VerSatilE plug-and-play platform enabling remote pREdictive mainteNAnce

Grant Agreement No	:	767561
Project Acronym	:	SERENA
Project Start Date	:	1st October 2017

Consortium	:	COMAU S.p.A. Finn-Power Oyj VDL Weweler BV WHIRLPOOL EMEA SpA Kone Industrial Ltd Engineering Ingegneria Informatica S.p.A. OCULAVIS GmbH SynArea Consultants S.r.l. DELL EMC Laboratory for Manufacturing Systems & Automation Fraunhofer Gesellschaft zur Förderung der angewandten Forschung VTT Technical Research Centre of Finland Ltd TRIMEK S.A. Politecnico Di Torino

Title	:	Design of versatile maintenance and planning
Reference	:	D3.1
Dissemination Level	:	PU (Public)
Date	:	2018-09-30
Author/s	:	VTT, LMS, POLITO, ENG, TRIMEK
Circulation	:	EU/Consortium

Summary:

Description of the SERENA AI Condition-based Maintenance and planning techniques. This deliverable is the main outcome of WP3, reporting completely the results of task 3.1 and partly the results of task 3.2 and task 3.3.

Contents

1 List of Abbreviations

CEN	European Committee for Standardisation
CWA	CEN Workshop Agreement
ICT	Information and Communications Technology
KPI	Key Performance Indicator
OEM	Original Equipment Manufacturer
PLC	Programmable Logic Controller
SME	Small and Medium Sized Enterprise
TRL	Technology Readiness Level
CBM	Condition based Maintenance
RUL	Remaining Useful Life
GUI	Graphical User Interface
SD	State Detection
HA	Health Assessment
PA	Prognostic Assessment
AG	Advisory Generation

2 List of Figures

3 Executive Summary

The purpose of this document is to report the outcomes of Serena WP3: AI Condition-based Maintenance and planning techniques describing completely the results of task 3.1 and partly the results of task 3.2 and task 3.3:

- Task 3.1: Design of condition-based techniques for versatile maintenance and planning.
- Task 3.2: Implementation of AI condition-based methods based on data mining and machine learning.
- Task 3.3: Scheduling and planning of maintenance activities for machine indicators improvement.

This document is aimed to provide the main technological solutions used in WP3 and the idea is to offer knowledge of the selected solutions mainly for WP2, WP4 and WP5 as well as to serve analytics solutions for WP6. The document is structured as follows:

- Chapter 1 describes the motivation of the project related to Condition-based Maintenance (CBM) and automatic diagnostics and prognostics using standardised data model MIMOSA. It also describes the objectives of the work package and requirements of the pilot use cases.
- Chapter 2 provides insight for system design covering theoretic background and principles related to standards ISO 13374-1 and ISO-13374-2. It describes the approach for the system itself and existing solutions and technologies to be used.
- Chapter 3 describes file and data management including description and purpose as well as database management system files.
- Chapter 4 discusses system interfaces solutions that are going to be built such as human-machine interfaces and external interfaces.
- Chapter 5 presents integrity controls such as internal security, data validation, verification for data handling (adding, deleting, updating, critical data management) and user verification, network identification, date/time, logs etc.
- Chapter 6 describes operational scenario of the system and its general functionality from the user's perspective. In addition, it shows how the proposed system should operate and interact (events, actions, etc.).
- Chapter 7 provides sum up, next steps and challenges.

The primary results of this deliverable include the following:

- Requirements have been collected from all pilots and are divided into three different categories.
- The presentation of the needed Serena AI predictive maintenance and planning system functions is given.
- The description of the operating CBM system is presented.

4 Introduction

4.1 Motivation

CBM is a widely accepted technique in industry today. It has been proven that it is possible to save money and decrease costs when using predictive health monitoring of machines instead of corrective maintenance that is concentrated only to change components when something really happens, and the machine is out of the use.

CBM is typically based on the idea that maintenance is made when there is a need. In practise, maintenance is performed after one or more indicators show that the equipment in question is going to fail or that the equipment performance is deteriorating [1].

Continuous condition monitoring system will take care of the CBM by continuously monitoring the system or the machine to produce the essential information that can be processed. Typically, the data is processed with the help of signal analysing methods and then the diagnosis of faults can be made. Finally comes the prognostics phase where the idea is to predict the Remaining Useful Life (RUL) of the component and to make recommendations of actions to improve it. In addition, it is important to make post mortems so that the same mistakes will not be made again.

In CBM a database system is needed because the information has to be saved to some place and there is a need to gather the information to the same place and make queries to get it in use. Operators, maintenance personnel, logistic managers, OEM's, parts suppliers, and engineers have always wanted to have information about the condition of production equipment readily available when they need it. Unfortunately, the categorized information typically is scattered among separate information systems, one for each platform and then separated by information type: manufacturer's nameplate data, as-installed data, as-maintained data, operational data, condition monitoring data (such as vibration readings, infrared thermography, oil analysis, control device monitoring, etc.), and asset diagnostic/health and reliability data [2].

It is difficult, if not impossible, to view the different information types on the same computer terminal, let alone compile and synchronize them into an integrated view or report on which to base intelligent asset management decisions. Even when the systems can be accessed from the same display, it usually requires customized efforts to integrate disparate databases, which use proprietary programs with different data models [2].

Interconnectivity of the islands of engineering, maintenance, operations, and reliability information is embodied in MIMOSA's Open Systems Architecture for Enterprise Application Integration (OSA-EAI) specifications. Previously, these separate information islands were built using specialized proprietary systems that provided value because they were optimized for a specific task or tasks, and they provided best results and value for those purposes. However, their combined value can be multiplied significantly, if they can be merged into an Open O&M™ information "data network" [2].

The "data network" can be building open OSA-EAI bridges to proprietary data stores to allow this information to be easily understood and utilized.

Existing solutions in maintenance section do not allow the easy reconfiguration without interrupting the production system. In order to improve the efficiency of maintenance activities, appropriate planning is required. Thus, inside WP3 a scheduling concept will be presented adjusting the execution of maintenance operations according to the production schedule or even modifying it if necessary.

4.2 Work package objectives

- Data analytics algorithms to accurately predict potential failures of the equipment;
- Hybrid approaches on data driven and physics-based models of the machine/equipment;

- Enable planning and scheduling maintenance activities in specific timeframes without interrupting the production process plan.

The WP improves existing solutions for predictive maintenance and the planning of maintenance solutions.

- Data analytics algorithms for predicting potential failures of the equipment or even aspects for being improved (e.g. some parameters).
- Hybrid approaches including both data driven and physics-based models of the machine/ equipment will be also considered in the cases when higher prediction accuracy is needed (e.g. critical situation, very sensitive machines, high-cost equipment etc.).
- Design and implementation of a decision-making component for enabling the scheduling of maintenance/repairing activities taking into consideration the production planning of the customer.

4.3 Requirements

The following requirements have been collected from all pilots and are divided into three different categories. Typically, prognostic applications are dependent on the use case, so the idea is to find customised solutions for each use case. Not all the pilots' requirements will be implemented but will be considered on a case-by-case basis. Table 1 presents cross-reference matrix between functionalities and pilots.

Serena AI predictive maintenance and planning system functions:

- Analysis of historical data
    - Machine learning
    - Correlation of measurement data with event data, e.g. failures
- Monitoring of real-time data
    - Detection and analysis of anomalies, alerting
    - Correlation of measurement data with a digital twin
    - Camera surveillance
- Predictive analytics
    - Model identification from historical measurement data, deploying a priori domain knowledge
    - Prediction of faults, developing performance degradations and service needs
- Maintenance aware operations planning and scheduling
    - Identification of maintenance needs
    - Best fit of maintenance plan to:
        - Existing production plans of the factory owner
        - Existing personnel availability and cost of maintenance provider company

The proposed scheduling concept consists of multiple modules and components, all combined in a way to find the optimum solution in order to assign the maintenance activities alongside with the production plan. In order to achieve this, some requirements are needed from the maintenance provider and the end-user perspective.

Table 1. Cross-reference matrix between functionalities and pilots.

Identified requirements	Use Case 1	Use Case 2	Use Case 3	Use Case 4
Machine Learning		X		
Correlation of measurement data with event data, e.g. failures	X	X		
Detection and analysis of anomalies, alerting		X		X
Correlation of measurement data with a digital twin			X	
Camera surveillance	X			
Model identification from historical measurement data, deploying a priori domain knowledge		X		
Prediction of faults, developing performance degradations and service needs	X		X	X

A list of these requirements is presented below:

- **WorkCentre:** This requirement defines the physical place where a maintenance or production activity will take place e.g. an assembly station. The equipment of a WorkCentre is used by resources.
- **Resources:** These are responsible for executing a task in a specific workstation. E.g. maintenance engineer, maintenance operator.
- **Jobs:** These are the sets of activities that take place in a certain period in the workstation. E.g. maintenance of the robot's belt gates. A job is comprised of tasks. Each job has a set of constraints such as completion time, assembly sequence, maintenance engineer unavailability.
- **Tasks:** e.g. visual inspection, check belt frequency
- **Dependencies between Tasks:** This means that the tasks can be modelled in steps. E.g. Welding Part 1, Welding Part 2, etc.
- **Suitability:** This requirement explains under which circumstances the resources are considered suitable for a task. E.g., a worker may be suitable for a welding task while the robot may be suitable for a pick & place task.
- **Criteria:** The chosen criteria are a set of rules that quantify a solution, measuring how good or how bad it is. For example, some used criteria are the cost, time or quality of a solution.
- **Task Duration:** E.g., the visual inspection lasts 10 minutes.
- **Context:** E.g. what events trigger the tool execution? Are the previous outputs decisions of the SERENA Scheduling Tool considered final or are the rescheduled? The context requirement is highly dependent on the pilot case of the SERENA project.

