Production harmonizEd Reconfiguration of Flexible Robots and Machinery

Horizon 2020 – Factories of the Future, Project ID: 680435

Deliverable 4.3

Decision rules and KPI and Functionality Visualizations

Lead Author: Technische Universität Braunschweig

Version: 1.0
Date: 31.05.2017
Status: Preliminary
Dissemination level: PUBLIC

Version history:

Version	Date	Partner	Content
0.1	15.02.2017	TUBS	Table of contents and responsibilities
0.2	27.03.2017	TUBS	Chapter 1, 2, 3 and 4 added partially
0.3	26.04.2017	IPB	KPI monitoring with what-if game functionality added
0.4	04.05.2017	L'boro	Bayesian Diagnostics & Prognostics for Manufacturing Equipment added
0.5	05.05.2017	MTC	Data mining added
0.6	12.05.2017	TUBS	Review draft version
0.7	16.05.2017	MTC	Data mining updated
0.8	24.05.2017	MTC & L'boro	Update regarding restrictions by Siemens
0.9	29.05.2017	L'boro	Bayesian Diagnostics & Prognostics for Manufacturing Equipment updated
1.0	29.05.2017	TUBS	Adjustment in formatting & final version after feedback

Author list:

Name	Institution	E-Mail
Benjamin Neef	TUBS	b.neef@tu-bs.de
José Barbosa	IPB	jbarbosa@ipb.pt
Ali Al-Yacoub	L'boro	A.Al-Yacoub@lboro.ac.uk
Mostafizur Rahman	MTC	Mostafizur.Rahman@the-mtc.org

Abstract – Executive summary

With the transformation of existing production systems to more flexibility the demand of monitoring and tracing facilities increases steadily. Knowledge about the actual system state is essential for the safe and reliable operation of production equipment. Additionally the increasing development and integration of modern information and communication technology into the field of manufacturing opens up new possibilities like better understanding of production facilities but at the same time increases the complexity. For these reasons task 4.3 aims for the development and demonstration of proper monitoring and visualisation techniques and on tools for intelligent decision support. Hereafter different strategies for data driven decision support strategies and continuous monitoring solutions for the presentation of interchangeable and flexible production systems are proposed.

After a short introduction in chapter 1, chapter 2 gives an overview of the PERFoRM project and the integration of task 4.3 into the project objectives. In chapter 3 two solutions for the monitoring of flexible manufacturing systems are discussed. Chapter 4 shows up three solutions for decision support systems regarding maintenance activities. Two solutions use already existing data from maintenance reporting systems. The other solution relies on power-signatures.

I. Table of Contents

II. List of Figures

1. Introduction

1.1. Objective of the document

This deliverable contains the outcome of task 4.3, entitled "Automatic Monitoring and Visualisation of KPIs", which has two main objectives: On the one hand to develop new approaches for decision support on shop floor level with the focus on the improvement of industrial maintenance tasks. On the other hand the development of methods and tools for visualization and monitoring of KPIs regarding flexible production systems.

1.2. Structure of the document

The document is divided into two main aspects: One important aim within task 4.3 is the development of tools for monitoring and visualisation of existing production environments and corresponding KPIs to improve overall production efficiency in the context of an agile and flexible production environment. Corresponding solutions will be found in chapter 3 "Visualisation and monitoring". The other objective of task 4.3 aims at the development of tools for decision and maintenance support. Corresponding information can be found in chapter 4 "Decision rules and support".

2. Classification into general framework

2.1. Overview of the PERFoRM project

Within the framework of the PERFoRM project, the activities performed in WP4 specifically address methods and tools for simulation, visualisation and decision support in existing factories. WP4 is embedded in various PERFoRM activities like the design of standard interfaces for machinery and machining cells, enterprise systems and data backbones. The project is divided in 12 workpackages (WP). In WP1 the overall project-vision and objectives are defined. WP2-WP4 comprise mainly the development of technical solutions. The deployment planning and the validation of the developed system take place in WP5 and 6. WP7 to WP 10 address the use cases. WP11 and WP12 aim for dissemination and project management. **Figure 1** represents the overall PERFoRM paradigm. All services, tools and clients are connected via the PERFoRM Middleware. A consistent communication structure is ensured by the use of several adapters, wrappers and communication interfaces between these inhomogeneous peers. In this logical process flow the use of a common semantic is necessary and enables the utilisation of possible synergies that are present in the interaction between the above mentioned production and service systems.

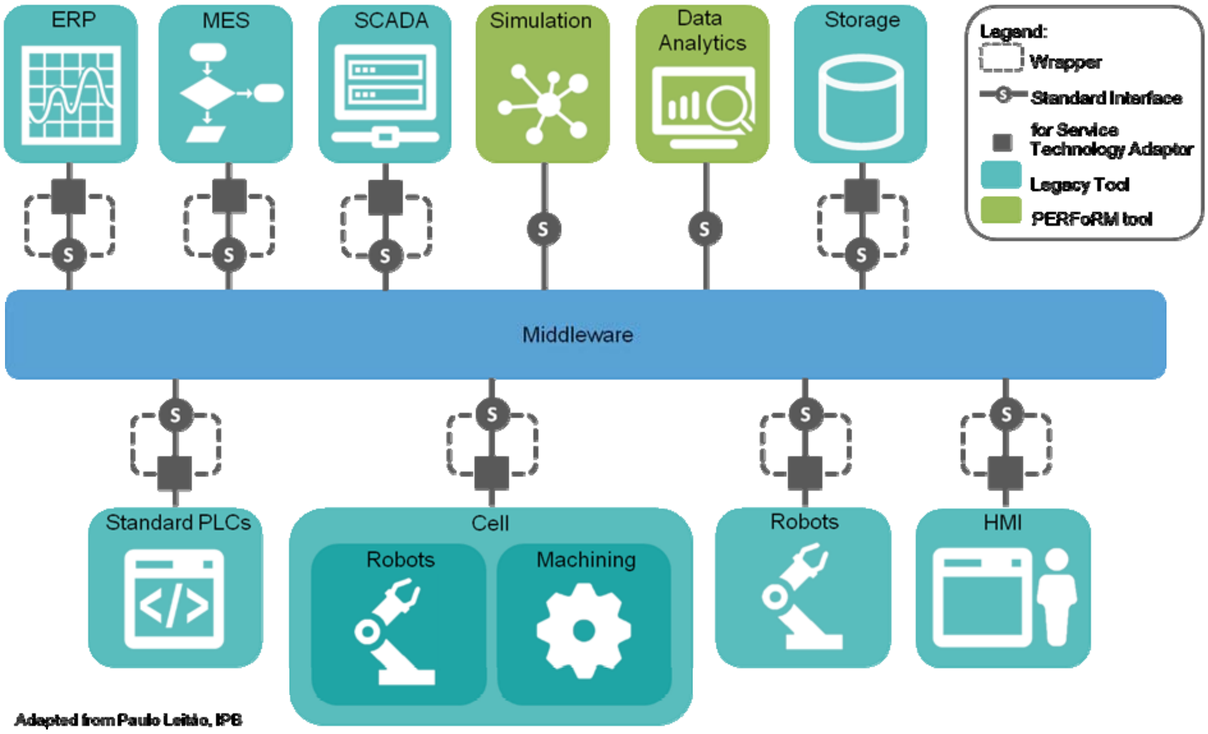


Figure 1: General PERFoRM framework

2.2. Task 4.3 overview

According to **Figure 2** the developments within task 4.3 are connected directly to the PERFoRM Middleware. Different solutions regarding visualisation and decision support features are implemented and operate jointly accordingly to the PERFoRM line of thinking. A more extensive description of tools developed in task 4.3 is provided in chapters 3 (Visualisation and monitoring) and 4 (Decision rules and support), while in this section only a brief description is given. Additionally to the detailed description in chapter 3 and chapter 4, the application of the solutions within the designated use case is described. A first item that has been identified as necessary is a flexible visualisation option. Two solutions addressing this issue were realised. Inspired by the philosophy of lean system architecture and a flexible system design two lightweight visualisation interfaces could be created. These tools can be found in the upper section of **Figure 2**. Both solutions work as a web visualisation and are accessible directly in a JavaScript capable browser. Hence, portability and adaptability are given. In the lower section of **Figure 2** tools for decision support are shown.

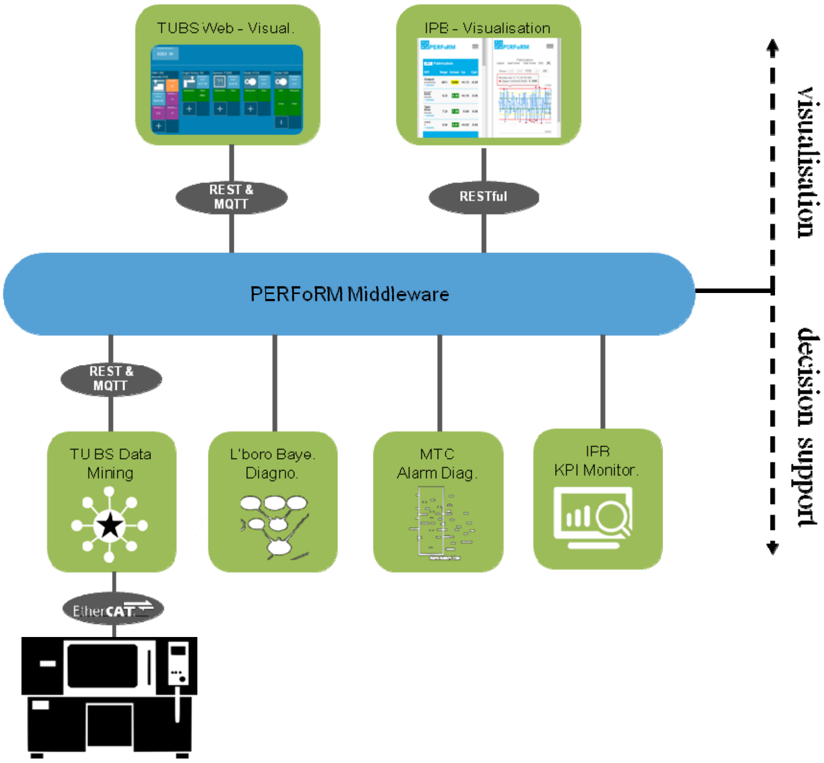


Figure 2: Task 4.3 visualisation & decision support framework

Beside the use case independent (technical) connection to the PERFoRM Middleware, the application and demonstration of the developed solutions to the use cases is one of the most important outcomes of the project. The matrix shown in **Figure 3** represents the allocation of the solutions developed in task 4.3 to the existing use and test cases. The matrix gives a brief overview of demonstration cases and the actual progress of the solutions within the project is represented.

Contributor	Solution/main goal	Siemens	eDistrict	Whirlpool	GKN	MTC	Smart Factory
Polimi	Value Stream Model in Excel						
IPB	KPI monitoring (with what-if-game functionality)						
TUBS	Min-Max Data Mining Toolbox						
MTC	Data Mining						
Lboro	Bayesian Diagnostics & Prognostics for Manuf. Equipm.						
TUBS	Universal web based KPI visualization						

Figure 3: Task 4.3 solution allocation matrix

2.3. Interfaces to other work packages / partners

Task 4.3 has diverse relations with other tasks and work packages, which are described briefly below. The interconnections were chosen partially on purpose in order to support an integrated view of the PERFoRM concept. As illustrated in **Figure 4**, this task considers the requirements defined in WP2 and WP3 as inputs to address the established main objective.

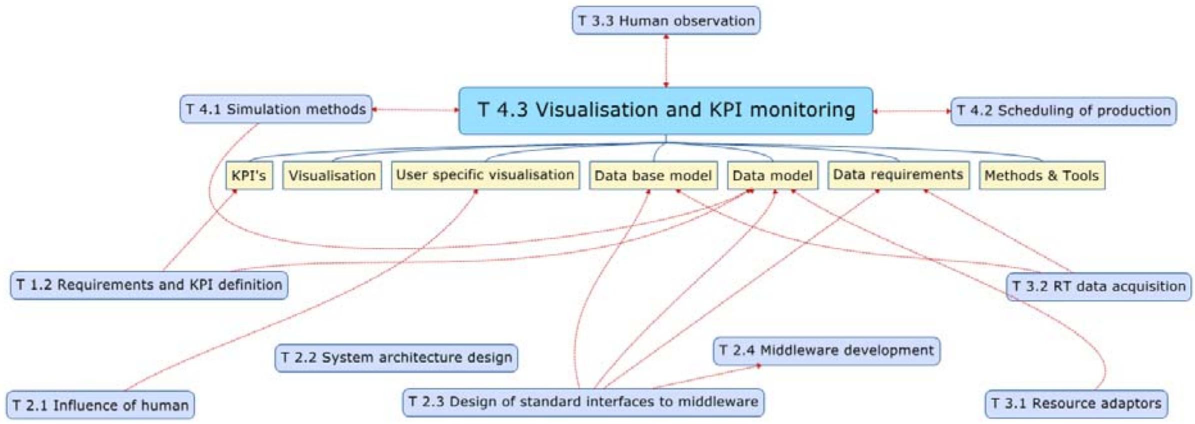


Figure 4: Interconnections of task 4.3 with other tasks and work packages

The results of this task will be used in different WPs for use case application:

- Planning of deployment within the PERFoRM use cases for WP5
- Validation and demonstration of developed methodologies within WP6
- Application of developed methodologies in industrial use cases will be performed in WP7, WP8, WP9, WP10

3. Visualisation and monitoring

3.1. KPI monitoring with what-if game functionality (IPB)

Target definition:
Dynamic monitoring of KPIs and KBFs for tactical and strategical decision support

The KPI monitoring with what-if-game functionality tool (KPI tool or What-if tool, for short) is a Web-based solution to support decision-making strategies, by monitoring Key Performance Indicators (KPIs), detecting trends and deviations, and performing what-if game based on the variation of Key Business Factors (KBFs) and generating the associated KPI implication. Architecturally, the tool is composed by the KPI Calculation & Statistical Quality Control Module, What-if-game Module, Data Service Module, Data Handler Module and Visualization Module, as depicted in **Figure 5**.

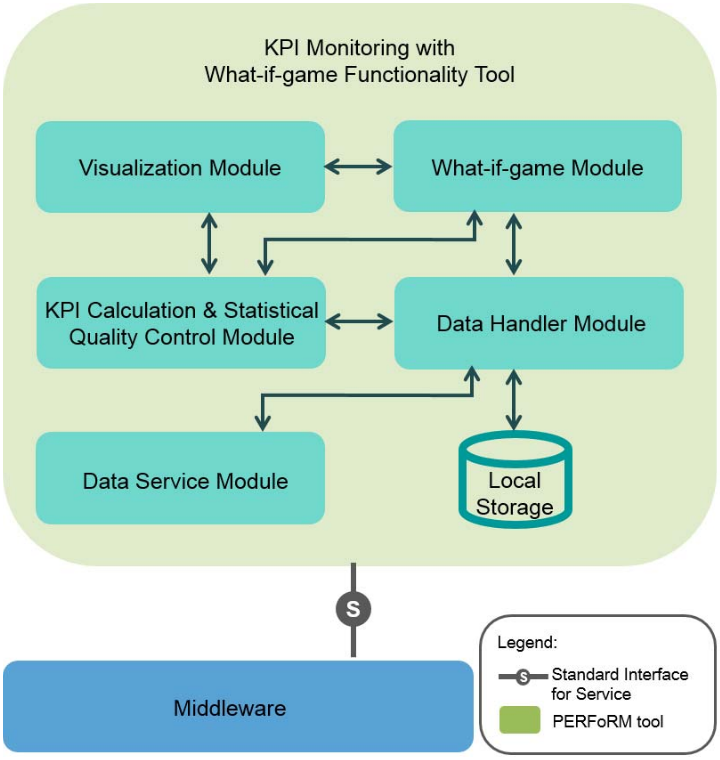


Figure 5: Architecture of the KPI monitoring with what-if-game functionality tool

To properly operate, the tool needs to access the production data, which in this case is available in a PERFoRM-compliant way, following the specified data model (as referred in Deliverable 2.3), by means of the use of a technological adapter that converts the native data structure into the PERFoRM data model (WHP use case, as showed in **Figure 6**) or by interact with a PERFoRM compliant tools (GKN use case).

