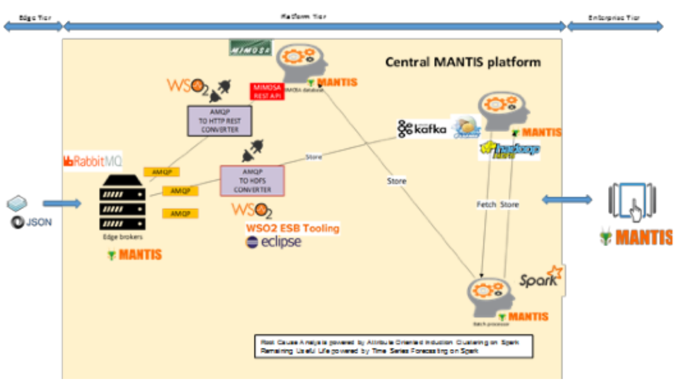

Fig. 5. Goizper platform architecture.

V. Data Analysis for CBM

The following aspects of CBM have been addressed for the Goizper use case: Equipment Failure Root Cause Analysis: The Root Cause Analysis (RCA) is the first and necessary step to identify the main equipment failure causes. Within Goizper use case, Attribute Oriented Induction (AOI) algorithm has been used as the principal RCA algorithm.

Attribute Oriented Induction algorithm is considered a hierarchical clustering algorithm. First proposed by [14] Jiawei Han et al. as a method for knowledge discovery in databases, it is currently considered a rule-based concept hierarchy algorithm. The representation of the knowledge is structured in different generalization-levels of the concept hierarchy with IF-THEN rules. The execution of the algorithm AOI follows an iterative process in which each variable (also referred as attribute) is generalized based on its own hierarchy-tree. This step is denoted as concept-tree ascension [15]. To ensure the correct functioning of the algorithm, it is necessary to establish background knowledge, which specifies attribute generalization levels.

Table 1 shows a visual representation of the generalization process. The first step is to select the variable with the higher number of distinct values, 39 in the example, to then generalize following the criteria established by the background knowledge (e.g. [0, 3]: X, (3, 100]: Y).

By identifying data similarity clusters, AOI provides a knowledge representation of different machine behaviour states. The generation of machine behaviour states knowledge base during the learning process, ensures the representation of all possible machine-working states. Hence, the anomaly working states detection step is simplified, since it only requires the identification of a previously unknown machine behaviour state. This step is called Detection.

TABLE I
Results before Generalization step

	Var 1	Var 2	...	Var N
Tuple 1	0	0.50	A	21
Tuple 2	1	0.90	A	24
Tuple 3	1	1.80	A	30
Tuple 4	1	2.40	B	38
Tuple 5	1	4.20	C	42
Tuple 6	1	4.80	C	43
...	...	...	...	...
Tuple 39	1	3.21	F	39
# Dist. Val.	2	**39**	6	30

TABLE II
Results after Generalization step

	Var 1	Var 2	...	Var N
Tuple 1	0	X	A	21
Tuple 2	1	X	A	24
Tuple 3	1	X	A	30
Tuple 4	1	X	B	38
Tuple 5	1	Y	C	42
Tuple 6	1	Y	C	43
...	...	...	...	...
Tuple 39	1	Y	F	39
# Dist. Val.	2	2	**6**	30

The process of estimation of RCA is a specification of the Detection results. The RCA estimation process requires the previous definition of variable and failure-types relations (i.e. If an anomaly occurs with high temperatures but low-pressure values, it is possible to have a problem on the component X). Thus, domain knowledge is mandatory if accurate results are expected.

Equipment Remaining Useful Life estimation: The main objective of the Remaining Useful Life (RUL) estimation process is to estimate the useful life of an asset before a catastrophic failure occurs. Within Goizper use case, the RUL estimation process is performed as a combination of AOI algorithm outcome and Auto Regressive Integrated Moving Average (ARIMA) statistical time series forecasting models. A common objective of Time Series Forecasting methods is to learn from previous data in order to be able to make predictions of future behaviours. The knowledge base generated by AOI algorithm let us to check whether a machine state is already registered in the knowledge base. Furthermore, the order of appearance of machine states can be also evaluated, which provides knowledge about machine behaviours.

