CISTER
Research Centre in
Real-Time & Embedded
Computing Systems

Conference Paper

A Pilot for Proactive Maintenance in Industry 4.0

Luis Lino Ferreira*
Michele Albano*
José Silva*
Diogo Martinho
Goreti Marreiros
Giovanni di Orio
Pedro Maló
Hugo Ferreira

*CISTER Research Centre

CISTER-TR-170407

2017/05/31

A Pilot for Proactive Maintenance in Industry 4.0

Luis Lino Ferreira*, Michele Albano*, José Silva*, Diogo Martinho, Goreti Marreiros, Giovanni di Orio, Pedro Maló, Hugo Ferreira

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: llf@isep.ipp.pt, mialb@isep.ipp.pt, 1111782@isep.ipp.pt

http://www.cister.isep.ipp.pt

Abstract

© CISTER Research Center
www.cister.isep.ipp.pt

A Pilot for Proactive Maintenance in Industry 4.0

Luis Lino Ferreira[1], Michele Albano[1], José Silva[1], Diogo Martinho[1], Goreti Marreiros[1], Giovanni di Orio[2], Pedro Maló[2], Hugo Ferreira[3]

[1]ISEP, School of Engineering, Polytechnic Institute of Porto, Portugal
[2]CTS, UNINOVA, Dep. De Eng. Electrotécnica, FCT-UNL, Portugal
[3] INESC TEC, INESC Technology and Science, Portugal

***Abstract* — The reliability and safety of industrial machines depends on their timely maintenance. The integration of Cyber Physical Systems within the maintenance process enables both continuous machine monitoring and the application of advanced techniques for predictive and proactive machine maintenance. The building blocks for this revolution – embedded sensors, efficient preprocessing capabilities, ubiquitous connection to the internet, cloud-based analysis of the data, prediction algorithms, and advanced visualization methods – are already in place, but several hurdles have to be overcome to enable their application in real scenarios, namely: the integration with existing machines and existing maintenance processes. Current research and development efforts are building pilots and prototypes to demonstrate the feasibility and the merits of advanced maintenance techniques, and this paper describes a system for the industrial maintenance of sheet metal working machinery and its evolution towards a full proactive maintenance system.**

Keywords—industry, analytics, Cyber Physical Systems

I. INTRODUCTION

Cyber Physical Systems (CPS) have been game-changers in several areas, from deeply embedded systems to smart cities. Industrial systems can benefit from the advances developed in the field of CPS, particularly for the predictive and proactive maintenance of industrial machines. In fact, access to large volumes of data can be instrumental in forecasting machines malfunctions, and scheduling maintenance operations [1, 2]. On the other hand, the collection, transport and management of large volumes of data poses a number of challenges, which span from technical to organizational, the latter usually being related with the integration of existing industrial processes.

The Cyber Physical System based Proactive Collaborative Maintenance (MANTIS) project [3] is an international initiative that aims at building a platform for proactive maintenance of industrial machines. MANTIS proposes, as one of the options, for the structuring of the communication system upon a communication middleware and to provide a set of guidelines for the integration of the system in different scenarios in order to cope with organizational issues. The middleware allows integration in a compositional fashion in the sense that, new modules can be added without requiring any changes to the existing systems. These modules, supported by the middleware, offer data collection, processing and analysis, and data visualization through advanced Human Machine Interfaces (HMI) thereby achieving the goals of the machine monitoring system. Here we report on the development of a prototype that is being used to test and demonstrate the feasibility of the proposed solution.

978-1-5090-5788-7/17/$31.00 ©2017 European Union

The development of the system followed an incremental, goal-driven, bottom up process, by building modules for an initial scenario, and providing a middleware that acts both as a data transport and adaptation layer between the machines and the processing modules. Once the interfaces with the data processing modules have been defined, the middleware‘s role is to conserve backwards compatibility. However, it can still evolve by both ensuring compatibility in new scenarios and increasing the middleware’s capabilities, e.g.: performance, access to external data, flexibility or support for different protocols.

Therefore, this paper describe a pilot being developed within the aims of the European MANTIS project for Press Break Steel Bending machines. Most of what is described is already implemented, but some parts are still being improved. The paper is organized as follows. After presenting some background information on the topic at hand in Section II, this paper describes in Section III the reference architecture of the maintenance system, describing its building blocks. Section IV, describes the pilot being implemented. Finally, Section V wraps up the paper with a discussion of the results.

II. BACKGROUND INFORMATION

A. *The MANTIS project*

The MANTIS project as a whole is a large undertaking with several use-cases pertaining to industrial maintenance in various industries comprising energy production, manufacturing and transportation. This article deals specifically with the system designed to aid in the context of industrial maintenance of sheet metal working machinery using a press break machine.

Concretely the machine under study in this paper is a press brake. Press Braking is the process of deforming a metal sheet

(workpiece) along a given axis by pressing it between clamps (tools). In order to have a finished part, a metal sheet will be consecutively bent at several places – e.g. to make a computer box.