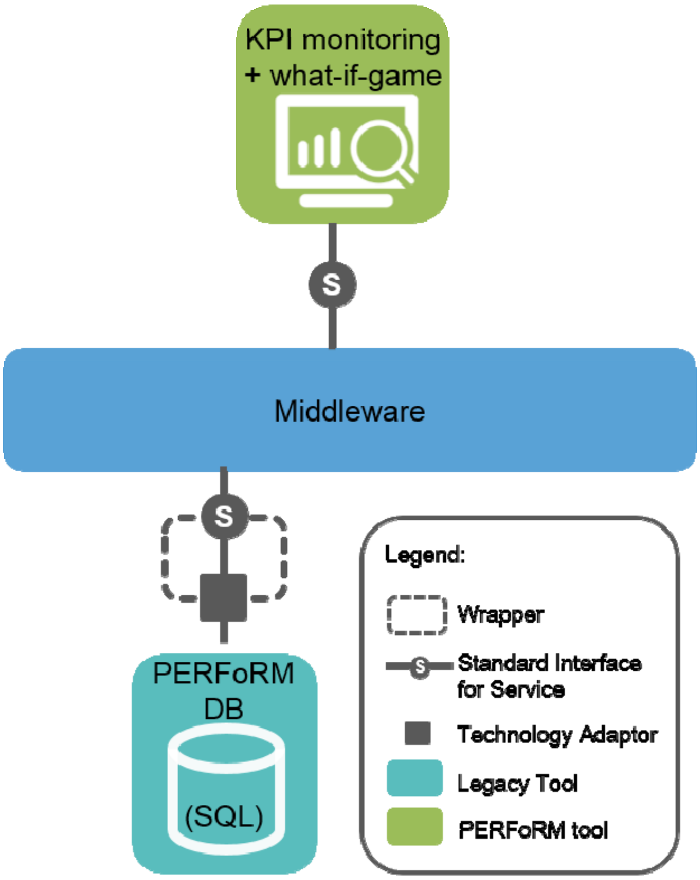


Figure 6: Positioning of the tool relative to the PERFoRM architecture concretization for the Whirlpool use case

Thus, from the point of view of the tool, the data access is transparent, executed by the Data Service Module, and operates on a request-reply manner, for a more static and historical data collection, or using a publish-subscribe mechanism, for a real-time data collection.

From the implementation point of view, a request–reply mechanism, using RESTful Web Services, implements the collection of the historical data, when needed. The following code excerpt, depicts an example of the used Java API for RESTful Web Services (JAX-RS) [1], focusing on the implementation of a getValue request to the industrial middleware.

```
public Collection<PMLValue> getValue(String tag, Collection<PMLParameter> parameters){
String uri = tag;
String param = "-";
for (Iterator iterator = parameters.iterator(); iterator.hasNext();) {
    PMLParameter p = (PMLParameter) iterator.next();
    param += p.getValue() + "-";
}
uri += "?param=" + param;
return ClientBuilder.newClient()
    .target(uri)
    .request(MediaType.APPLICATION_JSON)
    .get(new GenericType<Collection<PMLValue>>(){});
}
```

On the other hand, the on-line refresh of the data presented by the tool is achieved by using the publish-subscribe mechanism implemented using the Message Queuing Telemetry Transport (MQTT) protocol [2]. For this purpose, the Eclipse Paho client was used. The following excerpt depicts a topic subscription for a Quality of Service of 2.

```
import org.eclipse.paho.client.mqttv3.*
...
int qos = 2;
...
MqttClient client = new MqttClient(this.serverURI, this.clientId);
MqttConnectOptions connOpts = new MqttConnectOptions();
connOpts.setCleanSession(false);
client.connect(connOpts); // connection to broker
try {
    client.subscribe(this.topicFilter, this.qos);
} catch (MqttException ex) {
...
```

In addition to this remote data access, local data is required to support the continuous operation of the KPI Calculation & Statistical Quality Control Module. Therefore, the Data Handler Module manages the local repository and evaluates whether there is the need to trigger further data acquisition via the Data Service Module. Finally, the Visualization Module renders the information in a suitable way to be presented in web-based UI applications.

In order to best fit the description of the tool into the aspects previously defined in this document (on the one hand Visualisation and Monitoring and on the other hand Decision Rules and Support), the description of the KPI Calculation & Statistical Quality Control Module and of the What-if-game Module will be broken down by this section and section 4.

3.1.1. WHP case study

The main objective of the Whirlpool use case is to establish a real time monitoring system empowering the decision maker user with a tool for a continuous supervision of KPIs, using appropriate user interfaces (UI). Besides the responsive feature, this tool should be able to warn and alert the user for critical situations, namely those that at a bare eye would be difficult to oversee. This tool should also correlate crucial KPIs and KBFs, increasing the system reconfiguration responsiveness due to a faster assessment of the KPI-KBF interdependencies.

The KPI Calculation & Statistical Quality Control Module is responsible for the calculation of the KPIs, as defined and evaluated in [3]. For this purpose, the set of KBFs presented in **Table 1** are used as input while two different sets of KPIs are calculated allowing to assess the evolution of the undergoing industrial process.

Symbol	Description	Unit
$\mathbf{W_d}$	Number of working day during the year	d
$\mathbf{h_S}$	Hour for each shift	h
$\mathbf{S_d}$	Number of shift for each working day	
TD	Total Demand: number of pcs requested for each hour	pcs/h
CT	Cycle Time: time between the beginning and end of the process of making a product	s
OM	The number of operator/machine that are involved in the same working station	
NC	Non-conforming product	%
Av	Availability: percentage of production losses due to availability problem	%
P	Performance: percentage of production losses due to performance problem	%
ST	Set Up Time: the time required to set up a device for production of a new batch	s/batch
BS	Batch Size: the number of items that will be produced after a machine has been set up	pcs/batch

Table 1: Key business factors (KBF)

The first set, presented in **Table 2** (with the respective mathematical formulation), is composed of KPIs, related with the M monitored working stations.

Calculation formulas	Unit
$\mathbf{Total\ Cycle\ Time\ (TCT) = CT * OM}$	s
$\mathbf{Actual\ Daily\ Machine\ Availability\ (ADMA) = h_S * 60 * 60 * S_d * OM * Av}$	s/d
$\mathbf{Actual\ Processing\ Time\ (APT) = CT + ST * (1 + NC)}$	s/pcs
$\mathbf{Output = Throughput = ADMA * APT * P}$	pcs/d
$\mathbf{Lead\ Time\ (LT) = 1/Output * 3600}$	s
$\mathbf{Takt\ Time\ (TT) = \dfrac{h_S * 60 * 60 * S_d * W_d}{TD * h_S * W_d}}$	s

$\text{Overall Equipment Effectiveness (OEE)} = \mathbf{LT/TT}$	%
$\text{Time of Stock}_i = \left(\frac{3600}{LT_i} - \frac{3600}{LT_{i+1}}\right) * LT_{i+1}, i = 1, \dots, M-1$	s

Table 2: KPIs for each working station (calculation formulas)

On the other hand, the second set of KPIs refers to the overall system parameters (see **Table 3**).

Calculation formulas	Unit
$\text{Bottleneck Machine} = \max\{LT_m : m = 1, 2, \dots, M\}$	s
$\text{Output/Hour} = \frac{3600}{\text{Bottleneck}}$	pcs/h
$\text{Lead Time} = \text{Bottleneck}$	s
$\text{Theoretical Production Time} = \sum_{m=1}^{M} TCP_m$	s
$\text{Total Production Time} = \sum_{m=1}^{M} (LT_m * OM_m)$	s
$\text{Total Actual Lead Time} = \sum_{m=1}^{M} (LT_m * OM_m) + \sum_{m=1}^{M} \text{Time of Stock}_m$	s
$\text{Stock(pcs)} = \text{Batch Size} * \text{Item}$	pcs
$\text{Stock(hours)} = \frac{\text{Stock(pcs)}}{O_h}$	pcs/h

Table 3: KPIs for the overall system (calculation formulas)

The tool UI presents data following a production stations layout sorted order, displaying the several KPIs for each one of them **Figure 7** - left). The target and the actual KPIs values are shown, as well as their evolution trend (positive or negative).

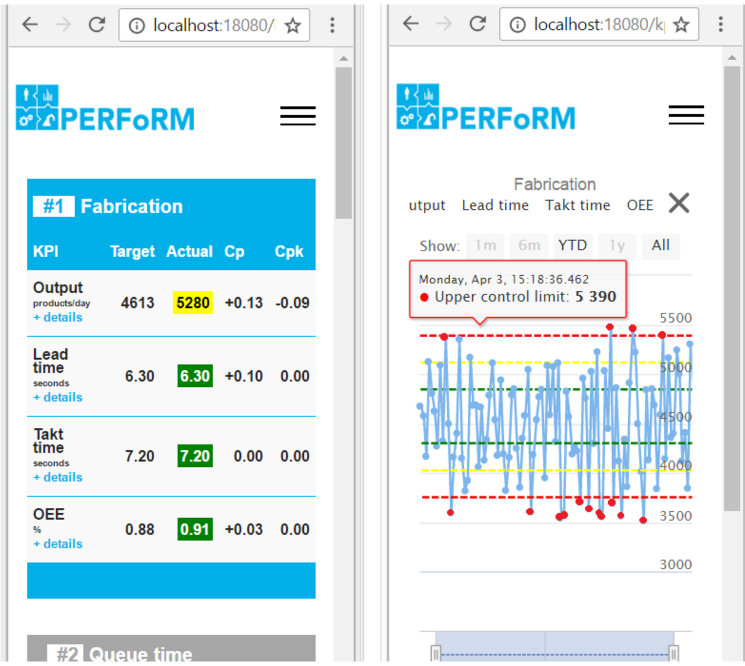


Figure 7: Two views of the tool in KPI monitoring mode

A colour scheme enriches the visual experience, enabling the user to quickly detect problematic situations. Additionally, a control chart for each KPI is accessible by opening a new UI perspective, displaying its timed evolution, as shown in **Figure 7** - right.

3.1.2. GKN case study

The GKN use case main objective is to implement a new industrial structure in the current production plant based on a Micro-Production Flow Cell that is able to reduce change over time and to realize different product and, therefore, also be able to give a flexible and reconfigurable aspect to the whole industrial plant (see Deliverable 10.1 for more details). To pursue the objectives defined for the use case, an agent-based reconfiguration tool, focused on the logical re-organization, and a scheduling tool are used in combination with the KPI tool who participates of the real-time cell KPI monitoring, as shown in **Figure 8**.

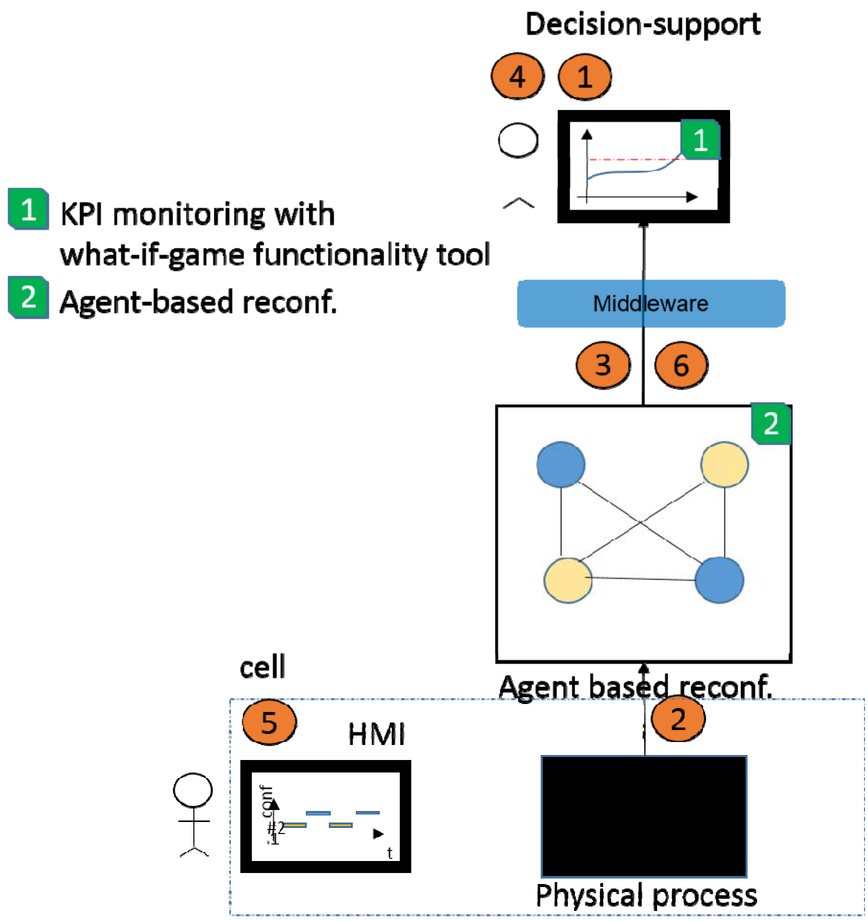


Figure 8: Real-time cell KPI monitoring

The set of actions and exchange of messages identified in the figure are:

- KPI monitoring tool is constantly monitoring KPIs;

- MAS are always collecting data from cell;
- Data is being sent to KPI monitoring tool;
- Human supervises the KPI evolution (Human-in-the-mesh);
- Human executes the reconfiguration scheduling (Human-in-the-loop);
- MAS informs about cell reconfiguration status (Human-in-the-mesh);

In addition to the relevant KPIs, such as the number of processed parts, other information related to the processes is displayed, like the process name, process position (left or right) and process status, e.g., sleep, active (idle, execution) or error.

3.2. Universal web based KPI visualization (TU BS)

Target definition: A web based platform for visualisation of decision support tools with respect to flexible and reconfigurable production systems

To provide a generic tool for visualisation and monitoring of project outcomes, a universal visualisation tool was developed. The web based KPI visualisation tool is designed to meet the requirements of the different solutions supporting the transformation to flexible production systems. To cope with the diverse set of requirements during the project duration, the visualisation is based on a modular assembly principle. All elements within the user interface (UI) are defined as entities. Entities can take several different forms and adapt to the needs of the user. An entity can, for instance, be a machine, a process, a whole factory, an analytic method, a simulation or a whole simulation environment (c.f. deliverable 4.1). A hierarchic structure can be easily implemented and has a dynamic behaviour. Hence dependencies between production facilities and reconfiguration of an agile production system can be presented. Changes in dependencies between production facilities and entities are dynamically updated.

3.2.1. Technical specifications

The universal web visualisation is mainly based on html and JavaScript. Data binding is done by KnockOut.js for an easy association of DOM elements with data from the MQTT (Message Queue Telemetry Transport) client (c.f. chapter 3.2.2). An automatic update of the UI takes place when the data or the data model state change. A binding to an UI element can be every valid JavaScript expression. By using the server sided JavaScript platform Node.js, elements like the MQTT protocol can be easily adapted and used. The first choice for implementation of dynamic and interactive visualisations is D3.js. D3.js is a versatile JavaScript library for visualisations in web browsers based on html5, SVG and CSS standards [4].

3.2.2. Communication with the PERFoRM Middleware

The communication to the PERFoRM Middleware is realised by a combination of a REST Interface and MQTT connection. The visualisation allows a container based presentation of production entities.

The definition of these entities is realised by a registration over the PERFoRM Middleware. The entity information is provided directly by the PERFoRM Middleware. After subscription and definition of the entity and the proposed indicating elements MQTT-Topics are defined and registered over a REST interface. A continuous data stream is then realised by MQTT with a data resolution of 1 Hz. **Figure 9** depicts the structural approach for the transfer of configuration and visualisation data.