5 Predictive analytics and maintenance planning design

5.1 Concept

Rule-based systems also called production systems or expert systems are the simplest form of artificial intelligence. Rule-based systems are simple models and can be adjusted and applied for a number of kind of problems. A rule-based system uses rules as the knowledge representation for the knowledge coded into the system [3]. The definitions of a rule-based system depend almost completely on expert systems, which mimic the reasoning of human expert in solving a knowledge intensive problem. It represents knowledge in terms of a set of rules that express what to do or what to conclude in various situations. A rule-based system is a way to express a human expert's knowledge into an automated system. It can be simply created by using a set of statements and a set of rules that specify how to act on the statement set. Rules are expressed as a set of if-then statements (called IF-THEN rules or production rules) [4]. The larger the number of rules, the more comprehensive the artificial intelligence is.

The following example is a rule-based information model based on ISO-13374-1 and ISO-13374-2, called MIMOSA:

The Machinery Information Management Open Systems Alliance (MIMOSA) is a not-for-profit trade association composed of industrial asset management system providers and industrial asset end-users. It includes standard open source MIMOSA OSA-EAI and MIMOSA OSA-CBM. MIMOSA OSA-EAI presented in Figure 1 is a relational database that includes pre-designed domains such as registry, condition monitoring, reliability, maintenance and work management functions. MIMOSA OSA-CBM standardizes the moving of information in a CBM system. It describes the six functional blocks of CBM systems (Figure 2), as well as the interfaces between those blocks [2].

Condition Monitoring – Information management – Build on Open Standards

- MIMOSA (Machinery Information Management Open System Alliance)
    - Used by e.g. Boeing, Rockwell, B.P., U.S. Navy, VTT etc. [2]

1. Based to the standards ISO-13374-1, ISO-13374-2

- ISO-13374-1: Condition Monitoring and Diagnostics of Machines. Data Processing, Communication and Presentation. PART 1: General Guidelines
- ISO 13374-2: Condition Monitoring and Diagnostics of Machines. Data processing, Communication and Presentation PART 2: Data Processing

2. MIMOSA OSA-EAI

- Open System Architecture for Enterprise Application Integration
- Primary domains are registry, condition monitoring, reliability, maintenance and work management functions

3. MIMOSA OSA-CBM

- Open System Architecture for Condition-Based Maintenance
- Implements ISO-13374-1
- Harmonized with OSA-EAI

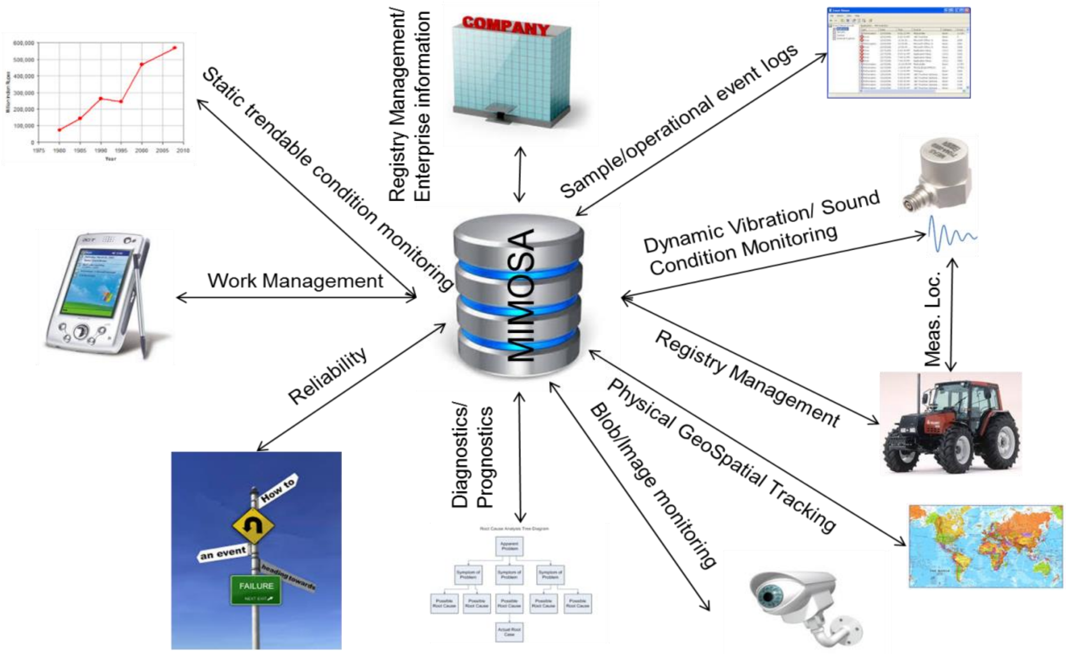

Figure 1. MIMOSA data model diagram.

Figure 2 presents the standardized CBM process where the MIMOSA database can be used when saving the data or making queries [5], [6].

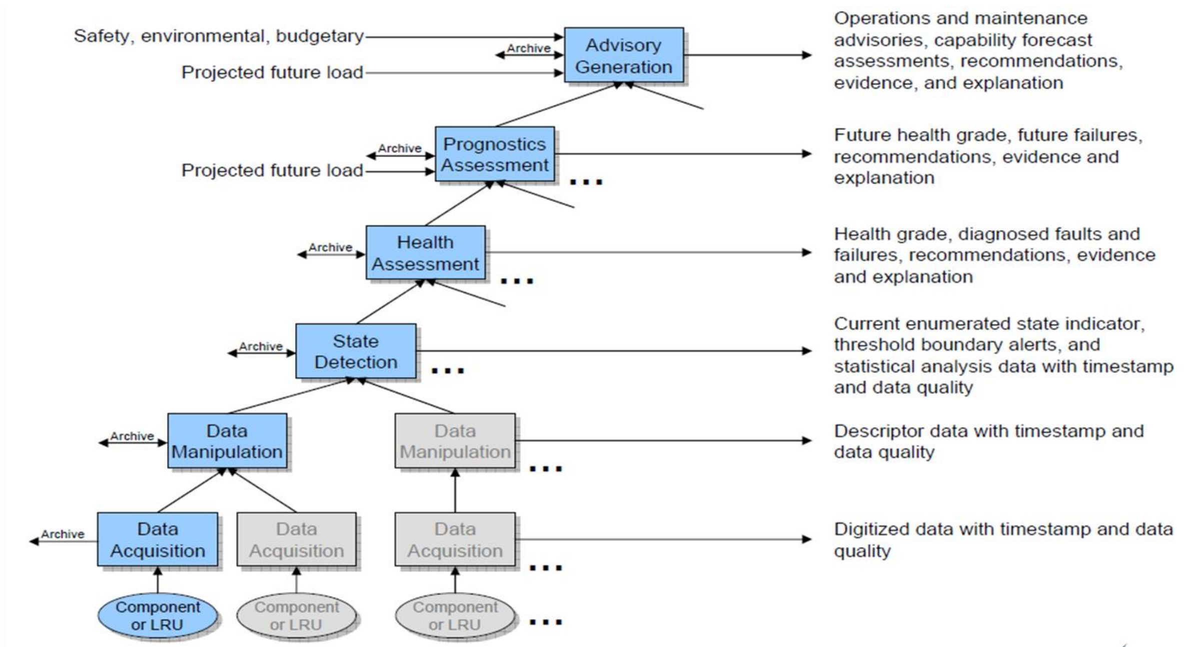

Figure 2. Data-processing and information-flow blocks. [5, 7]

The technology-specific blocks and the functions that will be used are as follows: [5]

- Data Acquisition (DA) block WP2
- Data Manipulation (DM block) WP2
- State Detection (SD block)
- Health Assessment (HA) block WP3
- Prognostic Assessment (PA) block
- Advisory Generation (AG) block

In the SERENA project, Mimosa is used for storing metadata, for retrieving data from a database and for exporting it to a database. In addition to this, functional blocks can be used to build a standardized completely automated maintenance system that works by utilizing artificial intelligence based on rule-based system. All of the needed actions e.g. scheduling and physical geospatial tracking can be taken with MIMOSA but some of the use cases are implemented using the following scheduling tool.