The press brake model used is a hybrid system that is powered both hydraulically and electrically, and is controlled via a fluid pumping sub-system. The hydraulics drive two pistons located on a pair of beams that serve as actuators. These actuators move a ram vertically up and down onto a die that is fixed on the machine's base. The ram holds a punch. The workpiece is placed between the punch and the die, acting as clamps, so that it can be deformed. The type of deformation depends on various factors such as the metal's characteristics (type, thickness) and the geometry of the dies and punch (which are manually changed according to the task). Figure 1, shows the front of the machine.

Figure 1 – Frontal view of the machine

Although the process seems simple, it requires a set of sophisticated control mechanisms to ensure correct bending (automatic back-gauge placing) and compensate for effects such as metal spring-back [19] and uneven loads [20]. In order to obtain precise results, the system uses sensors that detect the machine's frame deformation, evaluate the bend angle and measure the applied pressure. To achieve this, a sophisticated computer aided manufacturing simulation software is used to predict the bending, and a numerical control software to execute the bending process. The machine also includes sensors such as lasers and cameras, used to ensure safety and aid operators in their tasks execution.

The objective of the MANTIS project is to reduce maintenance costs by means of novel monitoring techniques. This requires detecting faults as early as possible to avoid catastrophic failure, predicting failures in order to facilitate the scheduling of the parts replacement and providing tools that ease the diagnosis of the problems. This approach reduces machine down-time, eliminates excess spare-parts stock, improves product quality, increases operator safety and lowers the overall cost of maintenance [8].

B. Maintenance in Industrial Scenarios

Maintenance is a set of tasks whose objective is to maintain the equipment and all other related physical assets in the desired operational state within a given economic and business context. This includes all the engineering decisions required to optimize operations in relation with: productive capacity, production quality, failure rate and response capabilities [4]. Optimization must be attained with a minimum amount of resources while ensuring the safety of the machine operators [5].

Maintenance has evolved from simple techniques such as visual inspection, using physics based models that describe the components' mechanical behavior, to more sophisticated methods based on signal processing, pattern recognition, and empirical data based prediction and classification models [1].

Maintenance can be characterized according to its function:

• Detection: is the identification of failure when it occurs (or is about to occur). The aim is to quickly terminate operation in order to safeguard operators and avoid further damage;

• Prognosis: is the prediction of failure within a given period of time. Usually a probability of failure is included with this prediction.;

• Diagnostics: identifies the failing component and/or the root cause of the failure. Empirical models, which may also include domains specific knowledge, can guide the user in diagnosing the problem. Another approach that can provide benefits, can be based on artificial intelligent tools such as knowledge-based expert systems to diagnose machine failures.

In fact, the MANTIS platform targets 3 main scenarios that are aligned with these functions. The first one considers the *Detection of Component Failure*, and it aims to demonstrate the ability to detect machine failures in a timely fashion by means of the detection models. The second scenario is focused on the *Prediction of component failure*, and it aims to demonstrate the ability to perform proactive maintenance of the machine, by predicting machine failures before they occur; in this case, the prediction models use sensed data to estimate the "remaining useful life of wearing components". Finally, the third scenario is centered on the *Diagnosis of component failure*, and it aims to demonstrate the ability of an analytics model to help identify the root cause of a failure, by means of "root cause failure analysis" models that guide the user in his efforts to diagnose the root cause of a problem.

C. Industrial Automation Architectures and Protocols

It is common practice to organize operations related to industrial automation into five levels [10]. Level 0 (Field Level), comprises sensor and actuators that interact directly with the process or the machine; Level 1 (Direct Control) controls special-purpose hardware, such as Programmable Logic Controllers (PLCs), industrial PCs and DSP processors, by means of real-time custom software; Level 2 (Supervisory Control) executed on general purpose processors to perform online control of the industrial process; Level 3 (Production Control) that considers the manufacturing operation as a whole, and includes maintenance, production, quality assurance and inventory management; Level 4 (Enterprise Control), which

consists mainly of management functions, and is used to drive the manufacturing process by scheduling its operations.

Communication among the devices in this levels is structured upon standards, like, CANopen, OPC-UA, PROFINET, MIMOSA, just to name a few. An alternative solution, which we advocate in this paper, is to structure communication among devices in each level or between levels upon a communication middleware. Middleware technologies are directed at hiding the complexity of underlying technologies and easing the usage and management of the CPS resources, abound today. Current solutions already provide many advanced capabilities that are built on solid architectural models.

The Self-organized and Intelligent Middleware (SIMPLE) [11] platform exhibits self-organizing properties, focuses on data dissemination using multi-level subscriptions processing, and uses a tiered networking approach to cope simultaneously with large, widespread and heterogeneous sensing networks. The SIMPLE middleware is able to provide a robust zero-configuration solution, with no central point of failure, and delivers the required data efficiently to the correct applications.

Another example is OpenIoT [12], an open source platform that includes unique functionalities such as the capability to compose (dynamically and on-demand) Internet of Things (IoT) services, adhering to the cloud/utility-based paradigm. OpenIoT developed a blueprint middleware infrastructure for implementing and integrating IoT/CPS solutions.