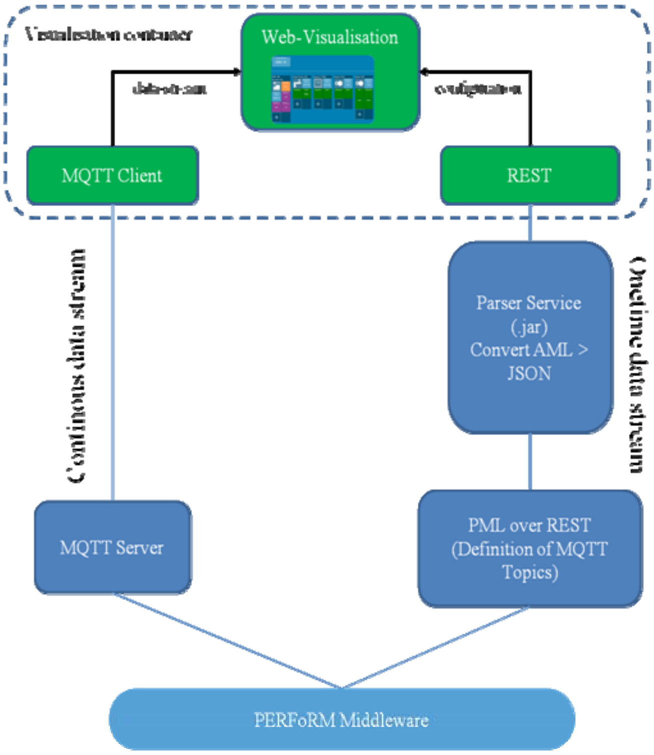


Figure 9: Communication to the visualisation container

3.2.3. User interface

The graphical user interface is shown in **Figure 10**. The left frame represents the navigation interface with a representation of the selected entity. Short information can be placed here. In the bottom area navigation buttons are placed. A shortcut to a default view of the selected entity, a home button and a back button is placed here. On the right side, all entities defined by the PERFoRM Middleware are placed here. The definition of all entities and sub-entities are stored in the PERFoRM-ML file. Tiles for process view and an evaluation of process properties are located in this view. These tiles are automatically generated from information saved in the P-ML file. Additional tiles for process evaluation can be dynamically added. Synchronously, the device description within the P-ML file will be updated. Dynamic and interactive graphical presentation is implemented by use of the D3.js JavaScript library.

Figure 10: Graphical user interface of the Universal Web-based Visualisation

Figure 11 depicts an example for the use of D3.js. in form of an evaluation view of a current signature. The signal is internally Fourier transformed and the result is shown as a heat map within the visualisation interface. In this case the amplitude is represented by the intensity of the heat map colour. The depiction is one example and shows the flexibility and adaptability of the applied visualisation solution.

Figure 11: Universal Web-based Visualisation: Exemplary view of a fast Fourier transformation on a power signature

Implementation of the Universal Web-based Visualisation

The proposed visualisation tool will be developed in the experimental lab of the IWF and will be applicable in the SmartFactory KL preindustrial use case and in the Siemens industrial use case. The visualisation views regarding **Figure 10** & **Figure 11** are screenshots from the application within the experimental lab of the IWF. The application at the “Industry 4.0 system” in the SmartFactory KL is under implementation. **Figure 12** contains the data point and entity definition at the SmartFactory demonstration case. The upper area contains the first level for visualisation. The first level contains the superordinated entities. An automatic reconfiguration is possible related to the real arrangement of the several production cubes. In the case of a change during the operation, a rearrangement within the visualisation takes place simultaneously.

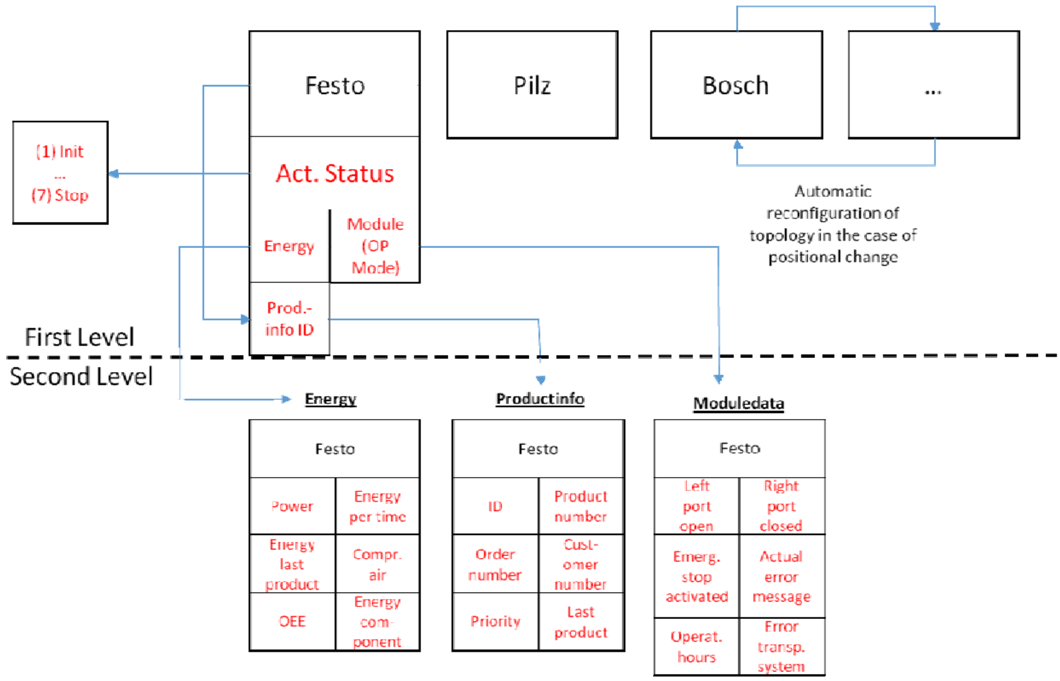


Figure 12: Draft for Industry 4.0 system SmartFactory visualisation containing entities, data points and functionalities

4. Decision rules and support

4.1. KPI monitoring with what-if game functionality (IPB)

Target definition: What-if game functionality for the determining optimized operational function points for tactical and strategical decision support

Used by many industrial sectors, the Six-Sigma is a set of techniques and tools for process improvement [5], targeting the variability reduction in key product quality indicators, which goal is the achievement of a very low level of defects. Assuming that the improvement of process performance and a reduced variability of the key parameters of the running manufacturing processes are critical features, the module is responsible for the statistical quality control of the temporal evolution of the KPIs, to ensure the expected quality of the products or product parts in production.

To perform the on-line statistical process control, a control chart for each KPI, relative to each of the stations, graphically displays the evolution of the KPI over time. The control chart plots the KPI's data points against some control lines. A centreline representing the average value of that KPI is displayed. Three other pairs of lines surround this centreline at one-sigma, two-sigma and three-sigma (i.e. at once, twice and three times the variance of the KPI). All these lines are used to assess some pattern denouncing out-of-control condition. Notably the Western Electric Rules [6] are used:

Rule 1: One data point falls outside the three-sigma control limits.

Rule 2: Two out of three consecutive data points fall beyond the two-sigma warning limits.

Rule 3: Four out of five consecutive data points fall beyond the one-sigma limit, on the same side of the centreline.

Rule 4: Eight consecutive data points fall on one side of the centre line.

Additionally, the following rules were also considered [5]:

Rule 5: Six data points in a row steadily increasing or decreasing.

Rule 6: Fifteen data points in a row fall within the one-sigma limits (stratification).

Rule 7: Fourteen data points in a row alternating up and down.

Rule 8: Eight points in a row on both sides of the centreline with none falling within the one-sigma limits (mixture).

Points that present a pattern abnormality are signalled (e.g., be painted red) for better visualization. Additionally, the points that falls between the two-sigma and the three-sigma limits are stand out by the use of appropriate interval limit colours. Revisiting **Figure 7**, it is possible to see the result of the practical implementation of Rule 1 (dots in red).

4.1.1. Application at WHP

The what-if-game mode, provided by this module, adds an extra functionality by introducing degrees of freedom (DoF) intervals on the KBFs, allowing to foresee how the system behave within the specified intervals. Therefore, new production scenarios can be elaborated considering a more sustained decision-making process, based on the obtained expected KPIs. This mode is initiated by the decision maker for possible scenarios assessment or e.g., in order to mitigate the deviation of the actual KPI values from target values.

The simultaneous interval variation of several KBFs increases the scenarios combinatorial search space. Therefore, and without compromising the tool responsiveness, an intelligent scenario generation is implemented, prioritizing the calculation of the most promising scenarios (as shown in **Figure 9**), in a similar way to that proposed in [7].

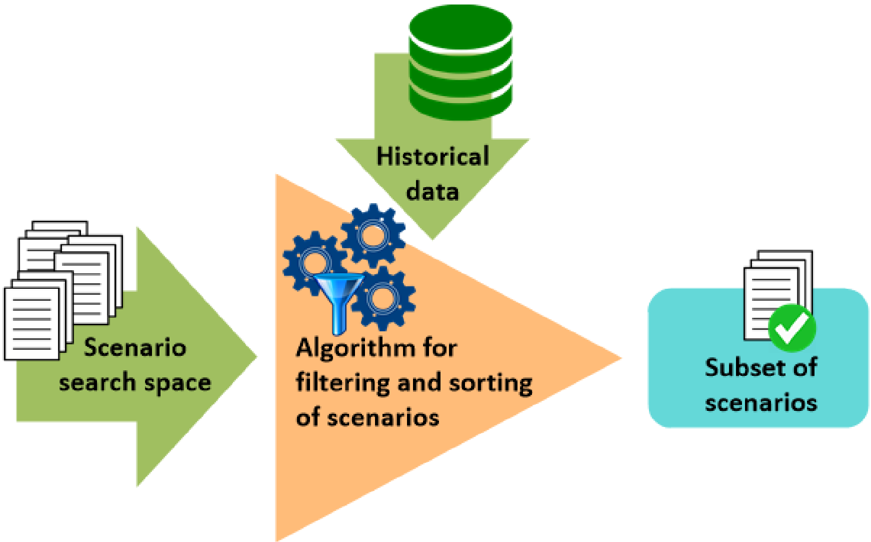


Figure 13: Scenario generation process

The What-if-game Module uses internal info and the KPI Calculation & Statistical Quality Control Module to generate relevant outputs that must be provided to the Visualization Module, allowing the presentation of data in a way that could be evaluated at a glance.

The what-if-game mode functionality presents data in a spider diagram, aggregating all the relevant KPIs into one graphical display, see **Figure 14**. The figure left side shows the overall system what-if KPIs results while, on the right part of the figure, the user can study other levels of granularity by searching KPIs at processing/station level.

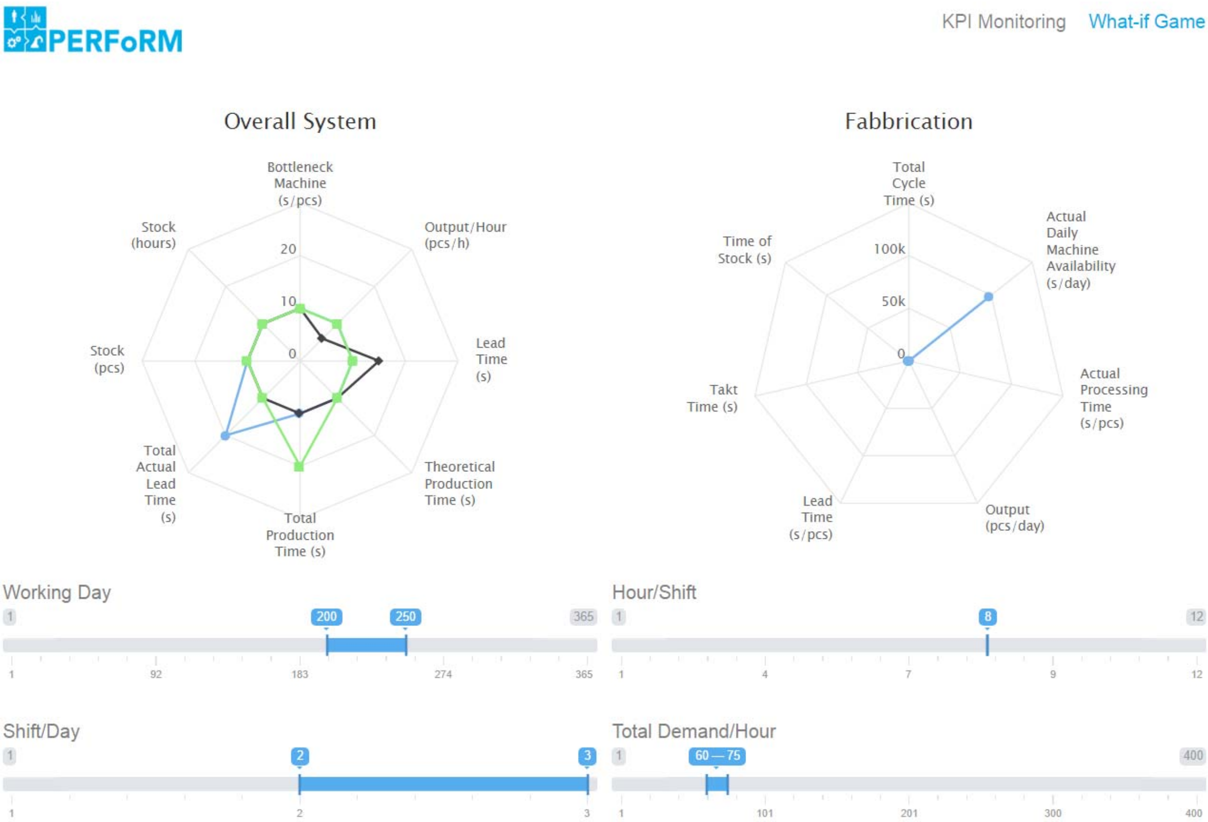


Figure 14: View of the tool in What-if Game mode

The decision maker, can change the desired KBFs by adjusting the sliders located at the lower part. With this, a new set of solutions will be displayed to the user allowing a thorough impact assessment of these adaptations.

4.2. Data-driven diagnostics for manufacturing equipment (L'boro)

Target definition: Fault-diagnostic based on data-driven approaches to support maintenance decisions in manufacturing applications.

This section deals with data-driven approaches for diagnostic application in manufacturing equipment. The basic idea is to identify the status of the machine and isolate the faulty element in failure case directly from data available about the machine (such as sensors, maintenance data, etc.), instead of the process model. Initially, a purely data-driven approach was introduced. Then due to performance limitation on the data-driven approaches, a hybrid approach that combine experts' knowledge with probabilistic machine learning approaches were introduced. Finally, a preliminary study and analysis of Siemens case-study was presented.

4.2.1. Failures statistical analysis

Statistical techniques were used to extract patterns and correlations in our observations and make inferences that help us understanding the nature of the failures and faulty situations. We look at statistical techniques and properties that are interpretable, such as decision trees, K-mean, etc., as opposed to neural networks, which is not easy to understand. Improvements in the performance and applicability of off-the-shelf statistical analysis techniques in a variety of domains have made their application to Diagnostics and Prognostics for industrial application a compelling option. The use of statistical techniques helps to understand structural behaviour of the failures and faults for different machines. Once we have built a general understanding of the fault behaviour, we can use them to detect faulty behaviours that are likely to indicate some failure in the high-level functionality being provided by the machine.

Complex machines are composed of elementary subsystems (different energy domain), such as hydraulic, electrical, mechanical, etc. In such complex systems, subsystems are coupled and fault in one subsystem might cause faults in other subsystems. Accordingly, it is essential to understand the structure of the complex system and the inter-dependency of the composed subsystems. In this work, we rely on the maintenance and operator's maintenance load data to understand the structure of the complex systems and find out the interdependence among the different subsystems [8].

In general, data mining is a set of methods for extracting hidden correlations and trends that are embedded within the dataset [9]. The use of data mining can be justified if the future behaviour of the machine will follow the same pattern in the past. It is computationally more efficient than ''learning from scratch'' and, perhaps more important, identifying explicitly the changes in the system that could provide further insights into the changes in the respective environment.