In addition, SERENA project provides the design of the scheduling components that will be used during the maintenance activities of the SERENA project. The proposed scheduling concept supports two different categories: a) factory user scheduling and b) maintenance facility scheduling. The first one refers to the scheduling from the end-user perspective while the second one refers to the maintenance provider scheduling. In the first case, the maintenance personnel are on premise but in the second one, they are outside of the factory department.

The scheduling approach of the maintenance operations will consider:

- The availability of the machines used in production
- The time required for the maintenance activities to complete
- The availability of maintenance personnel
- The production running and not running trade-off

The SERENA Scheduling Tool will consider the above components and will be responsible for retrieving the production plan from a cloud repository. These data will be used in order to provide the most effective and sustainable solution to the customer regarding the scheduling of the maintenance activities. In addition, the SERENA Scheduling Tool will interact and assist a maintenance engineer or operator to have a clearer understanding of the different maintenance activities being executed from the production point of view.

5.2 Predictive analytics and maintenance planning in SERENA

The SERENA solutions for predictive maintenance will be developed using a component-based approach, having in the SERENA Cloud Platform (designed and developed in WP5) as the central component where all the outcomes from the different WPs will be consolidated and integrated. WP3 will contribute to this overall picture mainly with two components groups, the AI CBM tools (i.e. the Analytics Framework) and the planning related solutions. All these components will be developed on top of the SERENA Cloud Platform defined in WP5 and designed for the time being in D5.1.

Data analytics will be implemented in terms of predictive maintenance, as well as data mining techniques for the data stored in secured cloud repository to proactively predict an impending issue and then deliver recommendations to operations, maintenance, and IT departments to address the issue. This will enable the monitoring of equipment and processes to adopt proactive maintenance and repairing procedures rather than fixed schedule-based approaches leading to time and cost saving.

Data mining helps in extracting meaningful data and building the model. Model building is finding out the relationship that expresses how the change in one or set of variables affect the other variables of the system. The design and development of hybrid approaches combining physical models of intelligent and complex machines together with data-driven algorithms to effectively support smart predictive diagnostics (prognostics), thus identifying symptoms of imminent machine failure before their actual occurrence.

The decision-making framework will be developed and adapted in order to enabling scheduling of maintenance/repairing activities that will be in line with the production planning. Existing decision support and optimization algorithms for proposing the maintenance in different levels and resources. For this purpose, existing solution for decision-making support and optimization algorithms regarding planning will be further improved for making the best possible decision on maintenance scheduling aspects.

Figure 3 presents the overall SERENA system architecture. The following sections will provide further details on the inner functional decomposition and design.

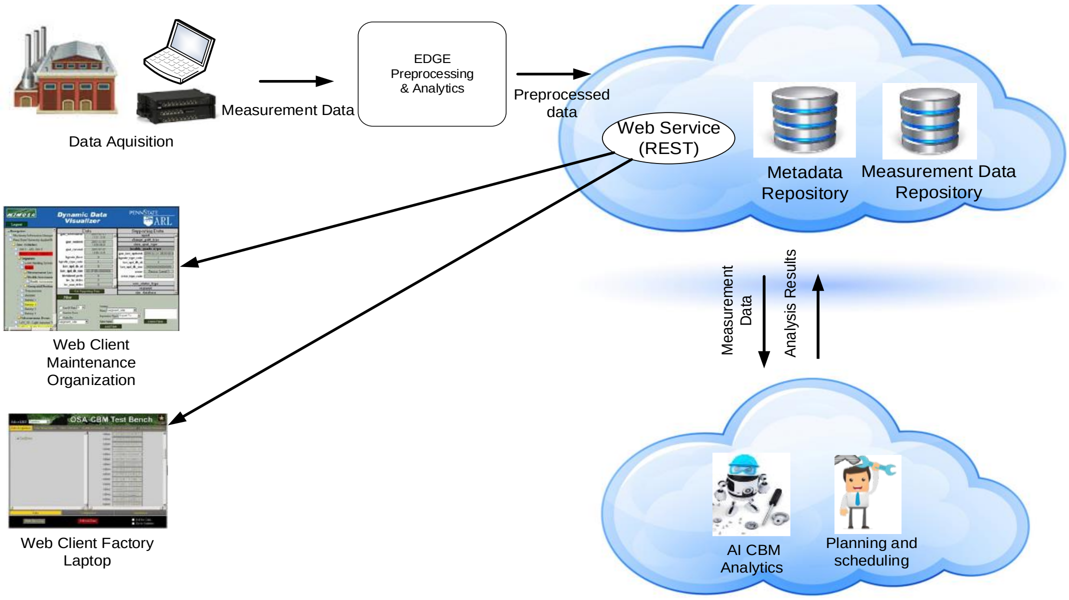

Figure 3. SERENA System Architecture.

5.3 Data analytics

Work package 3 work is related to techniques for CBM covering both diagnostics and prognostics features and scheduling and planning of maintenance. Python is used for creating data-analytics algorithms, machine learning etc. especially Numpy, Scipy and Scikit-Learn packages. Work package 5 will be responsible for the other modules of the cloud architecture.

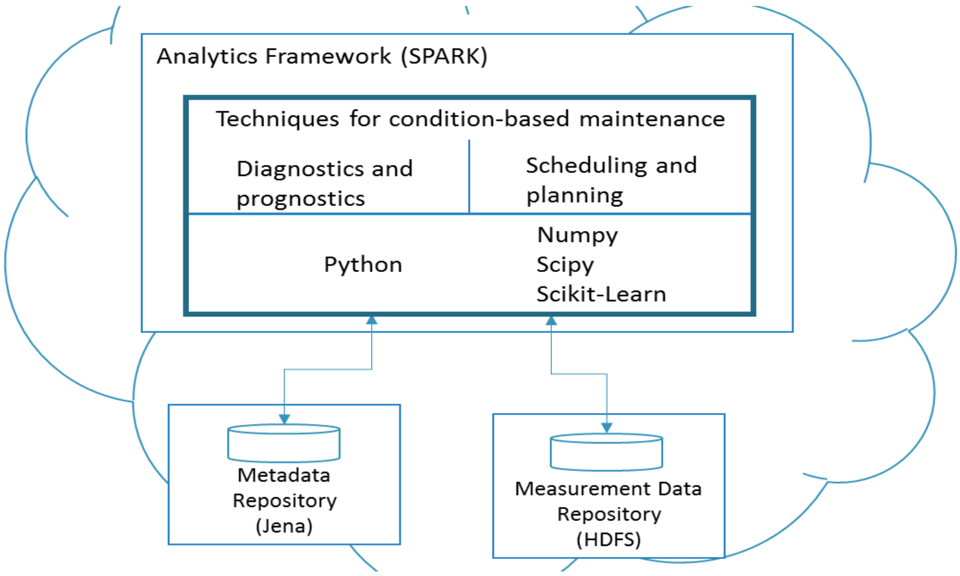

Figure 4. Analytics Framework (SPARK).

The SERENA Scheduling Tool consist of two main modules: a) the Scheduler Portlet Module and b) the Scheduler Operating Module. The Scheduler Operating Module is responsible to provide the necessary services in order to generate the best solution. In addition, it includes services for the communication with the other components of the SERENA platform and the data management. The Scheduler Portlet Module consists of multiple subcomponents, which are responsible for the interaction with the user and the functionalities that are offered by the UI. The SERENA Scheduling Tool is designed to facilitate the interaction with the users. The UI has been designed to depict the results of the subsystems and give access to the maintenance engineers. On the following sections, all the aforementioned modules and subcomponents will be further analysed.

5.3.1 Description

This section presents an overview of the proposed intelligent system designed to identify symptoms of imminent machine failure before their actual occurrence, hence implementing a smart CBM approach.

The design is based on combining physical models of complex devices (machines, robots, conveyors, etc.) together with data-driven algorithms to effectively support smart predictive diagnostics (prognostics).

With the aim of anticipating failures and estimating the RUL, innovative analytics methods, combining and customizing different state-of-the-art algorithms (e.g. neural networks, Bayesian models, Random Forests, etc.) will be designed and developed to forecast the future evolution of machine degradation through the characterization of the current dynamics of the process/machine (at any time) from the on-line data collected in the factory.

The proposed approach is designed to address some of the most common needs of manufacturing enterprises:

- Compatibility with both the on-premises and the in-the-cloud environments
- Exploitation of reliable and largely supported Big Data platforms
- Easy deployment through containerized software modules
- Virtually unlimited horizontal scalability
- Fault-tolerant self-reconfiguration
- Flexible yet friendly streaming-KPI computations
- The integrated provisioning of self-tuning machine-learning techniques for predictive maintenance.