The Service Oriented Architecture (SOA) paradigm advocates the design of software structures upon a set of distributed services. Some of the characteristics of service orientation, such as portability, loose coupling, reusability, composability, service discoverability and security, are also potentially beneficial for IIoT applications. The Arrowhead Framework [13, 24] implements a SOA for automation applications. The approach considers that every interaction, from service registration, orchestration to authentication, is mediated via services. The Framework provides interoperability and unified interaction mechanisms for different industrial and non-industrial scenarios.

The OPC-UA protocol [14] is a SOA multi-platform protocol for industrial automation. It focuses on machine-to-machine communication, and provides an information model based on nodes and attributes that is inspired on object oriented programming. OPC-UA provides connection heartbeat, automatic buffering of messages, scalability, discovery of devices and robust security, with implementations in several mainstream platforms. The protocol can be used to transfer data pertaining to many different industries, from smart grids, to industrial machine monitoring and maintenance.

Communication through a Message Oriented Middleware can also provide characteristics such as loose coupling and better performance [15]. Among the ones that exist in the market, the RabbitMQ solution [16] (which was chosen to be used in the pilot described in this paper) is an open source solution that supports multiple communication protocols, is proven to be very reliable and well tested, allows for flexible routing, and supports several topologies. Internally, message exchange can be made by remote procedure calls or via regular message queueing mechanisms. Moreover, RabbitMQ topologies scale well by means of its *clustering* [16] capability. Finally, this middleware runs on all major operating systems and is supported by a large developer community.

The use of standardized data models and ontologies is also important to guarantee the interoperability between different components. The Machinery Information Management Open Systems Alliance (MIMOSA) [17] is an international initiative that aims a wide adoption of information standards in manufacturing industries. The MIMOSA initiative spans from data modeling to information encoding, and it is promoted by many prominent companies in both manufacturing and IT. The MIMOSA approach embraces many hierarchical levels, supporting a widely used approach for data modeling, plus best practices [18].

III. Reference Architecture

The distributed platform employed in the MANTIS project, whose architecture is depicted in Figure 2, is composed of a number of modules that can be grouped into 5 logical blocks: Machine, Edge Local, Data Analysis and Human Machine Interface (HMI) connected together by a Communication Middleware, which can be split in Local Middleware and Cloud Middleware. This section includes information on the logical blocks and describes how they interact to provide an infrastructure that can be deployed and integrated into existing systems.

A. Machine

Data on the machine are collected by means of sensors that are part the machine's control systems or from sensors which were added specifically for maintenance purposes. This logical block can consist of several modules, each representing a machine's subsystems. Obviously, the number and kind of modules depends on the machine being monitored. Usually, these modules provide access to four different data sources: the machine CNC, which controls the machine, merging data from the PLC-connected sensors and actuators, and the Safety PLC. Data can also be obtained from Maintenance Sensors.

The Machine logical block is comprised of both pre-existing modules, and the ones added specifically for the maintenance platform itself. The PLC sensors are already part of the machine, and are used internally to control its operation. These range from buttons and pedals to advanced electric motor drives, with positioning information. Although, used primarily for control functions, these sensors can also be used to determine anomalous events or states, to diagnose problems and even to infer the root cause of problems. The CNC, normally, is also able to perform some diagnosis functions that allow the identification of some failures and the generation of warnings, locally.

The PLC works in close cooperation with the CNC controlling all automation functionalities and, at the same time, it is able to send information from its sensors to the CNC.

The Safety PLC handles only safety-related functions for the machine, such as preventing humans from being too close while the machine is working, detecting critical conditions, etc. Data from these sensors is mainly used to distinguish between

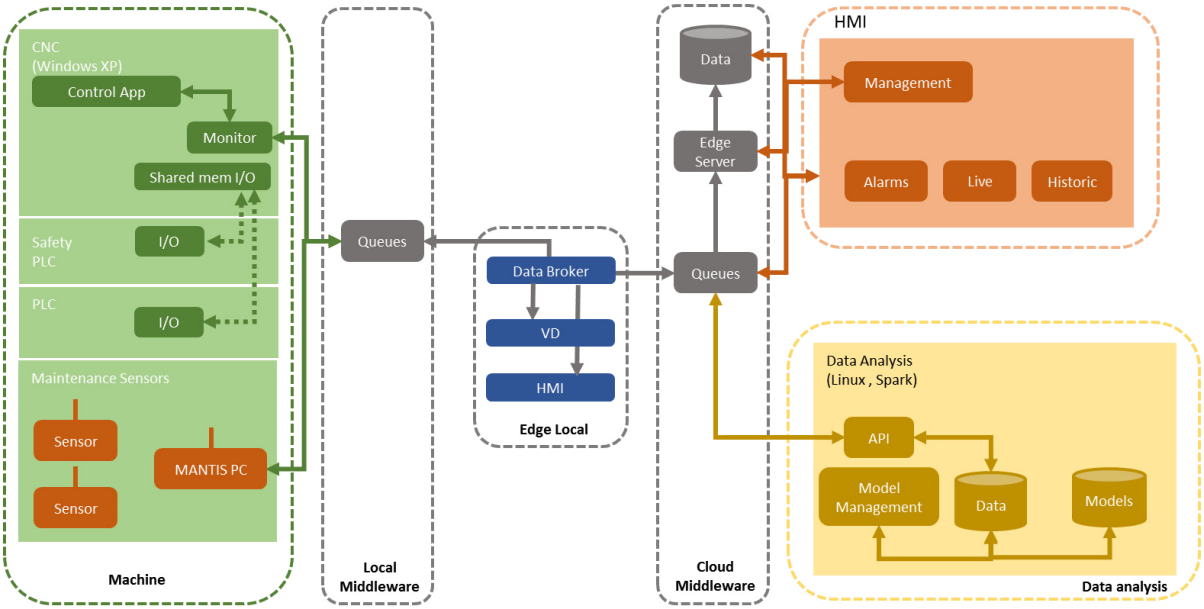

Figure 2 – Reference Architecture

component failures and safety-related events.