In order to estimate the RUL, the first step is to evaluate a new variable to represent the machine behaviour correction factor, denoted as Normality Factor. The Normality Factor quantifies the extent of the damage of the machine. Each inspected work-cycle has its own Normality Factor value to represent its own normality level. The average of the Normality Factor values of the work-cycles utilized to generate the knowledge will determine the Normality Factor value of the machine. By applying ARIMA time series forecasting models, the Normality Factor evolution is modelled. As a final result, the Normality Factor model allows to predict the wear of the Normality Factor, providing the machine RUL in terms of clutch-brake cycles. Finally, clutch-brake cycles are translated into days, by combining the number of cycles the clutch-brake system does per day.

VI. Results

The main results obtained after implementing the reference architecture and the data analysis performed over the data collected are:

- A platform that accommodates different industrial processes and assets data for CBM analysis.
- Integrate an interoperable data model for CBM.
- A data/protocol converter that enables translations between most common data formats and protocols.

Regarding data analysis, a small experiment has been performed as a proof of concept over the 'break' data in order to show and demonstrate the ability of the proposal. For this experiment, many features of the clutch-break machine have been used in the cluster generation step, such as: trigger, angular position, application pressure, line pressure and flywheel speed. After the knowledge base has been generated by the application of the algorithm AOI and the calculation of the most significant cluster-appearance order for the working cycles, the anomaly detection step has been processed. To detect an anomaly on the behaviour, the average value of the Normality Factor had been calculated. That Normality Factor value is the threshold to determine if a current work-cycle can be considered correct. In this experiment, the value was 0.70.

The Normality Factor evolution signal shown in the Figure 6 should be the result of applying ARIMA model over the training data utilized to generate the knowledge base. In this experiment, next two hundred and fifty 'break' working cycles have been predicted. As it can be observed, there are five different work cycles cutting the established Normality Threshold; thus, it can be inferred that five different anomalies are detected. Next step should be to analyse the characteristics of the anomalies, inspecting the reasons of their occurrence (e.g. if there is any cluster in the abnormal work cycle which is not registered in the knowledge-base, check the features or the grips of features in which the new values have occurred in order to establish the reason of the failure; if the order of the clusters inside the abnormal working cycle is significantly different respect to the ones registered in the knowledge-base, check the evolution of the values of the features in order to specify the reasons of the failure). As this is a preliminary work and has been shown as a proof of concept, there are several future work to perform in order to validate many other aspects of the system, especially concerning to the estimation of RCA.

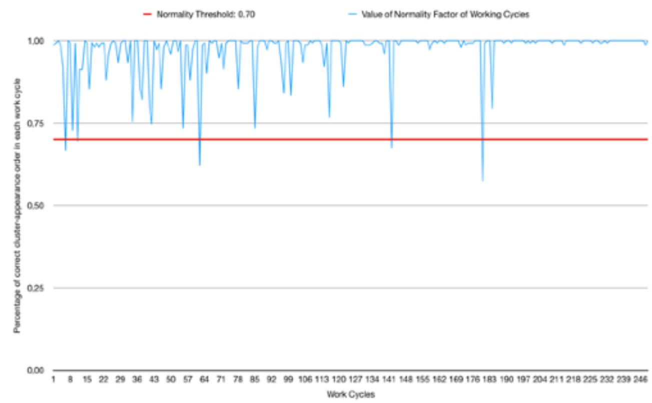

Fig. 6. Graphical representation of Anomaly detection based on the evolution of Normality Factor.