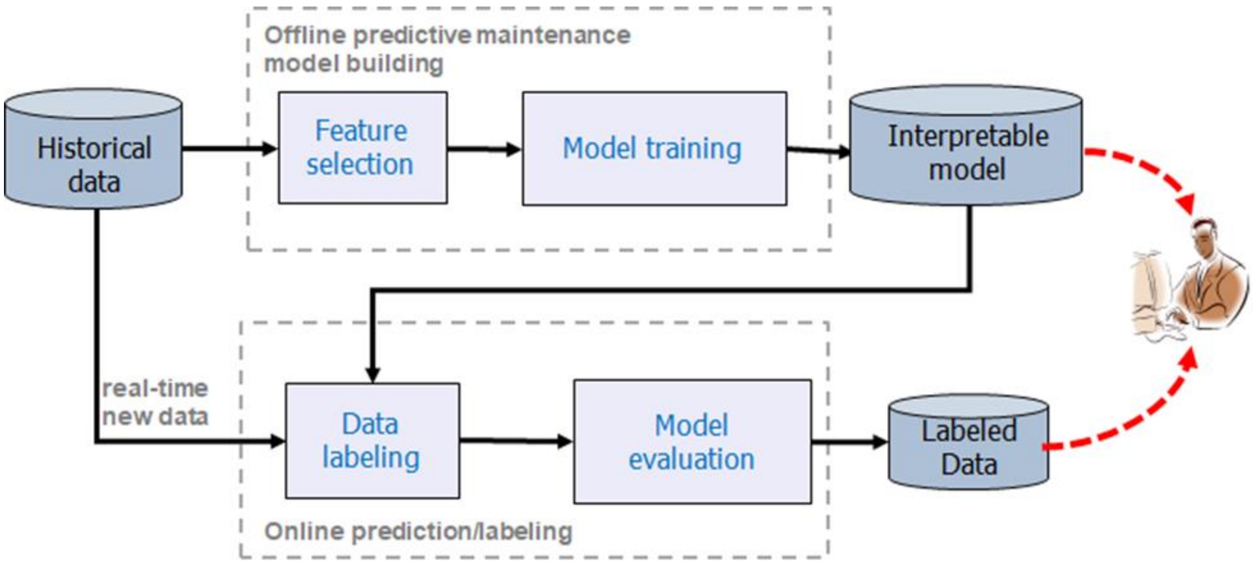

Figure 5. Overview of a data-driven machine-learning approach.

Figure 5 depicts the building blocks of a functional architecture based on data-driven machine-learning models. It exploits data mining algorithms to perform predictive maintenance strategies by managing, analysing and visualizing data of interest in an Industry 4.0 manufacturing plant.

The aim of the predictive maintenance approach is two-fold.

- Building a transparent prediction model based on historical data by means of machine learning algorithms.
- Applying such model in real time to new incoming data streams to identify possible failures.

Such approach consists of the following four steps:

1. Feature selection
2. Model training
3. Data labelling (health conditions and failure prediction)
4. Model evaluation (self-evolutionary feature of the model)

The feature selection and engineering block derives relevant static features from the most recent window of the input data time series, supporting the predictive maintenance goal. Among the many choices and challenges faced in this crucial task, an often under-evaluated requirement is the human-readability of both the extracted features and the predictive model, a requirement also known as transparency.

The understanding of the reasons for a possible failure and the link with the originally collected data is of paramount importance for the industries.

Apparently, in contrast with the academic quest for top prediction accuracy, industrial partners often prefer lower accuracy but more descriptive models, since such property translates into an actionable business advantage with respect to competitors.

Features will be selected to capture the meaning of most signals monitored in Industry 4.0. Results of the feature engineering are saved to the long-term storage database, ready to be used to train new updated models.

On a batch schedule, or driven by the model evaluation output, the model training block is executed on historical data. Historical data consist of:

- The original measurements
- The additionally extracted features for each time window
- The corresponding class labels (failure presence or absence)

All data are linked with an object of interest, the device or piece of equipment that can fail and whose predictive maintenance is desired.

The state-of-the-art is full of classifiers, each with its strengths and weaknesses: Nothing is usually better than the other in all conditions and for all data sets. For this reason, the approach designed for SERENA integrates different classifiers. The selection of the specific classifier is driven by the following purposes: Learning which input variables mainly predominantly guide the prediction problem and building an effective classification algorithm.

For this reason, the preferred algorithms in current preliminary experiments have been the following:

- Decision trees have been selected because of their interpretability.
- Multilayer perceptron neural networks and SVM have been chosen for their capability of separating classes with functions other than hyperplanes that are orthogonal to the dimensions available, as is the case with decision trees.

Finally, the prediction model is applied to new incoming data streams. The output is the prediction result, whose aim is to anticipate a failure event in the near future. Results are sent to the dashboard collecting all outputs from both the monitoring and the predictive blocks.

The Serena solution is designed with an optional model evaluation phase, which purpose is to provide a self-evolutionary feature to the overall system. Its aim is to address the requirement of detecting when a model trained on historical data is to be updated due to change in the type, distribution and characteristics of failures, machines, and processes under analysis.

Often, notable changes are known, and a model rebuilding can be manually triggered, e.g., when a machine is changed with a different model. Additionally, a weekly or monthly batch can routinely update the model with newly collected historical data to keep the model updated. However, such update can be itself triggered by a self-evaluation, which is able to assess the model performance over time and provide the end-user with a data-driven threshold automatically triggering the model training on new data.

5.3.2 Functionalities

The following part describes the six different functional blocks that the MIMOSA OSA-CBM standard defines for moving information in a condition-based maintenance system [5].

Data Acquisition

The data collection process is to change the real world analog quantities into digital form by sampling the measured sensor signal and converting it into digital format (ISO 13374-1). These signals can then be manipulated by means of signal processing as desired. Wireless technologies, such as 3G, 4G, WLAN and satellite have enabled data transfer almost all over the world and offered often cost-effective data communication alongside with communication by wire. Data is typically refined at the local server and then transferred to the maintenance information. The rapid development of computer, data acquisition, data transfer and sensor technologies have enabled more powerful and less expensive data acquisition for CBM.

Data Manipulation

Data manipulation part performs signal analysis, computes meaningful descriptors, and derives virtual sensor readings from the raw measurements (ISO-13374-1). The first step of data processing is data cleaning. Afterwards, some calculations are going to be made with that data to compute the useful information to be able to monitor the condition of the asset.

State Detection

State Detection (SD block): facilitates the creation and maintenance of normal baseline “profiles”, searches for abnormalities whenever new data are acquired, and determines in which abnormality zone, if any, the data belong (e.g. “alert” or “alarm”).

Health Assessment

Health Assessment (HA) block: diagnoses any faults and rates the current health of the equipment or process, considering all state information.

Prognostic Assessment

Prognostic Assessment (PA) block: determines future health states and failure modes based on the current health assessment and projected usage loads on the equipment and/or process, as well as RUL predictions.

Advisory Generation

Advisory Generation (AG) block: provides actionable information regarding maintenance or operational changes required to optimize the life of the process and/or equipment. This block obtains the information from the PA and HA and takes appropriate actions, e.g. make an order for a new component or send a notification to the manager.

Table 2. Cross-reference matrix between functionalities and technology-specific blocks.

Identified requirements	Data Manipulation	State Detection	Health Assessment	Prognostics Assessment	Advisory Generation
Machine Learning		X	X	X	
Correlation of measurement data with event data, e.g. failures			X	X	
Detection and analysis of anomalies, alerting		X	X		X
Correlation of measurement data with a digital twin		X	X	X	X
Camera surveillance		X	X		
Model identification from historical measurement data, deploying a priori domain knowledge		X	X	X	
Prediction of faults, developing performance degradations and service needs		X	X	X	X

5.3.3 Design Constraints

The most important design constraints are related to the following:

- Data-analytics solutions are case-specific and require customization in every use case
- The quality of the data in terms of its sufficiency, reliability and comprehensiveness
- Prognostic predictability regarding the RUL

5.3.4 Hardware architecture

Work package 3 operates in an environment defined by WP5 that operates in Docker environment. More information about hardware architecture can be found in D5.1.

5.3.5 Software architecture

Python is adopted as the programming language for this component. Python has built-in support for scientific computing: a numerical computation package called NumPy, scientific library called SciPy and multiple independent toolkits, e.g. Scikits-learn for machine learning. In addition, Python offers extensive open source libraries such as web frameworks, and testing instruments that makes it one of the largest ecosystems out of any programming community. The language is also widely taught in universities and it facilitates deployment on multiple platforms [8], [9].

5.4 Scheduling subsystem

5.4.1 Description

The SERENA Scheduling Tool consist of two main modules: a) the Scheduler Portlet Module and b) the Scheduler Operating Module. The Scheduler Operating Module is responsible to provide the necessary services in order to generate the best solution. In addition, it includes services for the communication with the other components of the SERENA platform and the data management. The Scheduler Portlet Module consists of multiple subcomponents, which are responsible for the interaction with the user and the functionalities that are offered by the UI. The SERENA Scheduling Tool is designed to facilitate the interaction with the users. The UI has been designed to depict the results of the subsystems and give access to the maintenance engineers. On the following sections all the aforementioned modules and subcomponents will be further analysed. Each one of them contains several components. The Scheduler Portlet Module is responsible for the functionality offered to the user via the User Interfaces that described in detail in section 4.