Operator's and maintenance team text comments on fault incidences requires text mining tool to extract relevant knowledge. However, text-based analysis depends on the quality of the comments (eg. spelling mistakes, grammatical mistakes). What is more, some comments are not clear and does not give any useful insight on the fault. In Spite of these limitations, text-mining provides us with a general understanding of the machine structure and fault statistics.

The aim of data-driven fault diagnosis is to identify the status of the machine and isolate the faulty element (subsystem) in case of fault or failure. In other words, the data-driven fault diagnosis can be performed into two steps, firstly; by extract the machine status, secondly; identify the faulty part in fault/failure case.

In manufacturing application, not all process parameters are accessible, due to the absences of sensors or since it is not observable parameters. Also, complex systems might have separate data-loggers and consequently data are not synchronized. Due to these limitations in the datasets, pure data-drive approach might fail to identify failures in complex machine. In order to overcome these limitations a hybrid approach that combine Bayesian networks with experts' knowledge will be introduced in the next section.

4.2.2. Bayesian network

The developed modular component-based modelling approach for diagnosis is based on the use of Bayesian Belief networks (also known as a Bayesian network). A Bayesian Network is a directed acyclic graph where nodes represent random variables and the directed arcs connecting pairs of nodes represent the probabilistic dependency relationships between the random variables. The node where the arc originates from is called the parent node and the one where the arc ends is called the child node. **Fehler! Verweisquelle konnte nicht gefunden werden.14** shows a simple Bayesian network structure.

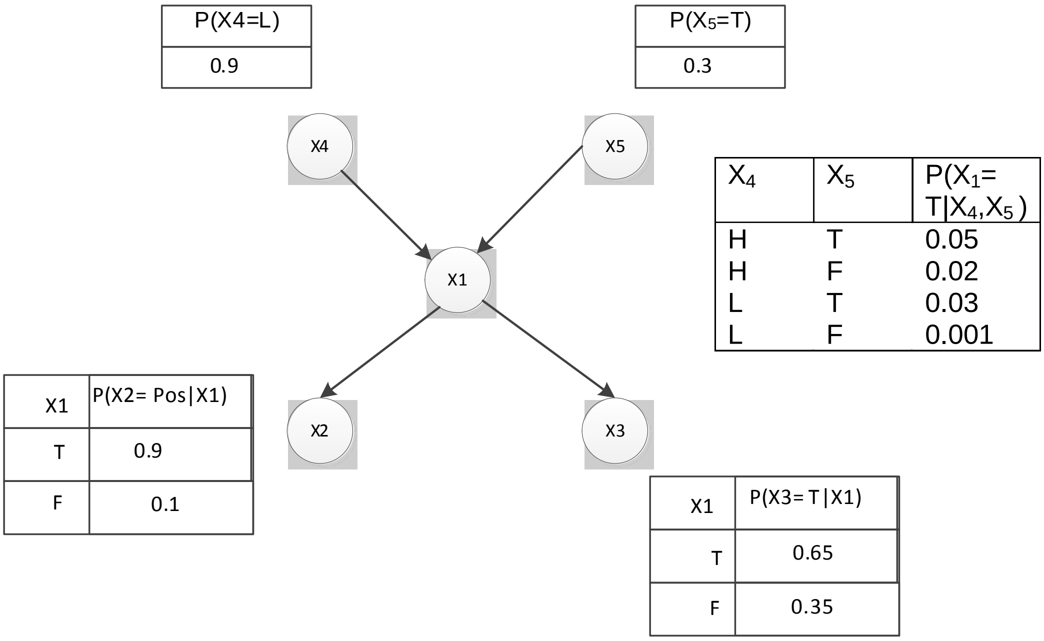


Figure 15: Example Bayesian network

Each node in the network has a conditional probability table (CPT) that quantifies the dependency relationships between the node and all of its parent nodes. A Bayesian network could be defined as a tuple <G, P, X> where G is a directed acyclic graph (DAG) whose nodes represent a set of random variables X = {X1, X2, …,Xn} and whose edges represent dependency relationships among the variables. In the Bayesian network of **Fehler! Verweisquelle konnte nicht gefunden werden.** the random variables X = {X1, X2, …,Xn} are represented by the nodes of the graph. The third component P = {Pr(X1|ΠX1),…, Pr(Xn|ΠXn) } is a set of conditional probability tables (CPT) that quantify the dependency relationships between every node in G and its parents nodes where for every node Xi ∈ $\boldsymbol{X}$ one CPT exists that defines a conditional probability distribution $P_r(X_i|\Pi_{X_i})$. The joint probability distribution over X can be given as:

$$P_r(\boldsymbol{X}) = P_r(X_1, \dots, X_n) = \prod_{i=1}^{n} P_r(X_i|\Pi_{X_i}) \qquad (1)$$

The diagnostic process in Bayesian Networks involves inferring the probability of the occurrence of an unobservable fault hypothesis based on the measured (observed) evidence (Symptoms). In other words inferring the probability of X_i being in a certain state, given the observed evidence E, which is expressed as $P_r(X_i = x|E)$ where $E = (y_1, \dots, y_m)$, y_j being the observed state of the variable Yj and $(Y_1, \dots, Y_m) \subset (X_1, \dots, X_n)$.

A very important and powerful characteristic of Bayesian Networks is their ability of Belief Updating via bidirectional propagation of new evidences throughout the network. This allows for the conditional probability of each node to be updated as new evidences or observations become available. In our case the diagnostic aim is to identify the root cause of an abnormal or faulty observed equipment behaviour. This should be a specific failure within a specific equipment module that explains the observed out-of-range characteristic. The status of equipment modules is thought to be hidden or unknown and the aim here is to estimate this status in light of other observed sensory variables within the system whose statuses are known or possible to estimate, including the observed fault itself. This makes the diagnostic modelling process a process of linking the observed variables to the hidden variables in the system in such a way that makes the probabilistic reasoning possible. Apart from the observed product characteristic other observable variables that should be used to assist in the diagnostic reasoning include those representing other observable parameters and observations in the system such as the various sensory data, operator observations and process settings.

The modelling approach that will be used in constructing the Bayesian models is Object-oriented Bayesian networks (OOBN). This approach enables component-based encapsulated modelling on the sub-system level, while enabling the individual sub-models to be integrated into a wider system-level model that can be used on the overall production system level.

The Bayesian networks are developed using HUGIN software environment following the modelling guidelines and procedures developed and devised during the FP7 SelSus project. Hugin reasoning engine provides an API for **Figure 16** shows the main required components to integrate the Bayesian models into the PERFoRM middleware. The sensory data, operators' observations and machine alarms are the input to the HUGIN engine. According to these parameters, the Bayesian model will isolate the faulty subsystem.

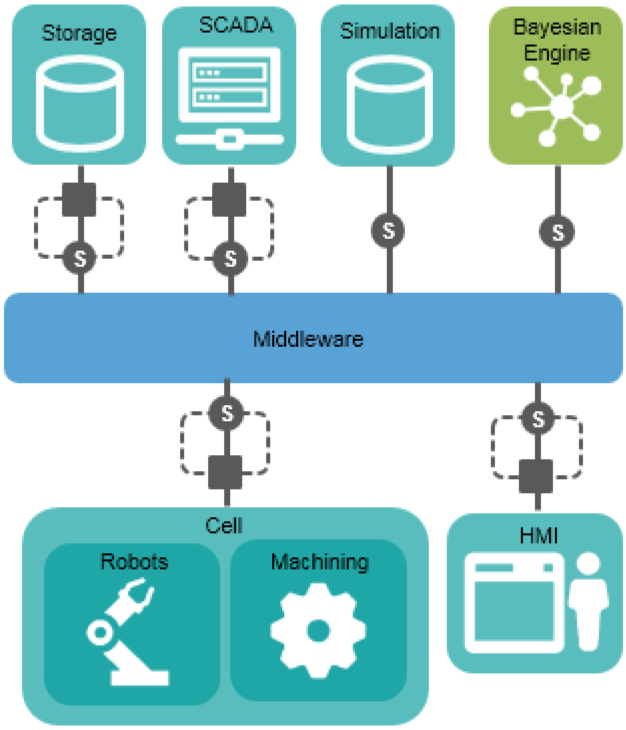


Figure 16: Bayesian models and engine integration with PERFoRM Middleware

The HUGIN software for probabilistic graphical models can handle Bayesian Networks (also known as Bayesian Belief Networks (BBNs)) and create a model, which can be used for probabilistic decisions. HUGIN software environment consists of two tools: the HUGIN Graphical User Interface (HGUI) and the HUGIN Decision Engine (HDE). The HGUI is an interactive user interface developed on top of the HDE used for developing and testing models. The HDE is the inference engine that can be integrated into other systems using its Application Programming Interfaces (APIs). The use of an API is necessary when integrating a HUGIN model into existing or new software like the control software of a machine or a maintenance decision support system as is the case in this task. APIs are available for C, C++, .NET, Java as well as a COM interface. In addition, the HDE has a Web Service API, which is the interface that will be used to deploy the HDE on the PERFoRM platform. Using the HUGIN Web Service API the state of the HDE is manipulated using primitive API calls. This usually requires that data is prepared and transformed prior to invoking HUGIN functionality.

The HUGIN Web Service API exposes the functionality of the regular HUGIN inference engine as a web service interface. It is a self-contained inference engine with a web server on top, providing a low level RESTful interface for invoking any HUGIN function. The common HUGIN objects Domains, Nodes, Tables, et cetera are exposed as resources each with a distinct URL. The server takes care of managing the life cycle of HUGIN objects, i.e., creating objects on request, invoking functions, updating state of objects and releasing objects again when not used. Any custom application can carry out inference computations by invoking HUGIN functions as plain HTTP requests. The technology stack is shown in **Figure 17**.

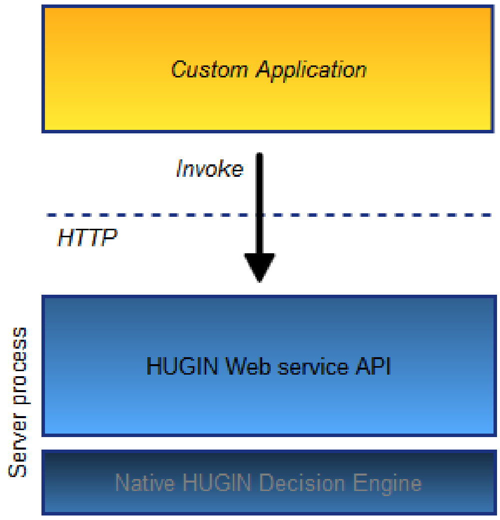


Figure 17: The HUGIN Web Service API technology stack

4.2.3. Siemens Case-Study

In Siemens case-study, maintenance data and alarms log datasets for three Carnaghi machines (AC16, AC32 and AC46) were provided. In this case-study, the machine learning techniques were used to find the correlations amongst different features in the given dataset with the failures and faulty subsystems in the three Carnaghi machines. The provided datasets were BDE data (plant data log) and LHNet failure reporting log. The BDE contains operational information (e.g. workpiece specifications), machine alarms logging data. While the LHNert contains the reported failures and faults were reported by the operator's/ maintenance team.

The three Carnaghi machines are almost identical. Each Carnaghi machine is composed of the following parts:

1. Tool-changer
2. Spindle
3. Chip removal system
4. The tool holder (Ram) system
5. Cross rail system
6. Hydraulic system

Alarms and maintenance data:

- BDE: Machine alarm messages
  - Number of machines: 3
    - Start date:20 Nov 2015
    - End date:7 Sep 2016

- LHNet: Operator's/maintenance team notes
  - Start date: 2 Jan 2012

- End date:14 Sep. 2016

From the LHNet (maintenance data), fault incidences can be grouped into two subcategory, namely, they are faults that cause alarms, and faults that cause a plant shutdown. **Figure 18**, indicates the overall failures in all machines.

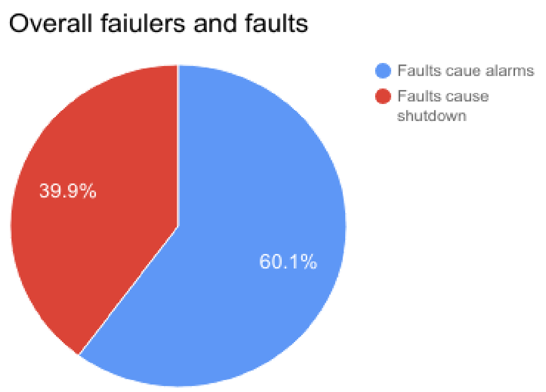
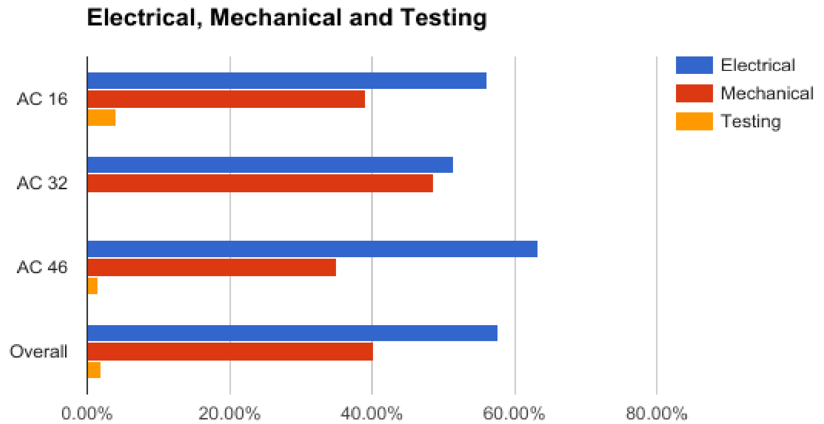


Figure 18: The overall faults and failures that triggers alarms or causes shutdown

In the maintenance data, operators classify the fault/failure incidents into three different subgroups as shown in **Figure 19**. In general, electrical faults seem to be dominant in all machines followed by mechanical faults. In the AC32 number of electrical and mechanical faults are almost the same. To sum up, based on the operator's comments 57.7 % of failure are electrical faults (**Figure 19** and **Figure 20**). Also, 40% of the faults case total failure for the machine. Amongst the 40% of shutdown failures, there are 60.2% electrical failures and 34.5% mechanical failures.



.

Figure 19: Types of failures and faults

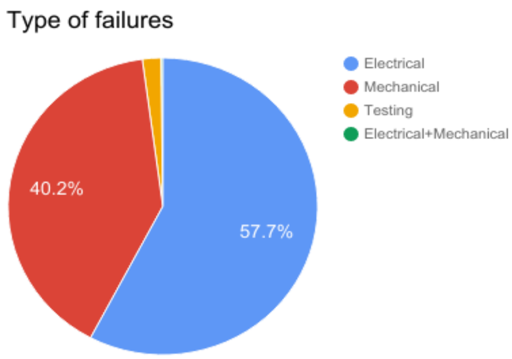


Figure 20: Types of failures and faults

According to the operator's /maintenance team comments, machine failures and faults can be categorised into eight different sub-groups using a set of keywords extracted from the maintenance data, the subgroups are:

1. Electrical system: actuators, junction-boxes, etc.
2. Pneumatic system: Clamping system, cooling, cleaning, etc.
3. Cooling system: coolant liquid, cooling fans, cooling pump, etc.
4. Hydraulic and lubricating system: lubricating/hydraulic oil, hydraulic pump, balancing cylinder
5. Spindle system
6. Tool-changer/Ram system
7. Movable-cross rail system
8. Conveyor Belt system

For each subsystem we have extracted key-words, then classify each failure/fault into a subcategory. **Figure 20** shows the overall failure (that caused shutdown) statistics. In general, tool changer failures over the given period are dominant. The second most occurrent failures were because of the cooling system, followed by conveyor belt and hydraulic system.