Finally, the Maintenance Sensors are sensors placed on the machine, usually communicating over an independent channel (e.g. a wireless network) that only acquire specific maintenance-related information, such as oil quality and data on the machine's moving parts. This data are then aggregated by the MANTIS PC, which establishes the interface between low level protocols and the Local Middleware.

B. Local Middleware

The Local Middleware is used inside a factory connecting its machines with management systems like to a Manufacturing Execution System (MES), SCADA systems, Enterprise Resource Planning (ERP) and also to the MANTIS maintenance system described in this paper.

The solution adopted is based on a communication middleware. The advantage of using such a solution is that the information can be easily shared among different applications, thus reducing the network traffic with the machines, easing the programmability of the system, its management, and finally, reducing the need to run additional software on the machines CNC controllers. Thus, the Local Middleware connects to each Monitor module, present in each machine, and delivers that information to the Edge Local block. A similar solution is also being adopted by OPC-UA [14] and this solution allows to decouple producers and consumers of information.

C. Edge Local

The main objective of this logical block is to isolate the factory from the outside world, at the same time providing some functionalities at local level. From the security point of view, the Edge Local can be seen as creating a DeMilitarized Zone (DMZ) in the sense that it is the only module in the factory premises that has network access, and thus concentrates all the security requirements on itself.

Data collected from multiple Machines, usually inside a factory, has to be made available to the other logical blocks of the platform. The Edge Local logical block provides mechanisms to support communication and management of the data acquired across multiple heterogeneous and distributed data sources. This is accomplished by providing an abstraction layer that detaches the application development from the intricacies of the lower level details. It acts as a virtualization platform and as data broker that connects the Machine logical block to the Cloud Middleware, capable of extracting, collecting, distributing/sharing, pre-processing, compressing, and semantically enhancing the data produced in an efficient manner. Therefore, the one of the fundamental goals of the Edge Local logical block is – from one side is to support the data integration of multiple data sources and – from the other side is the provisioning of data to the cloud where more complex and resource consuming data processing takes place.

The Edge Local logical block is composed by three modules - the Virtual Device, the local HMI service and the Data Broker. The Virtual Device is responsible for virtualizing physical entities (machines and industrial assets) available in the shop floor. These Machines and assets are virtualized in terms of their capabilities to facilitate and enhance the process of exchanging data (machine data readouts). The local HMI service is responsible for visualizing all the necessary information generated within the Edge Local logical block, i.e. Virtual Devices available, machine data readouts. Finally, the Data Broker operates as a gateway allowing the indirect connection between the machines in a factory and the Cloud Middleware, which performs data analysis and supports advanced HMI features.

D. Cloud Middleware

This logical block manages the data, by storing and transporting it between the Edge Local (eventually from multiple factories) and, both, the Data Analysis and HMI

modules. It operates through three modules, the Edge Server, the middleware Queues and the Database.

The Edge Server manages a communication middleware, which receives data from several Edge local devices through the queues and saves the data to a Database (DB) module, which is structured according to the MIMOSA standard [17]. Additionally, the queues also support the communication with the Data Analysis and HMI blocks, and between those two blocks.

The Edge server also makes available a set of services which are used by the HMI to configure the systems. This configuration information is then permanently stored on the Database and used to support system startup and resume system operation in case of a crash. As explained in more detail later in Section IV.E this solution provides an adequate level of security and enables the isolation of data between all factories, at the same time providing a highly scalable solution.

E. Data Analysis

The Data Analysis logical block includes three modules. The first is a set of Prediction Models module, used for the detection, prognosis and diagnosis of machine failures. The models can be built for one machine family, or can be generic and adapted to different machine families. The second is a Prediction Application Programming Interface (API) that outputs predictions from the models, and provides data to feed and train the models. The third module is an Intelligent Maintenance Decision Support System (IMDSS), which is used to manage the models (model generation, selection, training and testing), for example on reception of training data, or when the API is contacted. The IMDSS is composed of a Knowledge Base that uses diagnosis and prediction models and the data sent by sensors. On top of this Knowledge Base there will be a Rule based Reasoning Engine which includes all the rules that are necessary to deduce new knowledge that helps the maintenance crew to diagnose failures.

In addition to the data and algorithms, expert knowledge has been encoded as a set of rules that are used to detect and flag possible failures. Each rule indicates what sensor and CNC signals need to be acquired, how they are segmented, the type of analysis to be executed and what failure is associated with these signals.