The subcomponents of the Scheduler Portlet Module are:

- Production Equipment Viewer
- Gantt Result Viewer
- Report Viewer
- Alarm Viewer
- Scheduler Services Client

The Scheduler Operating Module consists of the following subcomponents:

- Communication Traffic Handler
- Data and Information Normalization Handler
- Data Verification Handler
- Entity Mapping Handler
- Repository Handler
- Core Input Preparator
- Scheduler Core

5.4.2 Functionalities

This section is dedicated to describing the main functionalities of all the subcomponents of the SERENA Scheduling Tool.

The Production Equipment Viewer is the subcomponent responsible to visualize the health status of the production equipment. A graphical representation of the fault causes alongside with the quantifies metrics is provided for the user’s preferable machine.

The Gantt Result Viewer is the subcomponent responsible for the visualization of the result from the overall SERENA Scheduling Tool evaluation. More specifically a Gannt chart is provided for planning in an optimized way the maintenance activities to be performed alongside with the production plan.

The Report Viewer is the subcomponent, which is responsible to facilitate the depiction of a history of events (reports) that could either refer to the maintenance schedule or to the equipment itself. These reports are also user specific. For example, an experienced maintenance engineer may have access to specific reports that may be not visible to other maintenance operators.

The Alarm Viewer is a subcomponent implemented to visualize possible events of the SERENA Scheduling Tool or others coming from the SERENA cloud platform.

The Scheduler Services Client is the subcomponent that is responsible to invoke the functionality that is offered by the Scheduler Operating Module. The other subcomponents of the Scheduler Portlet

Module utilize the Scheduler Services Client to access functionalities that are offered as services from the subcomponents of the Scheduler Operating Module.

The Communication Traffic Handler is the subcomponent responsible for the managing of the incoming traffic from the different sources such as the SERENA cloud platform and distributes them to the Data and Information Normalization Handler subcomponent.

The Data and Information Normalization Handler is a subcomponent that normalizes all the various types of the user information into JSON objects for the Data Verification Handler to manage the data.

Data Verification Handler is a subcomponent that verifies that the data that was transformed into JSON object are in an appropriate format to be sent to the Entity Mapping Handler.

Entity Mapping Handler is a subcomponent that creates a mapping between the JSON object and entities that are corresponding to the database of the SERENA Scheduling Tool.

Repository Handler is the subcomponent that creates the connection between the Entity Mapping Handler and the database of the SERENA Scheduling Tool and passes the information to the database.

Core Input Preparator is a subcomponent that is responsible for the translation of the data coming from the SERENA Scheduling Tool database to the desired form as for the Scheduler Core to further work on them. The Core Input Preparator consists of three main steps:

- Retrieve Input Values from the database
- Create plain old java objects using the Hibernate ORM framework
- Create the desired core input format using the JAXB library in order to convert the java objects into XML

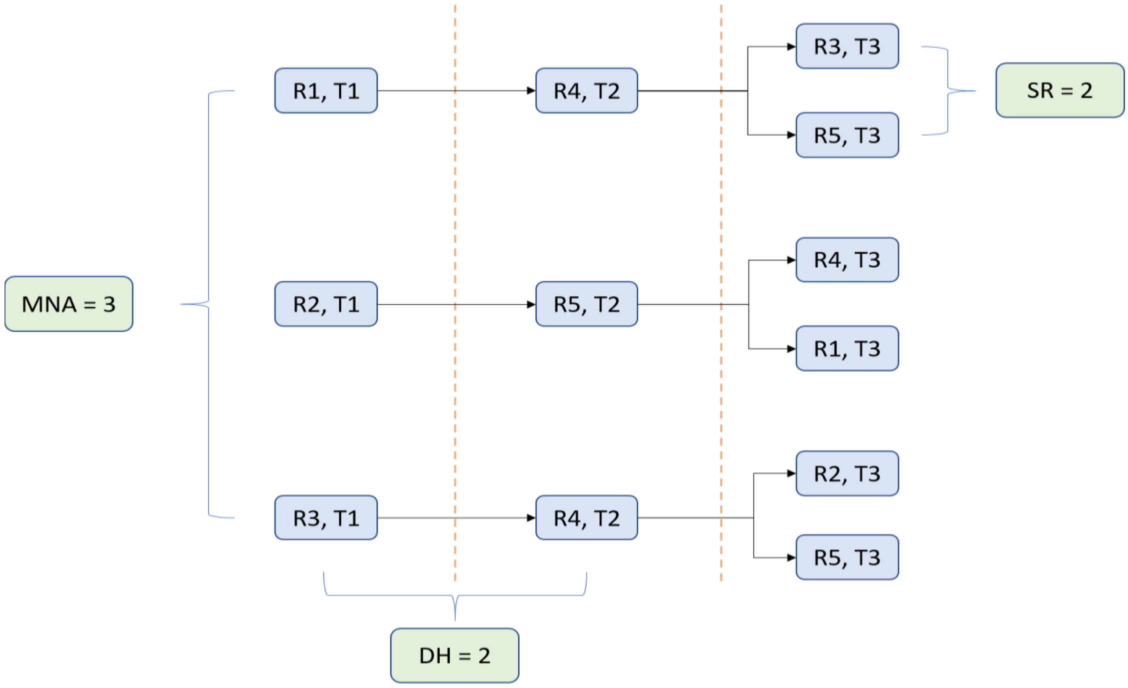

Figure 6. Intelligent Search Algorithm – Parameter Visualization

The Scheduler Core incorporates decision-making functionalities underneath with the purpose to identify the best solution for the scheduling of the maintenance activities with the production plan. To achieve this, it takes into consideration specific constraints and generates multiple assignments of the tasks to the available resources. In addition, the challenge is to combine multiple criteria simultaneously and try to fit the needs of the customer in order to reach an optimize solution. The weight of these criteria is getting defined and summed in order to calculate a utility value. The weights are estimated based on the production targets of each customer, such as minimize cycle time, production cost, energy consumption, etc. The utility value of each one of these alternatives is ranged between 0 and 1. The process of identifying the best alternative of these normalized values is being performed inside the Scheduler Core. This subcomponent contains an intelligent search algorithm in order to identify the best of the produced alternatives. This algorithm uses a three-parameter evaluation process to build its samples. These parameters are:

- MNA (Maximum Number of Alternatives): Defines the size of the grid
- DH (Decision Horizon): Defines the number of assignments that the algorithm will create
- SR (Sampling Rate): Defines the samples to be created

This process consists of four steps as described on the above section:

- Generation of the alternatives. One task is assigned to each available resource at a time and this can be easily adapted in the future.
- Choose of the desired criteria. It is very important to pick the correct criterion because this will quantify if an alternative provides a better solution than the other.
- Calculation of the sequence of the alternatives based on the desired criterion.
- Selection of the best alternative. The alternative with the highest utility value is chosen as the best solution.

Based on the above, two different criteria are chosen as basis for the initial design of the overall SERENA Scheduling Tool.

1. The first one is the total execution time of the tasks (T_R). This is estimated as the sum of the completion time of the assignment tasks both to the human maintenance operator and the monitored equipment. This criterion is based on the following equation:

$$T_R = \sum_{i=1}^{n} T_E^i$$

Where:

- T_E^i: execution time for a task i that is assigned to the human maintenance operator and the available equipment
- n: the total number of tasks that have been assigned to the selected resources

2. The second one is the overall cost of the selected resources, Ctotal, is estimated as the sum of the operating cost of each resource for each task. The equation describing the correlation is:

$$C_{total} = \sum_{j=1}^{n} C_R^j$$

Where:

C_R^j: operation cost of resource j that is responsible to perform each task

N: the total number of resources

The overall result of the SERENA scheduling Tool is the selection of the best alternative. This means that the solution includes the tasks assigned to specific resources alongside with the timing of this assignment and its duration. This schedule includes maintenance tasks fitted to an existing production plan of the customer. Depending on the case, the production plan may either change or remain the same.

5.4.3 Design Constraints

The most important design constraints are related to the following:

- Prognostic KPIs such as RUL as generated by the analytics component are received as the input triggering for the decision-making framework

- Maintenance service consumer existing production and/or maintenance plans
- Maintenance service provider plan and personnel availability and team costs

5.4.4 Hardware architecture

Work package 3 operates in an environment defined by WP5 that operates in Docker environment. More information about hardware architecture can be found in D5.1.

5.4.5 Software architecture

The scheduling tool is a web application designed as a client server system using Java supporting the connection to a central database containing the data required for planning/scheduling.

5.5 Internal communications

The analytics package communicates with the metadata and measurement data repositories of the Serena framework, see D5.1. Arrows A and B in Figure 7 depict this communication.