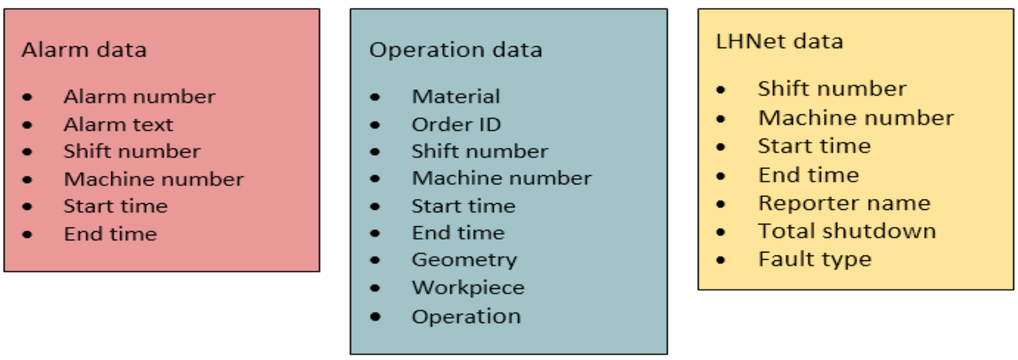


Figure 21: Important machine dataset

Figure 21 depicts the important dataset of the Siemens case study. Alarm data and operation data were labelled using the LHNet data, which were classified using the keywords extracted from the LHNet. The operation data is known in advance and it represents the production plan. This dataset, contains features related to the operation required, workpiece geometries, workpiece material and machine number. Based on these sub-groups operational and alarm datasets were labelled and aligned. The labelled dataset is strongly imbalanced data sets. Moreover, the available datasets are small in size. Due to this limitation, classes were weighted. Then, an RF models, Logistics classifiers Decision tree and Boosted Decision tree were used to predict the machine conditions. After that, the faulty subsystem isolated using a logistic regression.

Initially, labelled operation dataset was used to predict the faulty subsystem because it is known in advance and it allows us to predict a fault in the future. In this case, RF has the best performance with 83.2 % accuracy and 14.3% precision. This performance is very poor due to the weak correlation between operational data with respect to some failure and limited data size. **Table 4** shows the prediction of faulty elements for a given product in the operation dataset.

Class	Probability
No failure	0.2800
Conveyor-belt	0.0920
Pneumatic system	0.0898
Cooling system	0.0898
Lubricating system	0.0898
Tool changer	0.0898
Other failure	0.0898
Hydraulic system	0.0898
Control system	0.0898

Table 4: Failure probability at a given point in the operation dataset

Since operation plan is known in advance, machine status can be predicted in advance. To predict the status of the machine based on the operation dataset, two classifiers were used which are Random Forest and logistic classifier. However, the precision of fitted models were poor as shown in **Table 5**. This means that the correlation between operational data and machine status is weak. Then, the alarm dataset was labelled and used to train another set of classifiers model, in this way the precision were 10 times better than classifier trained only based on the operation dataset. It worth here to mention that classes in the dataset are highly imbalanced (machines are in the healthy state most of the time). **Table 5** illustrates the models fitted with different datasets and features and their corresponding evaluation. Fitted models were evaluated based on their accuracy (number of correct predictions/ number of data points), Precision (number of true positive/number of true positive +number of false positive) and the Area Under Recursive Operating Characteristic (AUR). RF model and logistics classifier trained using alarms and operation datasets have the best performance amongst other models fitted on different datasets.

ML Models	Dataset	Features	Evaluation (accuracy)	Evaluation (Precision)	Target
RF	Operation dataset	• Material • Order ID • Shift number • Machine number • Geometry • Workpiece • Operation	0.914	0.090	Fault: Yes/No
Logistic Classifier	Operation dataset	• Material • Order ID • Shift number	0.955	0.091	Fault: Yes/No

		• Machine number • Geometry • Workpiece • Operation			
Logistic Classifier	Alarm dataset	• Alarm number • Alarm text • Shift number • Machine number	0.990	0.990 *AUC=0.750	Fault: Yes/No
Logistic Classifier	Alarm+Operation dataset	• Material • Order ID • Shift number • Machine number • Alarm number • Alarm text • Shift number • Machine number	0.994	0.998 *AUC=0.990	Fault: Yes/No
RF	Alarm dataset	• Alarm number • Alarm text • Shift number • Machine number	0.904	0.99 *ACU=0.74	Fault: Yes/No
RF	Alarm+Operation dataset	• Material • Order ID • Shift number • Machine number • Alarm number • Alarm text • Shift number • Machine number	0.94	0.99 *ACU=0.94	Fault: Yes/No

Table 5: Machine status prediction using operation and alarm datasets

After predicting the machine status, faulty subsystem needs to be isolated (identified). For this purpose, RF and logistic classifier were fitted on the alarm dataset. **Table 6** depicts the performance for both models. Models shown in **Table 6** were selected automatically based on their performance in the validation dataset. In general, ensemble machine learning approaches seems to have better performance in isolating the faulty element in the machines.

ML Models	Dataset	Features	Evaluation (accuracy)	Evaluation (Precision)	Target
Boosted trees Classifier	Alarm dataset	• Alarm number • Alarm text • Shift number • Machine number	0.55	0.56 *AUC=0.86	Faulty sub system
Decision tree classifier	Alarm + Operation dataset	• Material • Order ID	0.98	0.98 *AUC=0.99	Faulty sub system

		• Shift number • Machine number • Alarm number • Alarm text • Shift number • Machine number			
RF	Alarm dataset	• Alarm number • Alarm text • Shift number • Machine number	0.42	0.45 *ACU=0.83	Faulty sub system
RF	Alarm+Operation dataset	• Material • Order ID • Shift number • Machine number • Alarm number • Alarm text • Shift number • Machine number	0.98	0.98 *ACU=0.999	Faulty sub system

Table 6: Identify faulty subsystem using alarm and operation data

Figure 22 depicts the overall structure of the proposed solution to predict faulty subsystem based on data-driven approaches. The rare alarms in the dataset were identified within two hours around the failure reporting time. Then, according to the percentage of the alarm occurrence around the failure with respect to its percentage of the appearance in the whole dataset, the alarms were linked to the failure incidents.

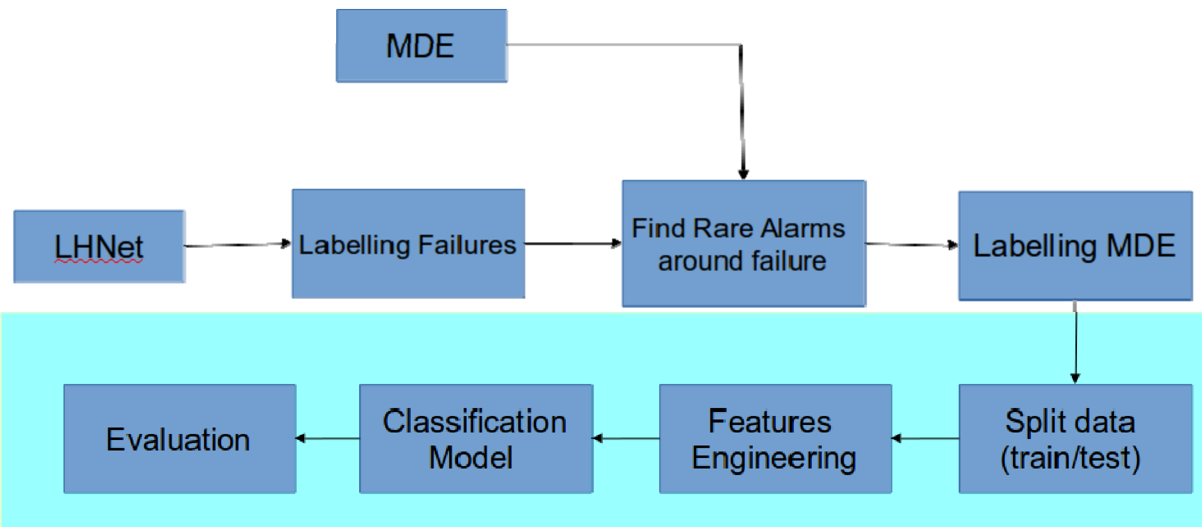


Figure 22: Data-Driven diagnostic system

To sum up, this section shows that it is possible to predict fault in the complex machine based on the operation plan that is available in advance. However, accuracy was not good due to the small available dataset in comparison with the number of variations such as the material used in the workpieces. The data-driven based models were improved by including the alarms from the machine. To improve fault diagnostic system human knowledge can be utilised with the Bayesian network. Firstly, human expertise must be structured and represented as shown in the next section

4.2.4. Failure Mode and Effects Analysis (FMEA)

FMEA is a systematic approach that determines potential failure modes in a complex system, e.g. production line in a factory, caused by either design or manufacturing process defects. Moreover, it identifies process significant characteristics that are required to prevent or detect failure mode. In general, FMEA is a tool used to understand the mechanics of failure in order to prevent pit from occurring.

The FMEA aims to prevent or reduce failures, starting with the highest-priority one's root cause. Also, FMEA assesses risks for mitigating known threat vulnerabilities. Accordingly, FMEA helps to improve the life-cycle consequences of the complex system.

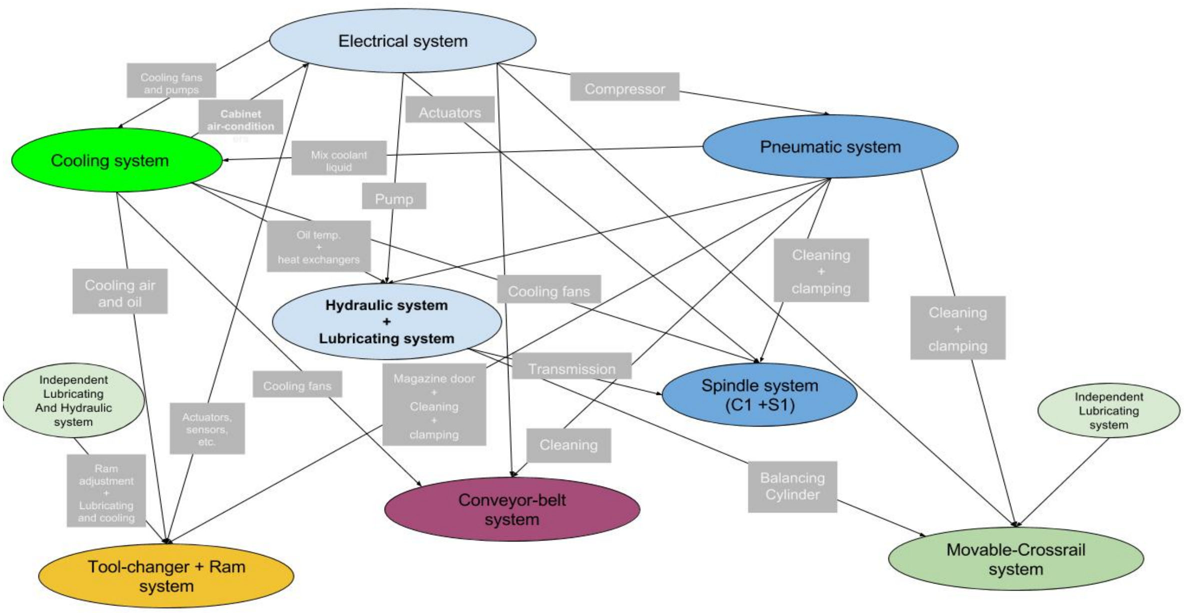


Figure 23: Machine sub-systems and thier interdependance relations

In the Siemens case study, the FMEA study was conducted by interviewing the operators and maintenance engineering, in order to identify the probable failures and their negative impacts on the overall process. In addition, the interdependence between different subsystems was extracted from the experts, as shown in **Figure 23**.

Based on the FMEA we noticed that the nature of the desired process (drilling and milling) and the workpiece properties (size, material) are directly affecting the failure modes. Moreover, crucial failures highlighted by experts rarely appeared in the maintenance data. In short, experts can identify and deal quickly with frequent problems.

The process characters extracted from the FMEA and the data-driven approaches are the core of the Bayesian models, which uses these characters as input while the outputs are the maintenance information with faulty systems probabilities.

Ultimately, the work introduced in this section is a preliminary investigation of the diagnostic approaches, which was successfully completed. The applicational side of this work will be continued in WP7.

4.3. Data mining (MTC)

Target definition:
Alarms and failure analysis for the purpose of maintenance decision support

Manufacturing industry is facing a myriad issues regarding the availability and reliability of their equipment. Equipment faults can cause long production line stoppages, high maintenance costs and low product quality. Well planned maintenance assists with keeping the equipment in a healthy condition, and helps to decrease the risk of large scale machine damages and revenue loss. There are mainly three categories of maintenance methodologies: reactive, preventive and predictive (condition-based) maintenance methods [10]. *Reactive maintenance* focuses on repairing an asset once the failure occurs. Unpredictability, costs associated with downtime, high cost of maintenance and risk of catastrophic failures are the main problems associated with the reactive maintenance approach. *Preventive maintenance* focuses on avoiding failures by performing maintenance tasks at predetermined intervals depending on the operating conditions. Over-maintenance, high cost of unnecessary change of spare parts, and expensive skilled personnel are the main issues associated with preventive maintenance. *Predictive maintenance* relies on condition monitoring techniques to predict the occurrence of a failure by monitoring relevant sensor data during the machine operation.

Condition based predictive maintenance is considered to be the best approach for improving equipment reliability, reducing production costs due to failure, reducing costs due to maintenance, optimising maintenance intervals, reducing the risks of catastrophic damage of the health of the machines and minimising unplanned downtime. However, in real production environment not all the failures and downtime reported are due to the condition of equipment. Many of the failures and downtime are caused by the production process itself rather than the condition of the machines. Hence, proper insight on the problem by analysing historical data is very important for selecting an appropriate maintenance task.

This section presents data mining and analytics techniques to explore the historical data related to failures and machine conditions (alarms) to get proper insight on the problem for guiding the future maintenance tasks. Data mining is defined as the exploration and analysis, by automatic or semiautomatic means, of large quantities of data stored in databases. Its main focus is the discovery of useful knowledge, including meaningful patterns and rules, from raw and apparently unrelated data [11].

4.3.1. Siemens case study

The production of industrial compressors and gas separators is being showcased within the Siemens use case. The Duisburg factory is responsible for the manufacturing of tailored compressors trains for oil and gas applications, such as air separation units or for the Liquefied Natural Gas (LNG) production. Currently the production of compressors is characterised by small lot sizes (1-30), machining, manual labour and a highly complex final assembly.

The objective of this use case is to improve the flexibility of manufacturing in particular focussing on the mechanical manufacturing of housing parts [12]. The intention is to use predictive and condition based maintenance, which shall improve the flexibility of manufacturing by better shifting production tasks between different machines and reduce quality issues and production failures. A failure or machine breakdown can lead to delays of the productions and missing parts to semi-finished products. The deployment of a predictive maintenance system involves the monitoring of the health of the equipment (three Carnaghi turning machines within the current context), the generation of related maintenance tasks and the combination of production tasks and maintenance tasks in the overall production schedule.

4.3.1.1. Siemens Architecture

The overall architecture of the Siemens use case can be seen in **Figure 24**. The architecture includes a ticketing terminal (LHnet) and the MDE/BDE data which includes the production data from the ERP system and machine data. It is envisaged that the machine conditions will be monitored by additional sensors which will automatically publish the data to the middleware (see **Figure 24**). Technology adaptors will be used to acquire live data from the SQL and Oracle databases. Additionally, standard interfaces (letter "S" as seen in **Figure 24**) will be used to enable plug-ability and interoperability. Further details on the technology adaptors and standard interfaces can be seen in D6.1: "*Self-Adaptive Machines Demonstrator Design and Set-up*" [13].

The *Data Analytics* tool should be able to access the data via the middleware, analyse it and visualise the result. The results generated by the *Data Analytics* tool will be used for the manual creation of new maintenance tasks within the *Maintenance Task Editor*. The *Scheduling* tool then accesses the maintenance tasks and proposes schedules for production and maintenance tasks. After evaluation of the schedules are done by the Simulation tool, the most appropriate maintenance task will be transferred to the *SAP* system. It is to be noted that the current work only presents the functionalities of the *Data Analytics* tool. Further work regarding the integration of the *Data Analytics* tool within the overall architecture will be conducted as a part of WP5, WP6 and WP7.