As an example, let us consider when the brake press is working in automatic mode, terminates its bend cycle and has parked the ram on the top position waiting for the next task. If no failure exists then the ram must remain still in the same position where it stopped. Because the hydraulic system is constantly losing pressure, the CNC compensates for any deviation. Normally, such deviations are minor (imperceptible to the naked eye) and occur at very low rates. However, if a hydraulic pump fails or a hydraulics tube ruptures, leaks will cause large deviations as the CNC compensates for this.

In order to detect such problem, the positions of the pistons are recorded when the control signal indicates that the ram is at top dead center (segmentation). Statistical tests are used to check that the deviation is within a specific tolerance threshold. This threshold is determined via the machine learning algorithm (stream based) and is tweaked in order to reduce the false positive and negative detection rates.

F. Human Machine Interface

This module provides a Human interface for the proactive maintenance system. It has two main components, one for data visualization and another for data management.

In the visualization component it is possible to view historical and live data, which is collected from specific machine sensors (e.g. machine status, speed, positioning and pedals state). It is also possible to show the results generated by the data analysis module, more specifically the alarms for unusual sensor data and the warnings regarding impending failures. It is possible to match the warnings from the Data Analysis block with historical data collected from the sensors.

The Management component includes all the administrative operations, like users and roles management, as well as factories and machines setup. Role management is a very important process that allow one to dynamically assign specific permissions to each type of user (e.g. a user with the “operator” role can view historical and live data only). Factories and machines management, allows an authorized user to setup a new factory and its machines.

The HMI follows a web-oriented design and therefore can be accessed from anywhere, at any time and through all sort of electronic devices with the only requirement being the use of the Internet to do so. This allows both remote (administrative) and on-site operations such as analyzing the machine’s state or view its past performance.

IV. Pilot Implementation

This section describes one of the pilots on industrial machine maintenance that has been implemented for the MANTIS European project, showing how the reference architecture in Figure 2 has been instantiated.

A. Machine

Data is collected indirectly from the CNC of an ADIRA Green Bender Press Break machine (Fig. 1), which in turn collects information from the PLC control system of the machine and from its Safety PLC. An application on the CNC stores data regarding raised alarms, machine configuration and ERP-related information (e.g.: production related data such as type of metal and bend) on a Microsoft Access database. Note that the CNC is based on a Windows machine. The same application stores data collected from existing machine sensors (e.g.: extensometers, pressure sensors, oil temperature and oil quality), which is collected from shared memory and to a file in the CNC filesystem. The information stored on that file is then sent to the Edge Local node though the Local Middleware. Obviously, the ideal solution would be to access the shared memory directly by a single software module, but this solution was a compromise in order to ensure the safety and certification of the machine control system. Nevertheless, this situation is currently being implemented for a future version of this software.

The application can be tailored and configured to different machines and applications, but the current pilot collects data from 50 machine sensors, with a periodicity of 20 ms, and from the MS Access database, which is scanned every second. The amount of data generated and transmitted to the cloud depends on the machine operation cycles. But, according to the data collected so far, we can extrapolate that it averages 300 MB per working day.

The application installed on the CNC also receives data directly from some of the sensors that were installed for maintenance-specific proposes and integrated with the PLC module. In particular, an oil sensor was installed, to monitor the presence of waste material and air bubbles in the machine oil.

The machine also contains wireless sensors which are able to collect information from its moving parts, particularly the ram and the back gauge (a machine subsystem which is capable of positioning metal sheets to be bend). This data can be used to detect problems with the ram guiding systems. Communication with these sensors is established via Bluetooth Low Energy (BLE) protocol and these data are gathered on the MANTIS PC.

B. Edge Local

The Edge Local receives data from the machines in a factory and connects to the Cloud Middleware. In practice it operates as a gateway to the Communication Middleware, and thus implements a De Militarized Zone (DMZ) inside the factory. Currently, the implementation of the Edge Local is limited to the Data Broker, which already satisfies the security requirements.

From the security point of view, the Edge-Local connects to both the Local Middleware and Cloud Middleware with Access Control (Authentication, Authorization) using TLS (SSL) support. The chosen communication bus was RabbitMQ and the AMQP protocol.

C. Data Analysis

The Data Analysis logical block receives data from the Cloud Middleware, processes it (signal selection and segmentation according to a set of rules), uses the Prediction Models to detect any potential failures (also updating the models' parameters to adapt to any drift if/when required), and sends failure related information back to the Communication Middleware, which is then used by the HMI. The information produced is assigned a criticality level (alarm, alert, info). All the information is then stored in the DB module of the Communication Middleware.

Currently the analysis is focused only on a particular model of the press brake machine. Preliminary data analysis was performed on the CNC's data of this machine in order to identify and select those signals that showed greater potential in identifying failures. The final selection of these signals and the identification of the corresponding failures was done based on domain expert knowledge, which was encoded as a set of rules. The Prediction Models module is populated with detection models (using the selected sensor signals), which are then tuned to the specific machine under study.

Two problems were identified when studying the preliminary data sets (including ERP data that indirectly recorded machine failure via replacement parts procurement). First of all, the data sets were very unbalanced – failures rarely occur. On average we had a hardware related failure every 150 days. This can impact the performance of the models, both in terms of learning and prediction accuracy. The Matthews Correlation Coefficient (MCC) [21] metric was therefore selected to measure the prediction accuracy in terms of true and false positive and negative counts.