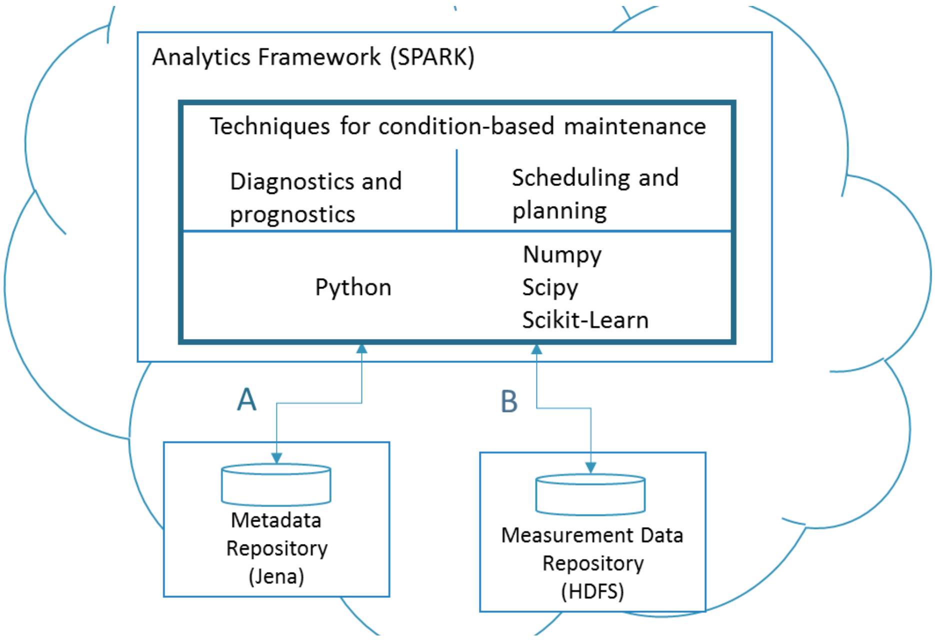

Figure 7. Communication between analytics module and data repositories.

Information from metadata repository (arrow A) consist of e.g.

- Measurement names, explanation information, units
- Analysis configuration data
- Analytics results

Information from measurement data repository (arrow B) consist of e.g.

- Vibration acceleration data
- Acoustic emission data
- Sound measurements

The SERENA Scheduling Tool will communicate with the SERENA cloud platform through RESTful services using JSON format for the exchange of messages. The input information from the SERENA platform will be received and manipulated by the Communication Traffic Handler as described in section 2.3.1. This kind of information will be related to production plan, RUL, etc. for each one of the use case scenarios and will be retrieved from a cloud repository. Apart from the cloud repository, the SERENA Scheduling Tool will have its own local database, which will be accessible to the other subcomponents through the Repository Handler as already described. The linking between the different subcomponents of the SERENA Scheduling Tool is visible in the following Figure 8.

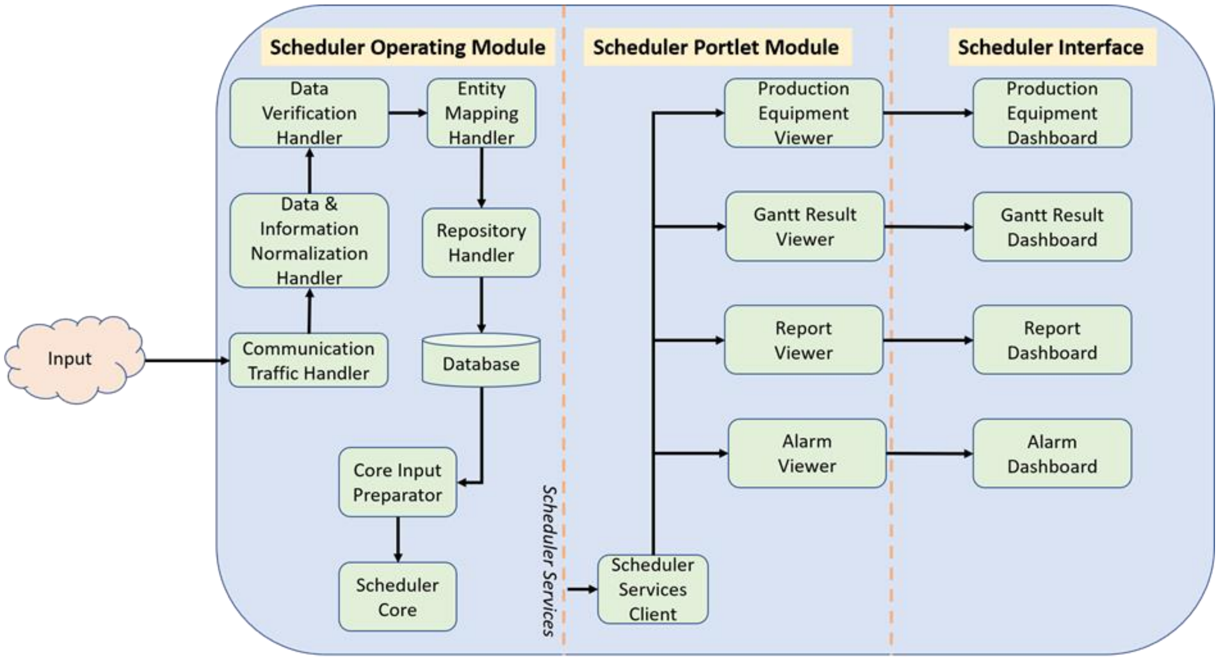

Figure 8. SERENA Scheduling Tool architecture.

6 File and data management

The purpose of this section is to describe the SERENA Scheduling Tool ontology that is being used by the SERENA platform as well as the data repository, which stores this ontology.

The role of this ontology is to model the concepts and information that are needed in the SERENA Scheduling Tool. The modelling of these concepts and information is required in order to implement the various functionalities offered by the SERENA platform such as the scheduling of maintenance activities. In addition, the database of the SERENA Scheduling Tool is needed in order to host the storing/updating of the collected requirements as described in section 1.3 through the form of data modelling.

Database management system files:
The SERENA Scheduling Tool Ontology consists of the SERENA Scheduler Data Model category, which contains several classes with certain properties and methods. Some of these classes are:

- Company

This class is a subcomponent class of the SERENASchedulingToolClass and it is responsible for the modelling of a company in term of the SERENA platform. Thus, it provides models of different companies, which may have different roles such as factory or maintenance provider. These different roles are also used to connect the companies to different aspects of the model. The Company class has its own members/subclasses such as:

  - Contact Information: This member contains all useful contact information of a company (e.g. telephone number, address, email, etc.)
  - Description: This member contains a short description of the company class. This kind of description may contain some information about the type of the company and their products.
  - Name: This entity is a subclass of the Company class and contains the actual name of the individual. This is more of a mnemonic and reference name and not a unique identifier.
  - hasCompanySize: This entity describes the size of the company in terms of employees.
  - FactoryRole: This subclass specifies if an individual company has the role of the factory inside the SERENA Scheduling Tool Ontology.
  - MaintenanceProviderRole: This subcomponent specifies if an individual company has the role of the maintenance provider.

The aforementioned subcomponents of the Company class are visualized through the Protégé software in the following Figure 9:

Figure 9. Company class subcomponents.

- ProductionResource

This class is a subcomponent class of the SERENASchedulingToolClass, which is responsible for the modelling of the resources of the production as define by the different use cases.

The ProductionResource class has its own subcomponents such as:

- Cost: This entity contains information regarding the cost of each maintenance tasks as performed by a specific maintenance engineer/operator.
- Description: This member class contains information in order to briefly describe a specific resource (e.g. industrial robot with 6 degrees of freedom). This description is used only to help human users and not for functionalities implementation.
- Resource Type: This member defines the type of a specific resource (e.g. human resource type or robot resource type)

These subcomponents of the ProductionResource class are depicted in the following Figure 10:

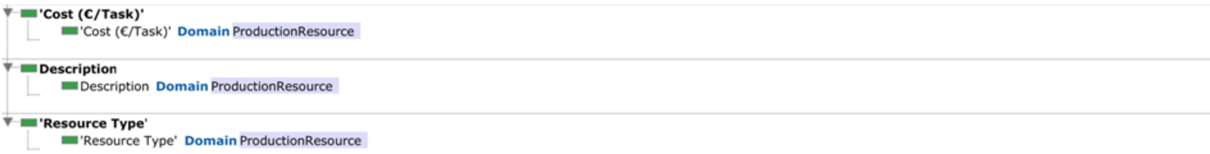

Figure 10. ProductionResource class subcomponents.

- Location

This class is a subcomponent of the SERENASchedulingToolClass and it is created in order to model the location data of each company.