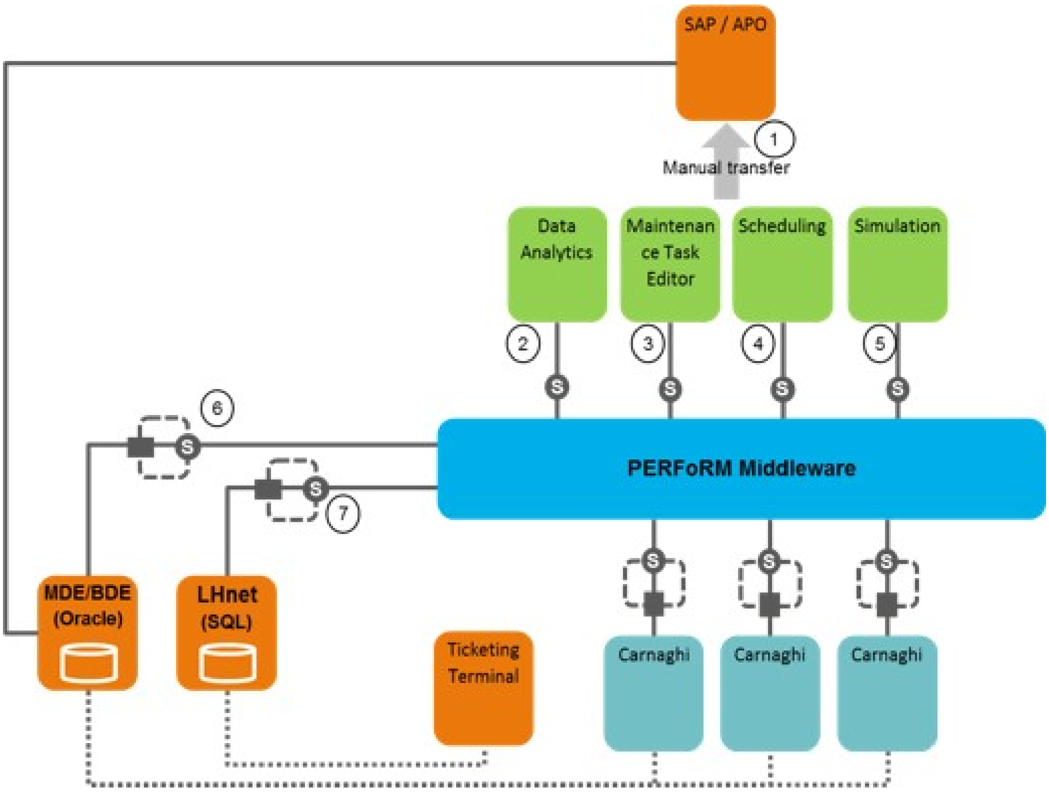


Figure 24. Siemens overall architecture

4.3.1.2. Proposed framework for data analysis and visualisation

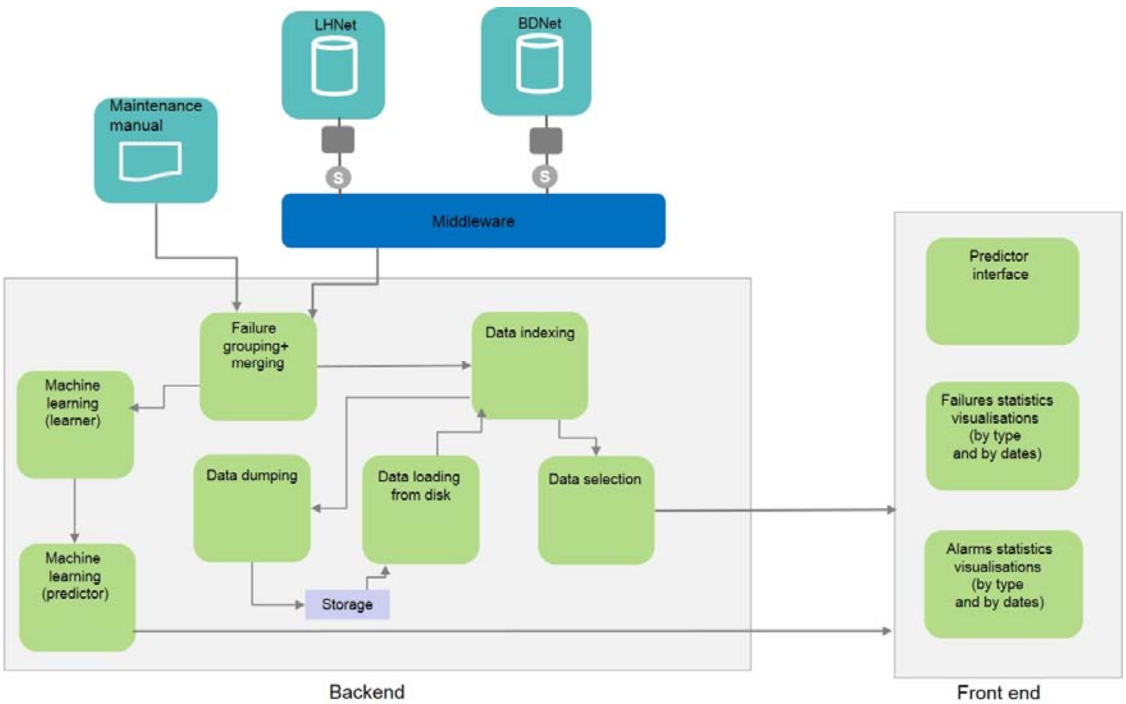


Figure 25: Integrated solution for visualisation and predictive maintenance

In order to analyse the data obtained from the Siemens Duisburg plant, a framework has been developed (see **Figure 25**). As seen in **Figure 25**, the data will be transferred from the LHNet and the BDNet database over the PERFoRM middleware. The proposed framework also utilises data from appropriate maintenance manuals. The proposed framework includes a *Backend System* for processing the data and a *Front end* for visualising the data. Further information regarding the various elements in **Figure 25** can be seen below:

- ***Backend system*:** The Backend system will mainly use the alarms data, maintenance data and the machine maintenance manual. The data will be collected from LHNet and BDNet databases via the middleware. The Maintenance database (LHnet) contains description of the problem reported in an unstructured textual form. This textual form is not the failure type despite it may give a hint about the failure type. Hence, there is a need of labelling to determine the machine failure from the textual form of the reported problem for a better analysis and data mining operations used for predicative maintenance. The Maintenance manual is being used for text mining for labelling the failure records.
- ***Data indexing:*** Once the data are grouped and merged, an indexer will process the data using the dates. The purpose of this step is to allow a faster query response for a range of dates. Consequently, the selection of the relevant data (depending on the visualisations) will be done in few memory accesses.
- ***Failure grouping/merging:*** One of the main challenge of the data provided is the categorisation of failures since they are described using unstructured text in the maintenance database. In order to match a particular trend in failures with its type, it is important to know the type of failure the system is dealing with. Thus giving the possibility of a more accurate prediction.
  A technique based on search engine technologies has been utilised in this work. The main question to answer is: given a particular failure description what are the probabilities that this failure belongs to the different categories? (a probability for each category, assuming that we will assign the failure type with the highest probability to the failure description message)
  (1) Understanding of failure categories is essential and this can be obtained from the maintenance manual. The maintenance manual consists of 16 different sections and each section is related to a particular part of the machine. These sections give the closest descriptions of type of failure in the available material (databases and manuals). Table 7 lists the type of failures derived from the manual.
  (2) A failure type is described by its label, e.g. "MASCHINENBETT" and the content of that section in the maintenance manual.
  A likelihood measure that assesses the resemblance between a failure textual description and the content of each section is established. The likelihood measure used in the current context is based on Lucene (which is a search engine and indexer technology).
  An analogy of this method is to consider the different sections of the manual as web pages to be retrieved by a search engine and the query used to retrieve these pages is the failure description. The first webpage retrieved is the failure's category. The corresponding algorithm can be seen below:

  Indexing for failure grouping:
  - Divide the manual into 16 parts. Each part represents a section.
  - The German analyser provided by the Lucene tool was used to reduce the derived words to their root form.
  - Create an index on the local file system ready to use in any search operation (no re-indexing will be required unless there is a change on the manual).

  Grouping:
  - Take a failure description (FD) from the database

  - Use a German analyser to get the root keywords.
  - Using the FD and the 16 chapters, return the section in the manual.
  - In case no section is retrieved return “Other”

- ***Data Dumping:*** A *dumping module* is used to store the indexed data on disk. The *loader* module will load the saved data from the disk into memory. The *Data selection* module will create a subset of the prerequisite data from memory depending on the requests from the frontend. The *machine learning* module generates an appropriate model for prediction. The model will be a rule based model where a pattern of alarms will define a type of failure to predict. The machine learning module will be an offline module and produce failure prediction model later used by the frontend to predict the failure for current alarms. The module will also produce key alarms for a group of failure.

- ***Front end:*** The frontend will be used for visualising the failure data and the alarms. The frontend will collect the pre-processed data from the backend and produce graphs for visualising the failures and key alarms.

- ***Class imbalance problem:*** A balanced dataset is very important for creating a good training set. Most existing classification methods tend to perform poorly on minority class examples when the dataset is extremely imbalanced. These methods aim to optimise the overall accuracy without considering the relative distribution of each class [14].
  The data used in the current case study was observed to be highly imbalanced, with very few records for each failure. Good sampling strategies are required to to overcome this problem. In the current context, an over-sampling technique proposed by [15] called SMOTE has been used wherein the minority class is over-sampled by creating “Synthetic” examples. This ensures that the data set represents all types of failures fairly.

- ***Machine learning for predictive model:*** After pre-processing (failure labelling and combining the alarms and failure data) of the data and balancing the data, machine learning techniques are applied to generate a predictive model to extract decision rules. The Decision tree classifier is one of the most widely used machine learning methods. Decision trees models are commonly used in data mining to explore and classify the data. The induced tree and its associated rules will be used to make predictions [16]. Details of the data mining and analytics approach are explained in the following section.

4.3.1.3. Analysis Process

The context of this deliverable involves offline processing of the data, i.e. the data from the LHNet and BDNet databases are stored in csv files and these files are used as inputs for generating decision rules. This technique can be used to generate decision rules which can be further used to develop a decision support system to predict failure using new alarms.

Details of the analysis process is illustrated by **Figure 26**. The inputs to the process are the csv files duplicating information from the LHNet and BDNet databases. The sub-processes involved are as follows:

1. Lucene [17] indexer/searcher: Failures are given a label using the failure text on the database and maintenance manual;
   a. The manual is divided into 16 groups
   b. Lucene is used to index the 16 groups using their contents;

   c. TF*IDF(Term Frequency*Invert Document Frequency) formula is used and a German analyser (for the failure text and maintenance manual written in German);
   d. Search metrics are used to determine the closest group to a failure description (also using a German analyser).
2. *Combine data and group by machine*: Data are combined from LHNet and DBNet. The alarms and failures are combined by date;
3. *Class Balancing*: Since the number of days where there were no occurrences of failure is more than the number of days were a failure was observed, a class balancing approach was used to balance the data before applying any predictive modelling technique for proper learning model;
4. *Classification*: After balancing the classes, a machine learning based classification method is used to generate a set of decision rules and determine the key alarms for each failure types (further details on the decision rules can be found in Section 4.3.1.4).

The output of the analysis process are as below:

- Decision rules in terms of alarm patterns that can be used for root cause analysis and also to develop a decision support system for early prediction of the failures;
- Identification of Key Alarms, i.e. the alarm subsets for each type of failure which can be used for monitoring the machine condition;
- Visualisation of the failures and alarms which will provide insight on the past events and help to generate appropriate maintenance tasks.

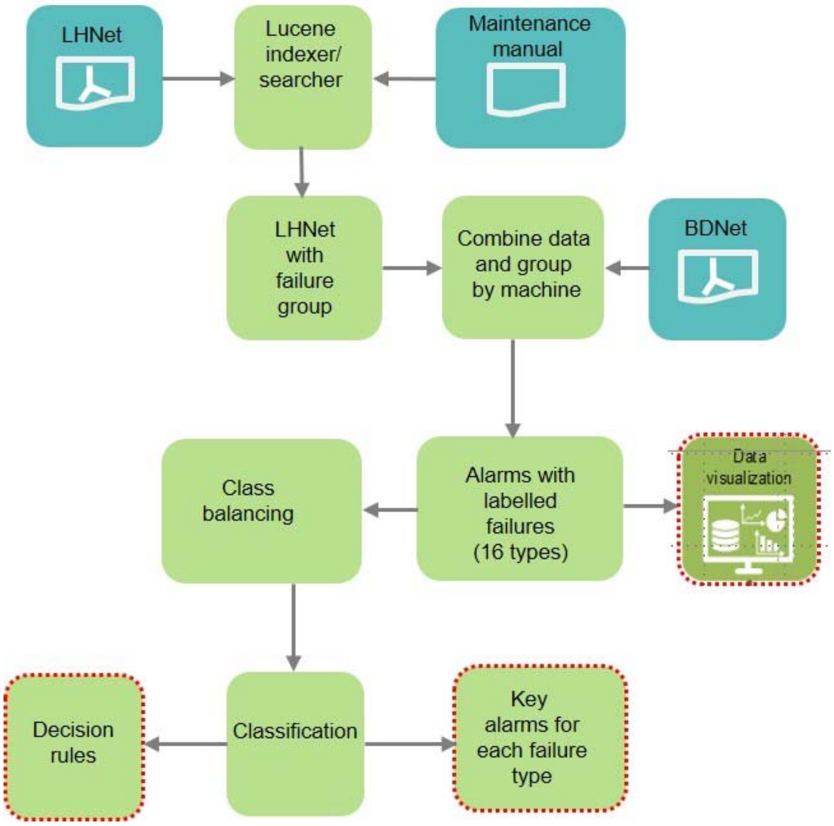


Figure 26: Data mining and analytics approach

4.3.1.4. Results

The machine maintenance manual was divided into 17 groups which was later used to group the failures and label them. **Table 7** below shows the lists of 17 failures.

No	Failure Types in German	Failure Types in English
1	DREHZENTRUM	ROTARY CENTER
2	MASCHINENBETT	MACHINE BED
3	TISCH	TABLE
4	STÄNDER	STANDS
5	VERFAHRBARER QUERBALKEN	PROCESSABLE CROSSBARS
6	SCHLITTEN	SLIMS
7	TELLERMAGAZIN	TELLER MAAGAZINE
8	KETTENMAGAZIN	CHAIN MAAGAZIN
9	ZUBEHÖRTEILE UND WERZEUGHALTER	ACCESSORIES AND TOOL HOLDER
10	HYDRAULIK- UND SCHMIERMITTELKREISLÄUFE	HYDRAULIC AND LUBRICATING CIRCUITS
11	KÜHLANLAGE	COOLING SYSTEM
12	PNEUMATISCHE ANLAGE	PNEUMATIC INSTALLATION
13	SPÄNEFÖRDERER	SPRING CONVEYORS
14	SCHUTZKASTEN	PROTECTION BOXES

15	MOTORGESTEUERTE TASTATUR	ENGINE CONTROLLED KEYBOARD
16	ELEKTRISCHE AUSRÜSTUNG	ELECTRICAL EQUIPMENT
17	ANDERE	OTHER

Table 7: Failure types

4.3.1.5. Outcomes of the Decision tree

Decision tree technique was applied on the data grouped by different failure categories and also on the complete dataset (without grouping them by types of failures). **Fehler! Verweisquelle konnte nicht gefunden werden.** illustrates some rules extracted via the decision tree technique on the complete set of data without grouping the data in different failure types. Whilst **Fehler! Verweisquelle konnte nicht gefunden werden.** and **Fehler! Verweisquelle konnte nicht gefunden werden.** illustrates some rules extracted via the same technique on the complete set of data after grouping the data by failure type.