VII. Conclusions

This paper describes a specific implementation of the MANTIS reference architecture. The technologies and tools employed to articulate the components for an Industry 4.0 platform are presented. The methodology proposed in the project has been also followed. This methodology focuses on the stakeholder requirements and provides well-structured steps necessary to build the solution. Further, the implementation of the platform enables the provision of maintenance services based on the data collected from sensors and CPS. As other implementations of the same reference architecture [16] [17], the objective is to focus on proactive maintenance services to improve asset availability at lower costs through continuous process and equipment monitoring and data analysis. Attribute Oriented Induction Clustering and Time Series Forecasting are being used to determine Root Cause Analysis powered and Remaining Useful Life. Although only preliminary results have been obtained, the solution provides the mechanisms to analyse data and estimate valuable maintenance parameters.

Acknowledgment

This work has been developed by the intelligent systems for industrial systems group supported by the Department of Education, Language policy and Culture of the Basque Government. It has been partially funded by the MANTIS project. The project has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 662189. This Joint Undertaking receives support from the European Union's Horizon 2020 research and Ministerio de Industria, Energia y Turismo (Spain). It has also been partially funded by the Basque Government.

REFERENCES

[1] Jantunen, E., Zurutuza, U., Albano, M., di Orio, G., Maló, P., and Hegedus, C.: The Way Cyber Physical Systems Will Revolutionise Maintenance, presented at the 30th Conference on Condition Monitoring and Diagnostic Engineering Management, (2017).

[2] The MANTIS consortium , "The mantis project website," http://www.mantis-project.eu/

[3] https://www.oasis-open.org/committees/soa-rm/faq.php

[4] ISO/IEC, ISO/IEC/IEEE 42010:2011 "Systems and software engineering - Architecture Description", ISO/IEC, 2011.

[5] C. Hegedus, P. Varga and I. Moldovan "The MANTIS architecture for proactive maintenance" 5th International Conference on Control, Decision and Information Technologies CODIT 2018 Thessaloniki (Greece) (accepted).

[6] Industrial Internet Consortium, "The Industrial Internet Reference Architecture," Technical Report, 2015.

[7] K. Pohl, H. Hnninger, R. Achatz and M. Broy, Model-Based Engineering of Embedded Systems: The SPES 2020 Methodology, Springer Publishing Company, 2012.

[8] https://www.rabbitmq.com/

[9] Vinoski, S. (2006). "Advanced Message Queuing Protocol". IEEE Internet Computing 10: 87-89.

[10] https://wso2.com/

[11] MIMOSA. Retrieved from: http://www.mimosa.org/mimosa/

[12] ISO 13374-1: Condition monitoring and diagnostics of machines - Part 1: General Guidelines.

[13] https://spark.apache.org/

[14] Han, J., Cai, Y., and Cercone, N. (1992). Knowledge Discovery in Databases: An Attribute-Oriented Approach. In Proc. 18th Int. Conf. Very Large Data Bases, Vancouver, Canada (pp. 547-559).

[15] Cheung D.W., Wai-Chee Fu A., Han J. (1994) Knowledge discovery in databases: A rule-based attribute-oriented approach. In: Raś Z.W., Zemankova M. (eds) Methodologies for Intelligent Systems. ISMIS 1994. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol 869. Springer, Berlin, Heidelberg.

[16] L. L. Ferreira, M. Albano, J. Silva, D. Martinho, G. Marreiros, G. di Orio, P. Maló, and H. Ferreira, "A pilot for proactive maintenance in industry 4.0," in Factory Communication Systems (WFCS), 2017 IEEE 13th International Workshop on. IEEE, 2017, pp. 1-9.

[17] Csaba Hegedűs, Paolo Ciancarini, Attila Frankó, Aleš Kancilija, Gregor Papa, Špela Poklukar, Mario Riccardi, Alberto Sillitti and Pal Varga: "Proactive Maintenance of Railway Switches", 5th International Conference on Control, Decision and Information Technologies CODIT 2018 Thessaloniki (Greece) (accepted).