Another issue that was identified is related to the usefulness of the prediction intervals versus the prediction range (short term, medium term and long term forecasts), since these ranges determine if a particular model can be used for maintenance planning. Since failure rates varied significantly (some machine seemed to have failures only after about 500 days), we therefore opted to forego medium-term planning and focus solely on short term predictions – the goal is to detect failures as soon as they occur via on-line stream analysis.

Data processing and analysis does not require all of the data that is being constantly produced by the controller and sensors. More concretely, when the machine is stopped (for production task reprograming, changing the tools, resetting the safety devices or simply being idle), the data cannot be used to identify failures. The data loggers at the machine level are used to detect and send data only when the machine is in production mode. This strategy reduces bandwidth and storage usage considerably. The data is also buffered in order to reduce the communications costs even further.

In addition to this, the communications infra-structure also takes care of fusing data from various sources (controller signals, controller sensors, external wireless sensors, etc.). It not only ensures that the signals that are recorded are properly synchronized, but also records a correct time-stamp. These timestamps are used by the failure detection algorithm to label the signal indicating were the failure occurred. The HMI then uses these timestamps to allow a user to search for and inspect the signal I order to further diagnose the problem.

D. Human Machine Interface

All data retrieved from the machine and the Data Analysis logical block can be inspected through graphs and tables, provided by the HMI module (see Figure 3). Also, each user can be promptly notified of any event on the machine through a text notification.

In the current state the HMI allows defining a user role and a set of corresponding permissions that defines the actions it can perform while using the system. Examples of such user roles include Data Analyst and Maintenance Manager.

The Data Analyst role allows inspecting live streamed data collected from the machine, such as oil-flow and temperature sensors. The data is displayed in near real-time (Fig. 3). The Data Analyst role also allows the visualization of historical data by selecting the data variables to be shown as well as the desired time-frame.

The HMI also displays the results of data aggregation and calculation of statistics. Several descriptive statistics provide useful (albeit simple) indicators that support the decisions making by those responsible for maintenance and design. These

indicators are therefore available to the Data Analysts. The results of the machine learning algorithms are displayed when they generate alerts and alarms, but they can also be visualized as historical data.

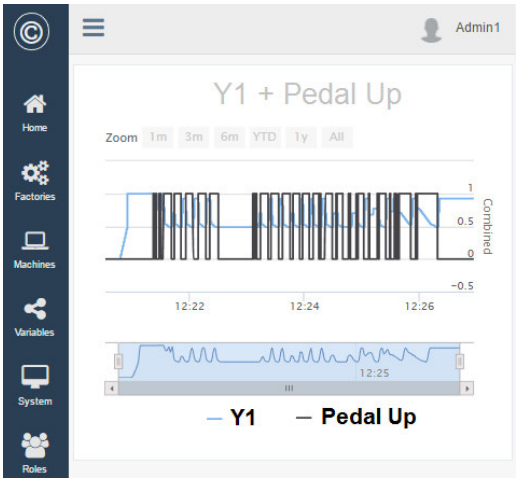

Figure 3 – Example graphic comparing pedals and positioning of the blade

The Maintenance Manager role allows to view some statistics, e.g. the type of components substituted and the frequency rate of the replacements. This should be specified for each monitored parameter according to the current number of cycles performed and to the maintenance actions of the machine tools. The user can also choose to display the results of the Data Analysis logical block, in the form of alarms, alerts and reports, which are displayed highlighting the relevant information for the maintenance manager and allowing the consultation of details on their provenance. These notifications include alarms that indicate unusual sensor data (for example based on simple statistics), unexpected behavior (for example, using outlier detection algorithms) and in the future warnings regarding impending failures (based on medium to long-term estimates). These notifications only allow detecting failures (corrective maintenance), but in the future may also be used to plan preventive maintenance tasks.

Security is implemented by means of SSL/TLS, for both the communication with the Cloud Middleware and the access to the HMI webpage (HTTPS). It is also important to note that the web-based HMI is running in the same node as Cloud Middleware, in order to reduce system complexity and to be able to access the same Database.

E. Cloud Middleware

The Cloud Middleware logical block provides all the benefits of the publish/subscribe communication paradigm [15], thus allowing the same data to be transferred to multiples recipients, maintaining information integrity across multiple systems, decoupling of consumers and producers, providing easier programming support, streaming capabilities, higher portability and scalability.

The middleware was configured using a topic exchange topology since it supports various publish/subscribe patterns that facilitate message routing. More specifically, this topology can route the same message to multiple queues by matching the message routing key and a queue pattern. Consequently, this allows sending the same message to all queues whose pattern matches the routing key. This is required because several modules must consume the same data. Figure 4 depicts the queues, the modules publishing to those queues and the module consuming from those queues.

Each factory (identified by its FID tag) monitored by this system has its own set of independent queues to where its data is published with different encryptions keys by each factory. The queuing system, for each factory, is composed by three queues (CPSSensorsFID, MachineSensorsFID, MachineInfoFID) and a unique topic exchange. This configuration allows the separation of data from different factories, which may belong to different owners. Thus coping with privacy concerns of the factory owners which do not allow sharing production-related information with other concurrent in the same markets. Additionally, it provides increased flexibility if in the future there is the need to support the middleware over a computing cluster – i.e. this solution supports the scalability of the system.