Rules / Alarm patterns
IF Alarm-10860 = No AND Alarm-20050 = No AND Alarm-700333 = No AND Alarm-510308 = Yes AND Alarm-700540 = No THEN failure = Yes
IF Alarm-10860 = No AND Alarm-20050 = No AND Alarm-700333 = No AND Alarm-700543 = No AND Alarm-10208 = Yes AND Alarm-67051 = No AND Alarm-700754 = No: THEN failure = Yes
IF Alarm-10860 = No AND Alarm-20050 = No AND Alarm-700333 = No AND Alarm-700543 = No AND Alarm-601012 = Yes AND Alarm-700636 = No: THEN failure = Yes
Alarm-10860 = No AND Alarm-20050 = No AND Alarm-700333 = No AND Alarm-700543 = No AND Alarm-700310 = Yes AND Alarm-700733 = Yes AND Alarm-16913 = Yes AND Alarm-67051 = No: THEN failure = Yes
IF Alarm-10860 = No AND Alarm-20050 = No AND Alarm-700333 = No AND Alarm-10621 = No AND Alarm-700310 = No AND Alarm-4075 = No AND Alarm-510216 = No AND Alarm-700239 = No AND Alarm-600609 = Yes AND Alarm-700635 = Yes AND Alarm-600908 = Yes AND Alarm-600410 = Yes AND Alarm-16906 = Yes AND Alarm-67050 = No: THEN failure = Yes
IF Alarm-10860 = No AND Alarm-20050 = No AND Alarm-700333 = No AND Alarm-10621 = No AND Alarm-700310 = Yes: THEN failure = Yes
IF Alarm-10860 = No AND Alarm-20050 = No AND Alarm-510216 = No AND Alarm-700239 = No AND Alarm-600609 = Yes AND Alarm-700635 = Yes AND Alarm-600908 = Yes AND Alarm-600410 = Yes AND Alarm-700646 = Yes AND Alarm-510309 = Yes AND Alarm-10208 = Yes AND Alarm-6406 = No AND Alarm-700732 = Yes AND Alarm-601012 = Yes: THEN failure = Yes

Table 8: Alarms pattern (rules) extracted from the decision tree model created using full ungrouped data

This technique also derives key alarms as seen below:

- 10208, 16906, 16913
- 20050
- 510225, 510308, 510309
- 600410, 600609, 600908, 600913, 601012, 601213, 60113, 67050, 67051
- 700310, 700540, 700635, 700636, 700646, 700661, 700732, 700733, 700754, 70032

Decision tree was further applied on the data grouped by different failure categories and the alarm patterns for the failure “HYDRAULIK- UND SCHMIERMITTELKREISLÄUFE” for predicting failure occurring on the same day and two days in advance are presented in the **Fehler! Verweisquelle konnte nicht gefunden werden.** and **Fehler! Verweisquelle konnte nicht gefunden werden.**.

Rules / Alarm patterns
IF 700142 = yes AND 510008 = no AND 700754 = yes AND 700147 = no THEN there is a failure

Table 9: Alarms pattern (rules) extracted from the decision tree model for "HYDRAULIK- UND SCHMIERMITTELKREISLÄUFE" failure for same day

Rules / Alarm patterns
IF 700040 = no AND 510315 = yes AND 14014 = no AND 700041 = no AND 700010 = no AND 700038 = no AND 510010 = yes AND 16912 = no AND 700533 = no AND 10299 = no AND 510229 = no AND 21612 = no AND 600612 = no AND 700161 = no AND 700307 = yes AND 600911 = no AND 700037 = yes: THEN there is a failure
IF 700040 = no AND 510315 = yes AND 700010 = no AND 510010 = yes AND 16912 = no AND 700159 = yes AND 600612 = no AND 700161 = no AND 17212 = no AND 16913 = no: THEN there is a failure

Table 10: Alarms pattern (rules) extracted from the decision tree model for "HYDRAULIK- UND SCHMIERMITTELKREISLÄUFE" failure two days before the failure

Key alarms for "HYDRAULIK- UND SCHMIERMITTELKREISLÄUFE" failure are 510315,510010, 700307,700037,700159, 700142, 700754.

The decision rules obtained above could be used for predicting failures based on alarms. Further work will be conducted to validate the accuracy and applicability of the rules in collaboration within domain experts at the Siemens facilities.

4.3.1.6. Visualisation

This section shows the results of the application of different visualisation techniques to facilitate data analysis and to provide domain related insight into the data. Different types of visualisation have been used to analyse the data: (1) pie chart to illustrate the percentage of each failure category for a certain period of time, (2) trend charts for showing how the value of one or more group of failures changes over time and (3) heat maps highlighting the density of data e.g. density of failures within a given period of time. Due to data confidentiality only some of them are presented below.

Figure 27 shows the graphical representation of the failure statistics of all the machines for the period between January 2011 and February 2017. The results show that majority of the failures observed during this period belong to the failure group "HYDRAULIC AND LUBRICATING CIRCUITS".

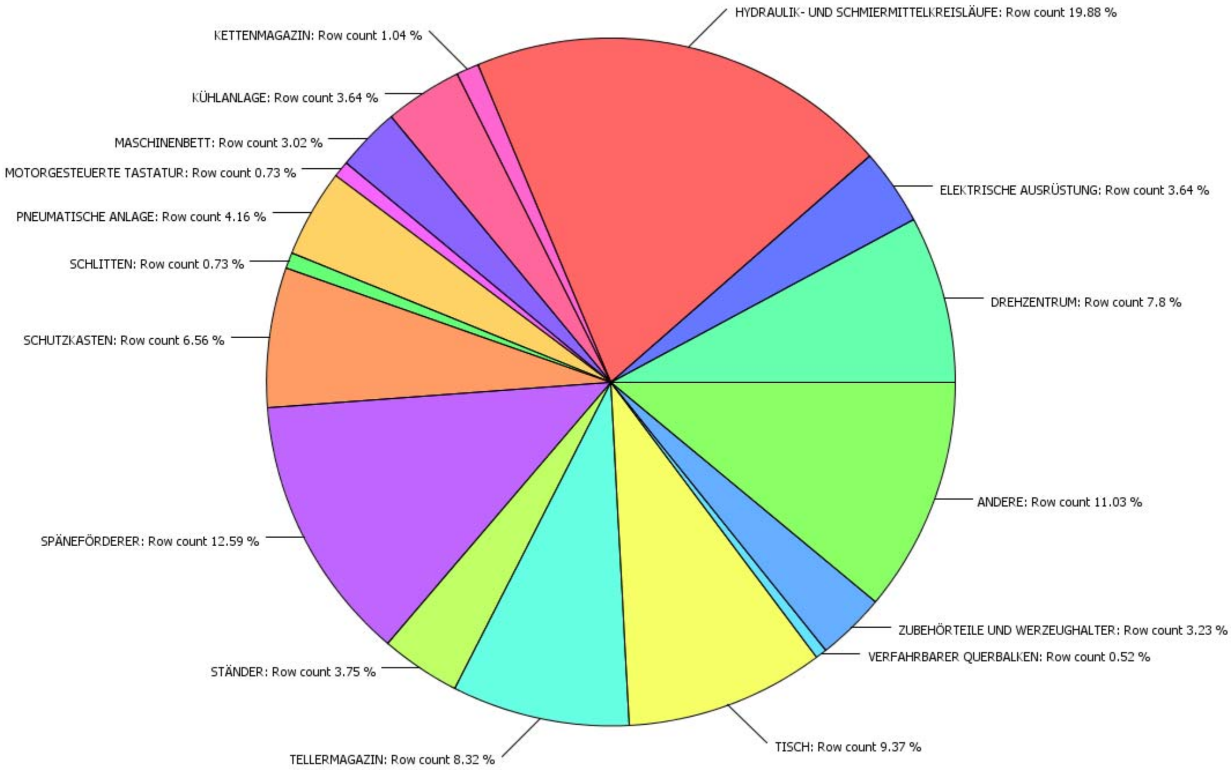


Figure 27: Percentage of type of failures for all machines for the period of 08/01/2011 to 28/02/2017

Figure 28 shows the graphical representation of the failure statistics of machine particular machine (Carnaghi AC16 in the current context) for the period between January 2012 and February 2017. The results indicate that the majority of the failures occurred during this period belong to the "HYDRAULIC AND LUBRICATING CIRCUITS" group.

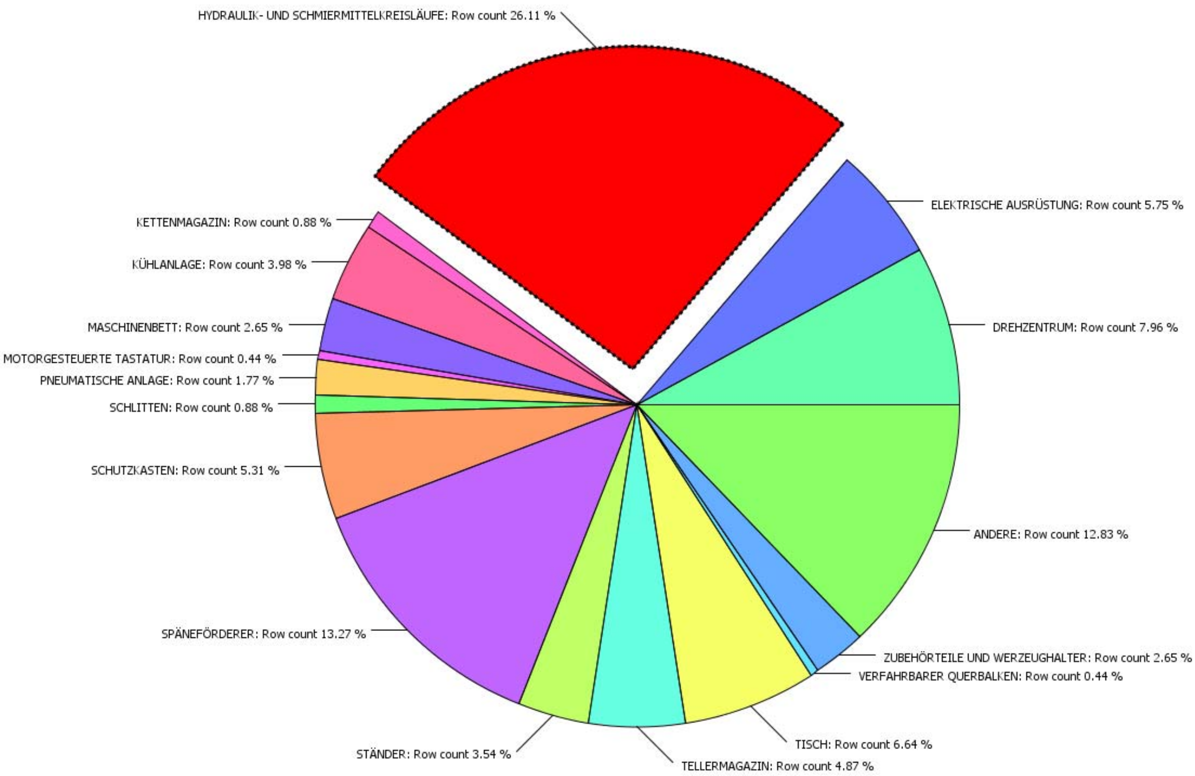


Figure 28: Percentage of type of failures - Carnaghi AC16

Figure 29 shows the graphical representation of the failure statistics of the machine Carnaghi AC32. The results show that the majority of the failure observed during this period belong to the "HYDRAULIC AND LUBRICATING CIRCUITS" group and the second most failures were reported "SPÄNEFÖRDERER" group.

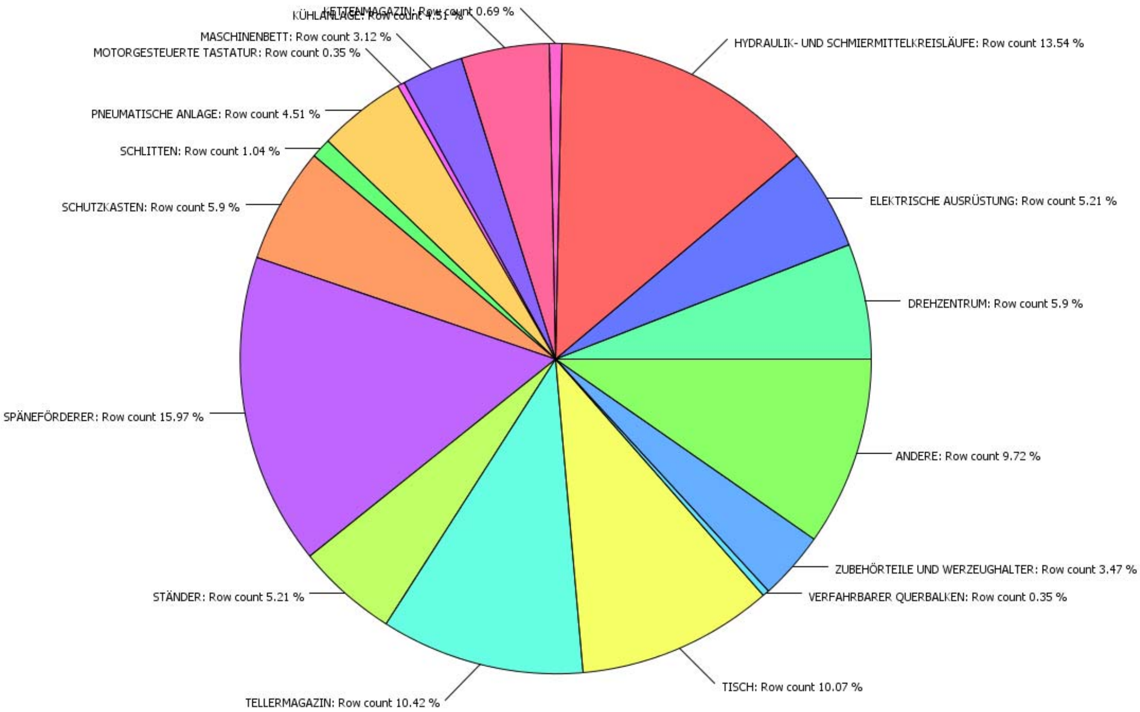


Figure 29: Percentage of type of failures - Carnaghi AC32

Figure 30 shows the graphical representation of the failure statistics of the machine Carnaghi AC46. The results shows that majority of the failure happened during this period are in the area of "HYDRAULIC AND LUBRICATING CIRCUITS" and the second most failures were reported in the area of "SPÄNEFÖRDERER".

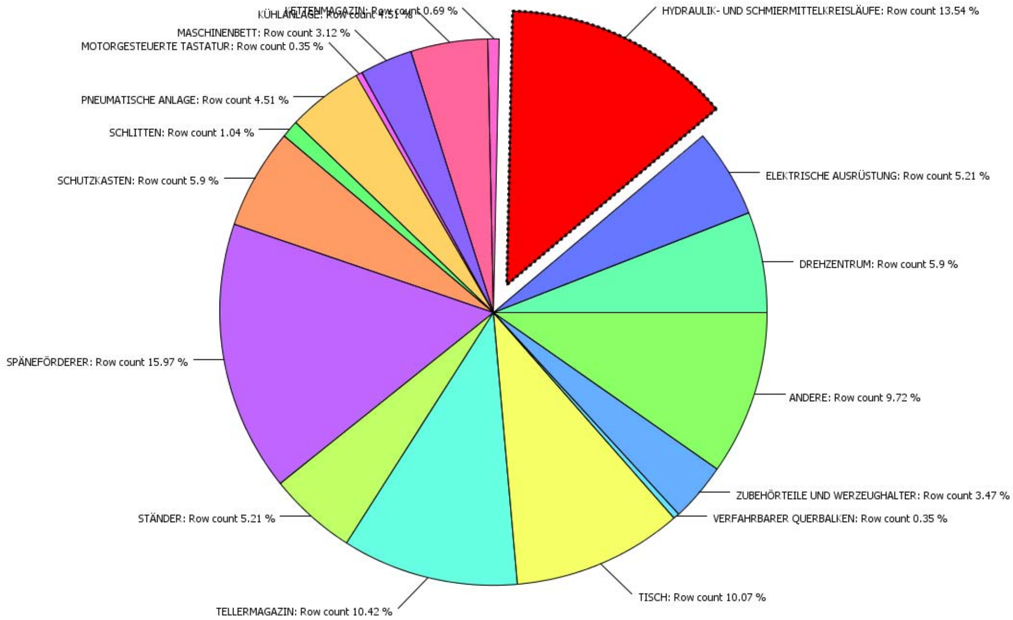


Figure 30: Percentage of type of failures - Carnaghi AC46

Alarm monitoring: As part of the visualisation, a tool has been developed to visualise alarms for a given machine and failures for a selected period of time. An illustration can be seen in below.