The Location class has several subclasses such as:

- City: Defines the city the company is located
- Country: Defines the country of the company's premises
- Latitude: Contains the latitude value of the company
- Longitude: Contains the longitude value of the company
- Street Name: Defines the street name of the company
- Street Number: Contains the street number value of the company
- ZIP: Defines the ZIP number of the company

The aforementioned subcomponents of the Location class can be shown in the following Figure 11:

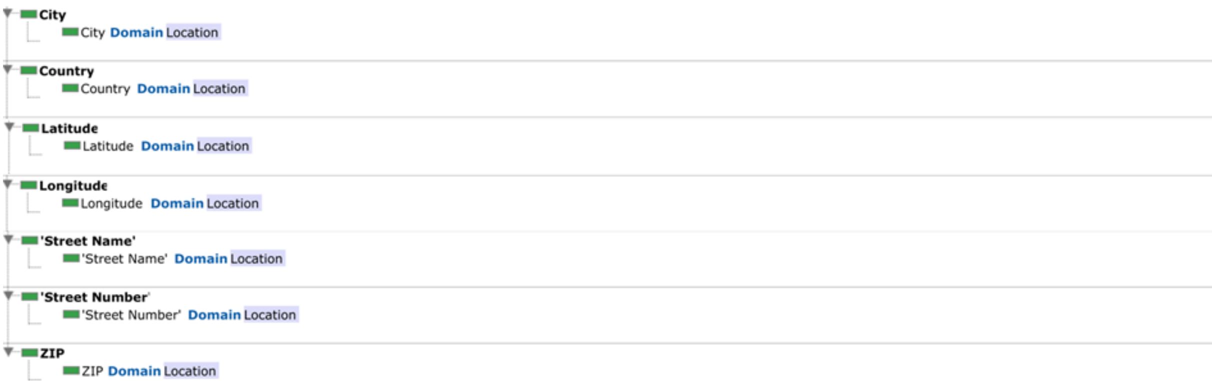

Figure 11. Location class subcomponents.

Configuration information is stored in the metadata database see Figure 4. Metadata database will be implemented in WP5.

7 System interfaces

7.1 Human – machine interfaces

Graphical User Interface (GUI) is produced by work packages 4 and 5.

7.2 External interfaces

7.2.1 Scheduler Interface

7.2.1.1 Purpose

The Scheduler Interface is the web interface of the SERENA Scheduling Tool and it is used for visualizing multiple information and results. A very first draft design of this interface is presented in this deliverable alongside with some possible mock-ups. The Scheduler Interface is divided in three different sections (tabs): a) production, b) reports, c) alarms. The information is provided through the functionalities of the Scheduler Portlet Module. In addition, the visualized content may be related to the main menu, the authentication details or related to the results from the Scheduler Operating Module. The Scheduler Interface consists of different components-dashboards depending on the type of the content to be visualized. Thus, the designed dashboards are:

- Production Equipment Dashboard
- Gantt Result Dashboard
- Report Dashboard
- Alarm Dashboard

Each one of these UI components are connected with a viewer which is included in the Scheduler Portlet. For example, the Production Equipment Dashboard is connected with the Production Equipment Viewer.

7.2.1.2 Inputs

The Production Equipment Dashboard is a subcomponent of the Scheduler Interface, which takes as input the status of the production equipment.

The Gantt Result Dashboard takes as input the decomposition of tasks in time and their assignment to different resources.

The Report Dashboard receives as input a list with all the recently reported events.

The Alarm Dashboard receives as input a list with all the raised alarms of the system or of a machine.

7.2.1.3 Outputs

The Production Equipment Dashboard provides a table with information about the health status of the equipment such as the RUL, OEE, etc. A list of the available machines is visible on the left column of the table and the user can select each one of them in order to see more detailed information. After the user's selection a click event is fired in order to display a separate view-holder on the right of the screen. The new displayed view contains graphs and charts both for the different KPIs over time and the fault causes for the selected machine.

The Gantt Result Dashboard provides a Gantt chart visualizing the result of the scheduling algorithm. This means that the different tasks are assigned in a sequence to the different resources inside a specific timeslot. In addition, another table is visualized on the right providing short information about the duration of a task, the maintenance tasks scheduled, and the production stops occurred. The Gantt Result Dashboard provides the user with the capability of rescheduling by clicking on the relative button underneath. More specifically, there will be cases that the user may be willing to change the resulted scheduling of the maintenance activities. Such cases may be that the selected criterion should be changed, or some constraint should be added (e.g. unavailability of the maintenance team). In these cases, the reschedule button provides this additional functionality. In addition, the simplicity of a button for such complex functionality gives the whole Scheduler Interface an extra user friendliness.

The Report Dashboard provides as an output information about the last reported issues in the form of a list. This list is updated each time something new is being reported. The visualized reports are being created either from the maintenance administrator, automatically from the machine's controller, or from the system itself.

The Alarm Dashboard provides information as an output about the last alarms. These alarms may be relative to machines' errors, alarm states raised by the system or the administrator.

The aforementioned UI subcomponents may be visible on the following mock-up Figure 12.

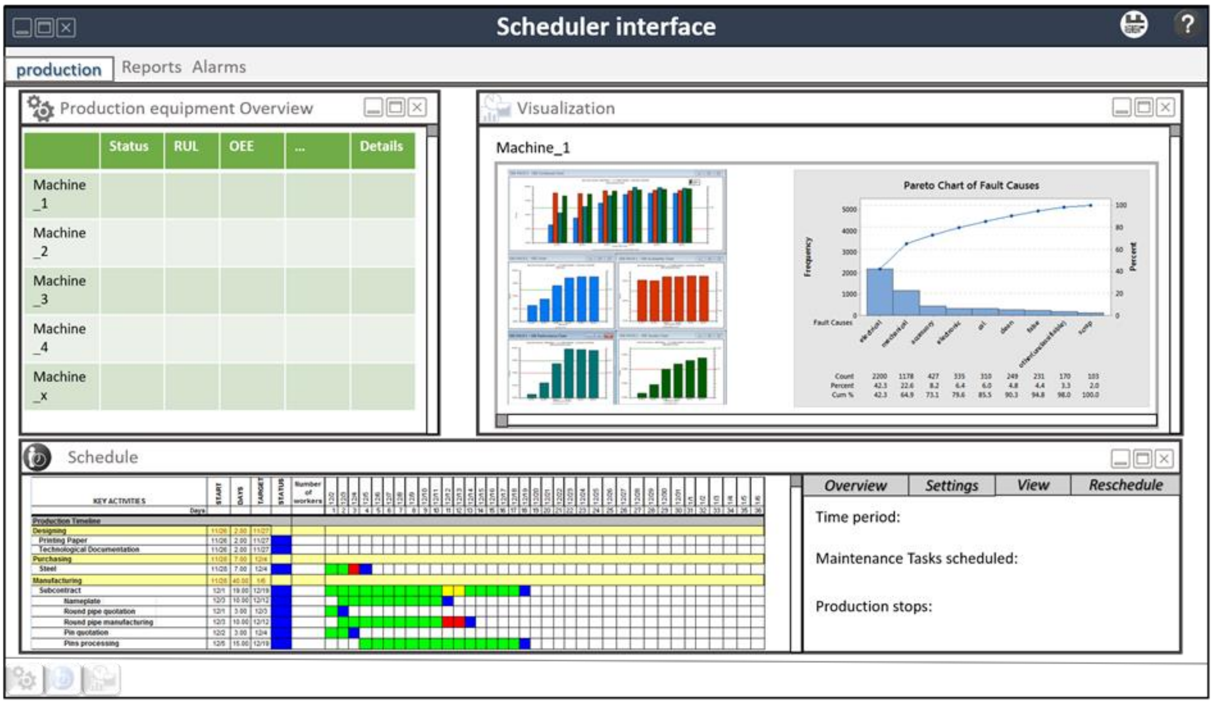

Figure 12. Scheduler Interface Mock-up.

8 Integrity controls

The SERENA Scheduling Tool contains a verification/authentication process through the Scheduler App Login Screen. Thus, the user can easily fulfil the credentials like username and password and press the Login button. Afterwards, a functionality is triggered, and the provided credentials are matched with a list of authenticated users in order to give access to the user. The Login Screen contains also functionalities for resetting the user's password or providing help in the case the user wants more detailed information in this phase. The Scheduler App Login Screen is depicted in the following mock-up Figure 13.

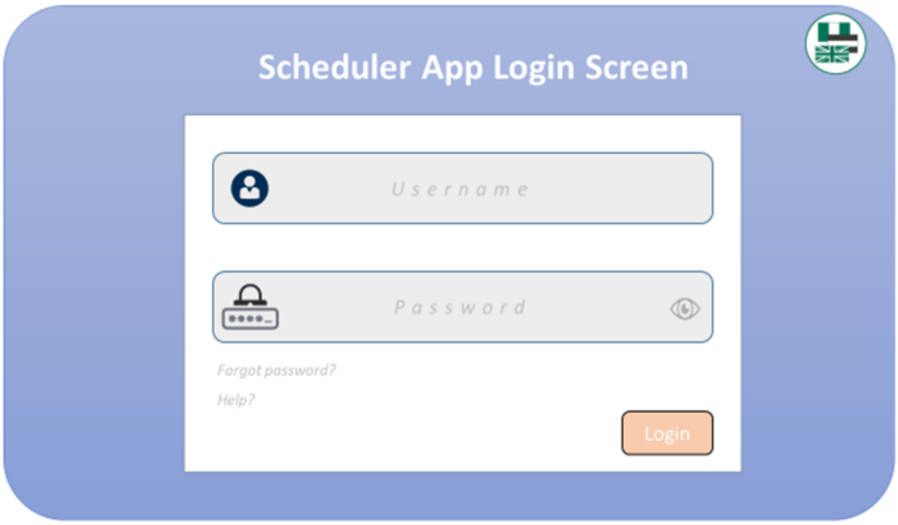

Figure 13. Scheduler App Login Screen Mock-up.