The MachineSensorsFID queue handles data from the internal sensors of all CNC machines in a factory. The MachineInfoFID handles data from the CNC machine databases. This database contains information about the operations performed by the machines and any event detected by its control program. The CPSSensorsFID queue handles data from sensors which are external to CNC machines, whose data is collected by the MANTIS PC module of the machine. In the case of the pilot described in this paper, sensors publishing information to this queue are wireless devices which collect information related to the moving parts of the machine, and are not integrated into the machine control system.

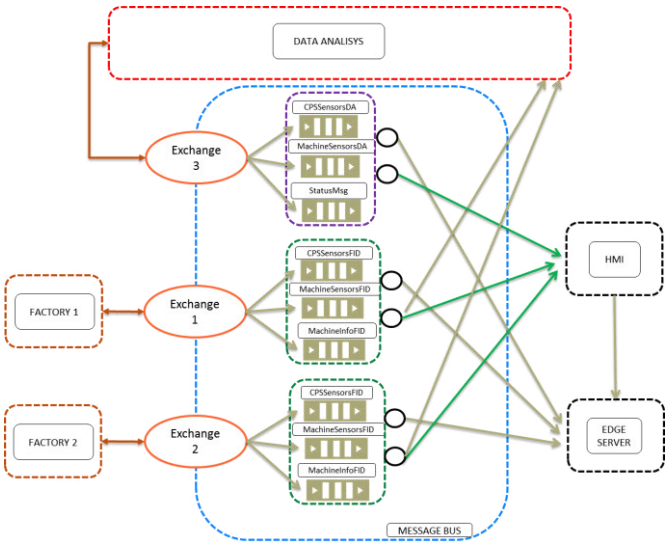

Figure 4 – Middleware queues

The communication infrastructure also comprises queues to feed data into the Data Analysis module and from this module

to the HMI. The Data Analysis logical block receives a copy of the data published to the CPSSensorsFID and MachineSensorsMID queues from each factory, which is used to run a series of different analysis on the data, as described in Section IV.B. The Data Analysis logical block publishes results concerning alarms and errors to the CPSSensorsDA and MachineSensorsDA queues. The StatusMsg queue handles data sent from the Data Analysis logical block to the HMI, convening information regarding the data that has been processed by the Data Analysis module. Note that this processing can be time consuming.

All data are stored on a database, structured upon the MIMOSA open standard for operations and maintenance in manufacturing [17]. Due to its volume, historical data is only stored for 3 months before being deleted or backed up.

The Factory ID (FID) parameter referred in each queue, identifies the factory from where data comes from. It is unique and allows, together with other credentials (user name, password, etc) for the Edge Local in a factory to connect to the middleware. Similarly, data consumers must also know the FID and connect with their own credentials.

Furthermore, the HMI is able to configure all these queues for each factory through a Representational State Transfer (REST) API, being able, for example to create a factory queue structure, as described above.

The queues are configured to temporarily retain messages for a given time to live (TTL) period of 3 hours, thus avoiding an excess of in-memory data if no subscriber reads the messages.

Also note in Figure 4 that the connections in green represent the usage of the STOMP over WebSockets protocol in order for the Web-based HMI to be able to connect with RabbitMQ. All other connections in gray color are supported over AMPQ protocol.

F. Local Middleware

The Local-Middleware logical block provides almost the same features and properties has Cloud Middleware regarding security, message Model, communication paradigm and topology.

The main server for this logical block is installed in each factory. It operates as a gateway allowing the connection between the Machine or Mantis-PC and the Edge-Local logical blocks. The queuing system, for each factory, is composed by three queues namely CPSSensorsFID, MachineSensorsFID, MachineInfoFID and a single topic exchange that uses the same FID tag between, for both the Local and Cloud Middleware in order to maintain information integrity across all logical blocks.

This middleware is based on the RabbitMQ, running the AMQP protocol. We are studying the possibility of using the MQTT protocol in the future to implement the communication system.

V. CONCLUSION

This paper describes the ongoing implementation of a pilot for proactive maintenance in Industry 4.0 for a Press Brake machine, within the aims of the MANTIS project. A higher level description of the architecture of the pilot that is used in the demonstrator is followed by details regarding the platform infrastructure. In particular, a detailed discussion describes how a message oriented middleware can be used to structure an Industry 4.0 application at factory level and, also, on supporting communications between the factory and the cloud where the data analysis is performed.

The current system still has room for improvement by supporting several enhancements to its architecture. Higher security levels can be attained by adding authentication between the cloud and the Edge Locals in each factory. Industrial applications are usually structured upon standards, where OPC-UA is one of the most adopted, and the organization developing it has recently released an enhanced version to support the usage of the AMQP protocol. Thus, the current plan for the pilot is to adopt the full OPC-UA protocol in our implemented platform, since this evolution would easily support the integration of the MANTIS solution with MES and ERP systems. More advanced preprocessing algorithms can also significantly reduce the amount of data to be transported by the communication middleware.

VI. ACKNOWLEDGMENTS

This research work was partially supported by national funds via the FCT Fundacao para a Ciencia e a Tecnologia and funds provided by the European Commission in the scope of ECSEL/H2020-662189 MANTIS RTD project (Project ID: 662189) and by the Portuguese Fundação para a Ciência e a Tecnologia (FCT, I.P.) in the framework of project UID/EEA/00066/2013 PEST (Strategic Plan for Science and Technology) for the Centre of Technology and Systems (CTS).

REFERENCES

[1] R. K. Mobley. An Introduction to Predictive Maintenance,. Butterworth-Heinemann, St. Louis, MO, USA, 2 edition, 2002.

[2] Faisal I Khan, Mahmoud M Haddara, Risk-based maintenance (RBM): a quantitative approach for maintenance/inspection scheduling and planning, Journal of Loss Prevention in the Process Industries, Volume 16, Issue 6, November 2003, Pages 561-573, ISSN 0950-4230.

[3] Jantunen, E, Zurutuza, U, Ferreira, L, Varga, P, "Optimising Maintenance: What are the expectations for Cyber Physical Systems", Cyber Physical Systems Week 2016 (CPS Week 2016 Vienna). 11, Apr, 2016, 3rd International IFIP Workshop on Emerging Ideas and Trends in Engineering of Cyber-Physical Systems (EITEC'16), Vienna, Austria.

[4] Albert H.C. Tsang, Andrew K.S. Jardine, Harvey Kolodny, "Measuring maintenance performance: a holistic approach". International Journal of Operations and Production Management, 19(7):691–715, 1999.

[5] A Kelly. Maintenance and its management. Conference Communication, 1989.

[6] European Committee for Standardization. EN 13306: Maintenance Terminology,. European Committee for Standardization,, Brussels, Belgium, 2011.

[7] Uday Kumar, Diego Galar, Aditya Parida, and Christer Stenstr¨om. Maintenance performance metrics: a state-of-the-art review. Journal of Quality in Maintenance Engineering, 19(3):233–277, 2013.

[8] Holmberg K. Jantunen E. Bellew J. Albarbar A. Starr A., Al-Najjar B. Maintenance Today and Future Trends. 2010.

[9] H. M. Hashemian, W. C. Bean. State-of-the-art predictive maintenance techniques. IEEE Transaction on Instrumentation and Measurement, 60(10):3480–3492, October 2011.

[10] Abdu Idris Omer, and M. M. Taleb. "Architecture of Industrial Automation Systems." European Scientific Journal, ESJ 10.3, 2014.

[11] R. Stevens, K. Kalaboukas and M. Forcolin, “Simple Sensor Network Middleware and FADs”, in Proceedings of the 17th International Conference on Concurrent Enterprising (ICE 2011), 2011, ISBN 978- 3-943024-05-0, 2011, pp. 1-9.

[12] John Soldatos, et al. "Openiot: Open source internet-of-things in the cloud." Interoperability and Open-Source Solutions for the Internet of Things. Springer International Publishing, 2015, pp 13-25.

[13] Jerker Delsing, Pal Varga, Luis Lino Ferreira, Michele Albano, Pablo Puñal Pereira, Jens Eliasson, Oscar Carlsson, and Hasan Derhamy, “The Arrowhead Framework architecture”, chapter 3 of book “IoT Automation: Arrowhead Framework” – ISBN 9781498756754, CRC Press Publisher, 2017, pp. 45-91.

[14] Tom Hannelius, Mikko Salmenpera, and Seppo Kuikka. "Roadmap to adopting OPC.UA." Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on. IEEE, 2008.

[15] Michele Albano, Luis Lino Ferreira, Luís Miguel Pinho, Abdel Rahman Alkhawaja, “Message-oriented middleware for smart grids”, Computer Standards & Interfaces, vol.38, pp. 133-143, Elsevier, 2015

[16] Joel L. Fernandes, et al. "Performance evaluation of RESTful web services and AMQP protocol." Ubiquitous and Future Networks (ICUFN), 2013 Fifth International Conference on. IEEE, 2013.

[17] Avin Mathew, et al. "A review of the MIMOSA OSA-EAI database for condition monitoring systems." Engineering Asset Management. Springer London, 2006. 837-846.

[18] Don W. Holley, "Understanding and using OPC for maintenance and reliability applications." Computing and Control Engineering 15.1 (2004): pp. 28-31.

[19] Ben L. Rapien, “Fundamentals of press brake tooling”, Hanser Gardner Publications, Cincinnati, 2005. Chap 3, pp 21

[20] Schuler , “Metal forming handbook”, Springer Verlag, Berlin Heidelberg, 1998

[21] Matthews, B.W. “Comparison of the predicted and observed secondary structure of T4 phage lysozyme”. Biochim. Biophys. Acta 405, 442-451 1975.

[22] Pal Varga, Fredrik Blomstedt, Luis Lino Ferreira, Jens Eliasson, Mats Johansson, Jerker Delsing, Iker Martínez de Soria, “Making system of systems interoperable–The core components of the arrowhead framework”, Journal of Network and Computer Applications,81,,85-95,2017,Academic Press, 2017.