Data validation is intended to provide certain well-defined guarantees for fitness, accuracy, and consistency for any of various kinds of user input into an application or automated system. It is important to understand that data quality issues are strongly dependent on the characteristics of each application. Hence, data validation has to be customised for each use case. Data validation rules can be defined and designed using any of various methodologies and be deployed in any of various contexts. [10] Typically, data validation assumes that the data is normally distributed, but in these use cases, it cannot be assumed directly, so the idea is that in Serena project these physical based models are used for data validation.

Other integrity controls e.g. internal security, verification for data handling, user verification etc. will be reported in D5.1. Data validation will be also discussed in D2.1.

9 Operational scenario

In automatic diagnostics approach data is continuously acquired from factory floors then manipulated first on the edge and then further in clouds. Clouds also offer state detection of the machine and health assessment of it. Analysis results are transferred to the cloud services and if the situation is severe, the alert is produced for the maintenance engineer.

After alert maintenance engineer starts decision support via web client. This will activate decision support and send request for prognostics. Prognostics part will produce information of RUL and advisory generation will form O&M advisories and recommendation of actions to the maintenance engineer. Maintenance engineer will select actions and the system will finally produce work order for the maintenance staff.

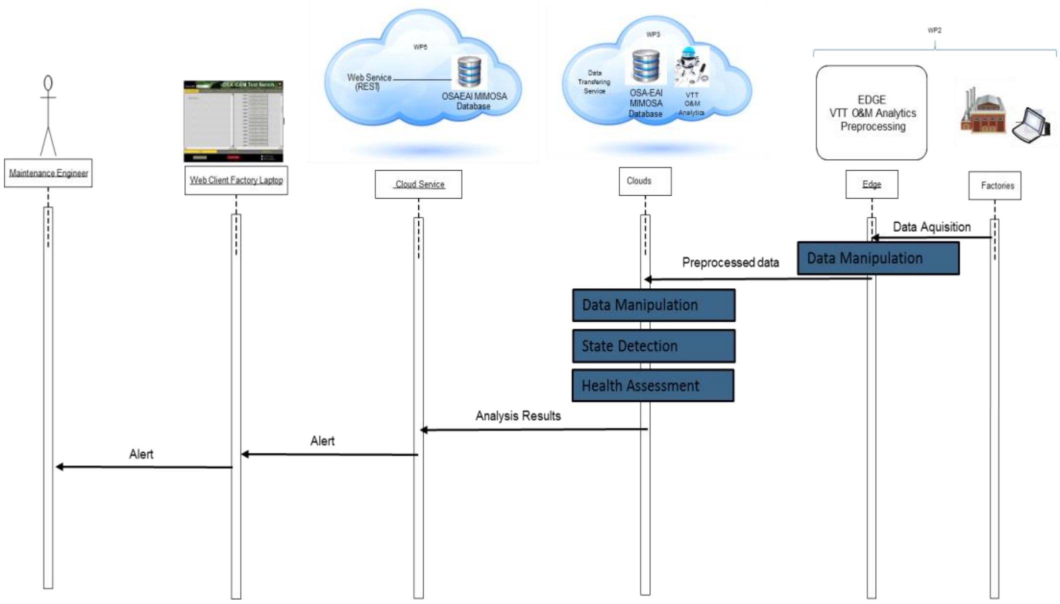

Figure 14. Use case description of automatic diagnostics (1/2).

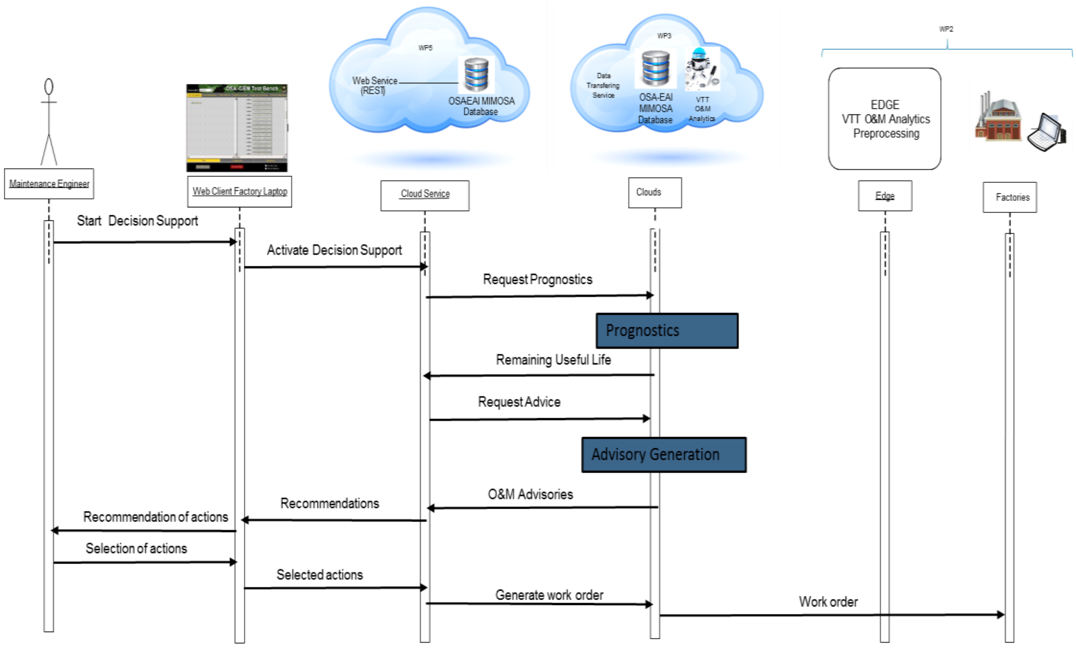

Figure 15. Use case description of automatic diagnostics (2/2).

10 Conclusion

This deliverable reports the design activities carried out within **SENERA** WP3 during the first twelve months of the project. Purpose of WP3 is to improve existing solutions for predictive maintenance as well as planning of maintenance solutions regarding data analytics algorithms and predicting potential failures on the equipment. Hybrid approaches including both data driven and physics-based models of the machine/ equipment will be implemented in the cases if higher prediction accuracy is needed. The production activities will be further considered enabling the scheduling of maintenance activities in specific time frames without interrupting the production process.

With respect to the aforementioned, the main conclusions of this deliverable are summarized as follows:

- The description of each **SERENA** use case requirements and the identification of its unique characteristics along with the data-analytics needs. All the needs are summarized to the Table 1 and divided to three main functionalities: Analysis of historical data, monitoring of real-time data and predictive analytics.
- The description of the AI CBM and planning system that will be provided by WP3 including system design, file and data management, system interfaces, integrity controls and operational scenario.

The characteristics of the AI CBM and planning system reported in this deliverable will offer knowledge mainly for WP2, WP4 and WP5 and finally be demonstrated in WP6 through the pilot cases demonstrators.

The most important challenges are related to the following:

- Some of the data-analytics solutions are case-specific and require customization in every use case, which means that resource adequacy, is a challenge.
- The quality of the data in terms of its sufficiency, reliability and comprehensiveness is another challenge.
- The quality of prognostic predictability regarding the RUL is also a challenge.

11 References

[1] Holmberg, K., Jantunen, E., Adgar, A., Arnaiz, A., Mascolo, J., Mekid, S., 2010. E-maintenance. doi:10.1007/978-1-84996-205-6

[2] www.mimosa.org

[3] Winston, P.H., 1984. Artificial Intelligence. ISBN 0-201-08259-4

[4] Grosan, C., Abraham, A., 2011. Intelligent Systems - A Modern Approach. doi: 10.1007/978-3-642-21004-4

[5] ISO 13374-1:2003 Condition monitoring and diagnostics of machines -- Data processing, communication and presentation -- Part 1: General guidelines

[6] ISO 13374-2:2007 Condition monitoring and diagnostics of machines -- Data processing, communication and presentation -- Part 2: Data processing

[7] The Pennsylvania State University, Applied Research Laboratory

[8] https://www.fullstackpython.com/why-use-python.html

[9] https://www.datacamp.com/community/blog/python-scientific-computing-case

[10] Zio, M.D., Fursova N., Gelsema T., Gießing S., Guarnera U., Petrauskienė J., Quensel von Kalben L., Scanu M., Bosch K.O., van der Loo M., Walsdorfer K., 2016. Methodology for data validation 1.0.