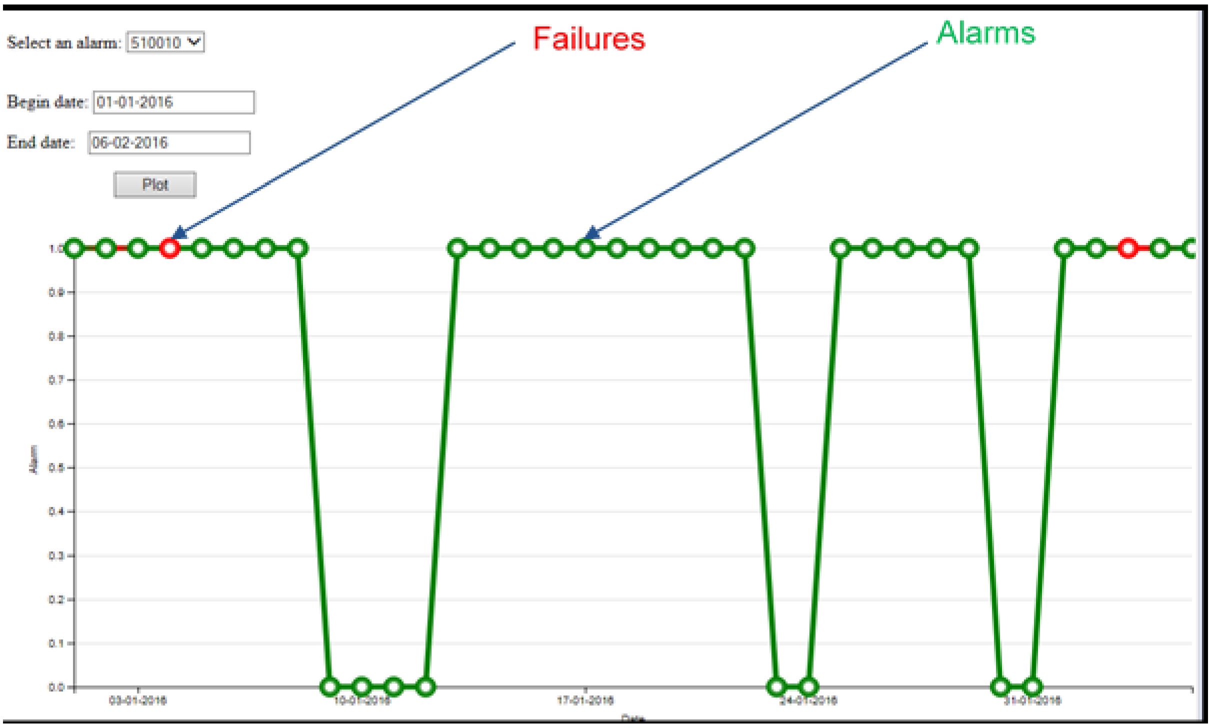


Figure 31: Alarm and failure monitoring

The figure above tells that not all alarms result in the failure of machine. This graph combined with the heat maps and trend charts could be used to monitor particular machines and/or particular alarms.

4.3.2. Further work

Currently the input data is being stored in a csv file and there is no interaction with the LHNet and BDNet databases via a middleware. With the implementation of the presented methodological approach into the Siemens use case an automated analysis will be enabled. The integrated solution for the Siemens use case can be seen in **Figure 24**.

The results from the failure labelling approach and the decision rules need to be validated as a part of future work. This validation may lead to modification within the model.

The overall aim is to enable the results generated by the *Data Analytics* tool to create new maintenance tasks within the *Maintenance Task Editor*. The *Scheduling* tool should then access the maintenance tasks and propose schedules for production and maintenance tasks. After evaluation of the schedules are done by the *Simulation* tool, the most appropriate maintenance task will be transferred to the *SAP* system. It is to be noted that the integration of the Data Analytics tool within the PERFoRM architecture will be conducted as a part of WP5 and WP7.

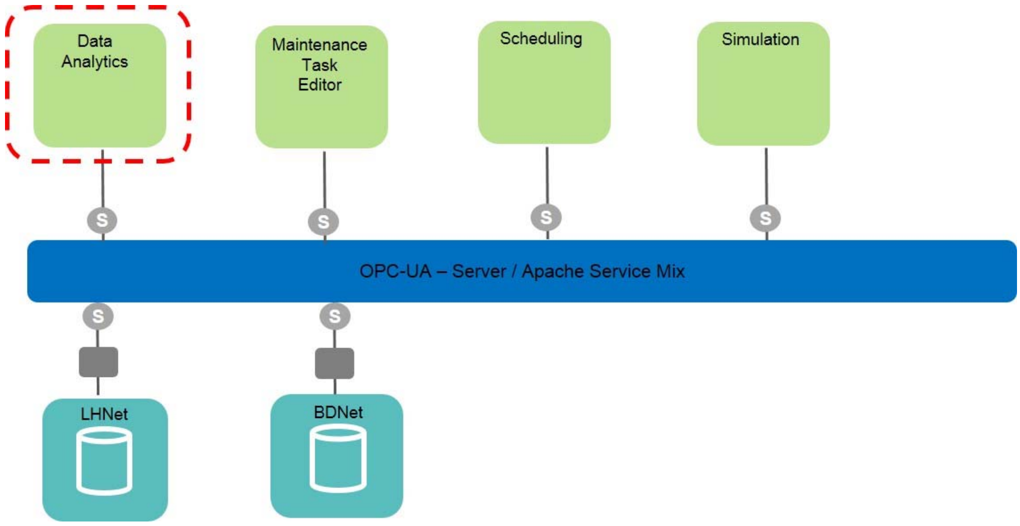


Figure 32: Integrated solution for Siemens use case

4.4. Min-Max Data Mining Toolbox (TU BS)

Target definition:
Component based condition monitoring by the use of high frequency current signatures

4.4.1. General approach

Production facilities are complex and highly automated combinations between mechanical and electrical components. A breakdown results in significant economic losses, material losses, environmental damage and in the worst case to bodily harm. Operational reliability and maintenance of production facilities and components is an obligatory necessity. Predictive maintenance leads to an early detection of expected facility failures. Potential hidden failures and their progression can be detected and corrected before a machine downtime occurs.

The topic and the objective of this research work is to develop a condition based predictive maintenance system in accordance with the overall PERFoRM line of thinking to detect abnormal behaviour of production equipment. The proposed approach should detect defects and unexpected behaviour down to component level (e.g. electrical drives, pumps or mechanical parts). All components of e.g. a machine tool are connected indirectly or directly to an electricity supply. The combination of measuring the central electricity connection of a machine with a high data acquisition rate and perform an analysis of the resulting power signature can give important evidence to the actual component conditions. The advantage of this procedure is the simplicity in data acquisition. Most condition based predictive maintenances approaches are based on more complicated and more expensive data acquisition techniques (e.g. vibration monitoring). According to **Figure 33** a virtual image of the real physical condition is used, to gain the most value information for maintenance decisions. On the physical side, the power signature of whole plants and information of maintenance logs and process failure information is used. On the cyber side, features from the power signature are extracted and used for the assessment of component conditions. System characteristics readable from the current signal are saved and evaluated against characteristics that have been saved before. As a result the combination in use of a low effort data acquisition system and the complete assessment in the cyber layer of the received data lead to the Min-Max Data Mining Toolbox system.

The proposed solution should be applied at the Siemens use case within the compressor plant in Duisburg. It is for this reason that metal processing, particularly CNC machining processes, are in the focus of interest. Selected for the PERFoRM project within the Siemens plant are three different CNC vertical lathes. The first focus is on the Carnaghi AC32. Preliminary work will be conducted at a CNC machining center as well as at the testing ground at the TU Braunschweig. All important components of a CNC machining center are in the focus of consideration:

- Induction motors
- Servo motors
- Ball screws
- Pump motors
- Hydraulic pumps
- Lubrication pumps
- Cooling lubricant pumps
- Frequency converters

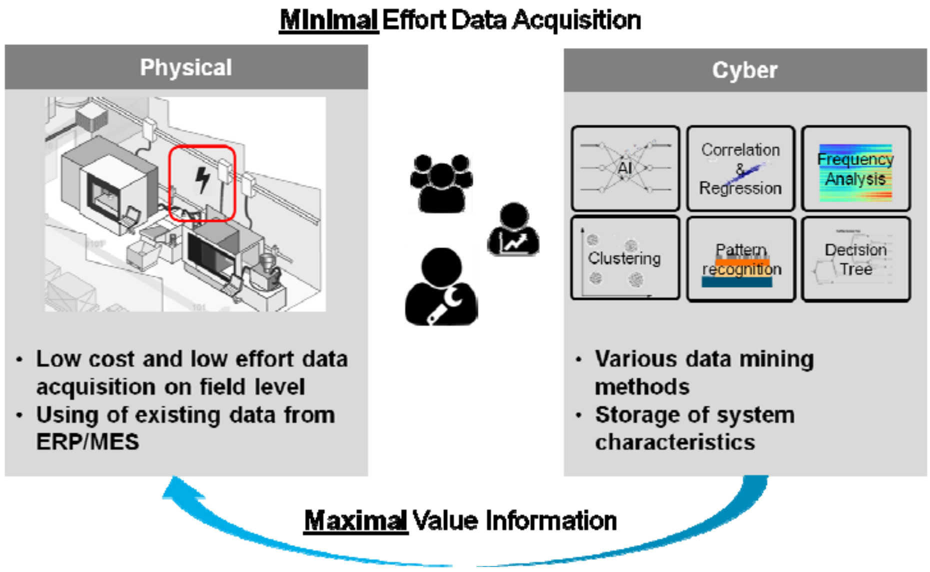


Figure 33: General apporach of the MinMax Data Mining Toolbox with the physical layer (left) and cyber layer (right)

4.4.2. Project approach

In addition to the description of the general approach, the general proceeding in the context of the PERFoRM project is explained here. **Figure 34** depicts the embedding of the Min-Max Data Mining Toolbox beside the web-based visualisation within the PERFoRM project. To achieve the possibility for live prediction of the Min-Max Data Mining Toolbox necessary training data has to be recorded. Recording of training data is necessary to distinguish between different operational sequences during the predetermined test run (c.f. chapter 4.4.6) and to gain knowledge about the conditions of considered components. In this stage additional current meters on selected components need to be installed to distinguish between operation of controllable and non-controllable components. Once the necessary information is recorded, the number of samples are sufficient for a reliable classification of the stored signatures that are used for the live prediction mode.

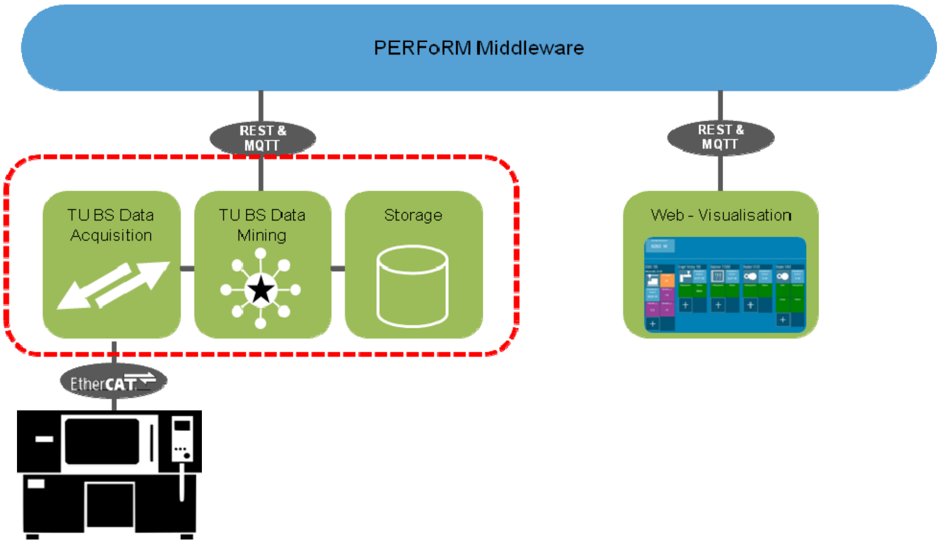


Figure 34: General framework for the Min-Max Data Mining Toolbox and the web-visualisation in context of the PERFoRM project

4.4.3. Developed Hardware

To work with measured data from machine tools a metering system was developed within WP4 and implemented at the Siemens use case within WP7. The metering system enables the acquisition of the data needed for later analysis processes. Data acquisition and analysis are performed on the same hardware platform. **Table 11** contains the technical specifications of the developed system. To allow the installation within the electrical cabinet of a machine tool, the dimensions of the metering system are restricted. A single board computer allows the application of edge computing. Most calculations and arithmetic operations can be directly performed in the metering box. The single board computer runs with a Linux Debian operation system with real time capability (RT Kernel). Communication is enabled by a remote shell interface via the ethernet communication port. A graphical desktop interface is not provided due to performance issues. Instead a graphical interface provided by a reduced web-based visualisation running on an Apache2 web server directly on the single board computer is used to monitor measurement data and perform data bank queries. System internal and external data distribution and real-time application for detecting the current condition are executed in the programming language Node.js. The metering system relies on an industrial metering and bus system from Beckhoff Automation.

Processing unit: ODROID XU4	
CPU	SAMSUNG Exynos 5422, Cortex-A15 & Cortex-A7 big.LITTLE, Octa core
RAM	2 GB LPDDR3 RAM
Relevant Interfaces	1 x Gigabit-LAN (EtherCAT communication)
	1 x USB to Ethernet-LAN (Network communication e.g. PERFoRM Middleware
Storage	64 GB eMMC for OS
	1 TB external Harddrive for Data storage
Software	
OS	Debian with RT Kernel
Data distribution and processing	NodeJS
Bus Master	EtherCAT Master
A/D-Converter	
Beckhoff EL3403	machines main electricity supply
Beckhoff EL3068	component monitoring (training box only)
Bus system	
Beckhoff EK1100 (EtherCAT Coupler)	
Current Transformers	
Wago Series 855	
Efergy Transformers (training box only)	

Table 11: Technical specifications of developed metering and analysis platform

4.4.4. Communication with the PERFoRM Middleware and to the machine tool

In this chapter the needed input and output data flows from the Min-Max-Data Mining Toolbox to the PERFoRM system are discussed. A necessary operational prerequisite of the proposed solution is the record from the high resolution current profile, current harmonics and phase shift at the electrical main connection of the machine tool as an **input** signal. This data is acquired by current transformers (CT) in combination with voltage metering (see **Figure 38**). More detailed information about the utilised system is available in chapter 4.4.3. As an optionally function the **input** from available maintenance reporting systems can be used to evaluate findings from the classification and the detection of abnormal behaviour of metered signals. The data of maintenance reporting systems is acquired by the PERFoRM Middleware as a device and system independent communication infrastructure.

As a primary **output** the toolbox returns indication of abnormal behaviour on component level. This data is generated by a deviation vector in dependence on the existent data sets (c.f. 4.4.5) and is distributed via a MQTT connection to the PERFoRM Middleware. The collected data acquired by CTs and the voltage metering is published by a WebSocket connection to the PERFoRM Middleware. Other participants can profit from the distributed data (e.g. utilisation of the data for KPI definition).

4.4.5. Data mining framework

In order to develop a promising data mining framework a combination of several subtasks must be considered. **Figure 35** facilitates the overview of the different modules within the data mining framework.

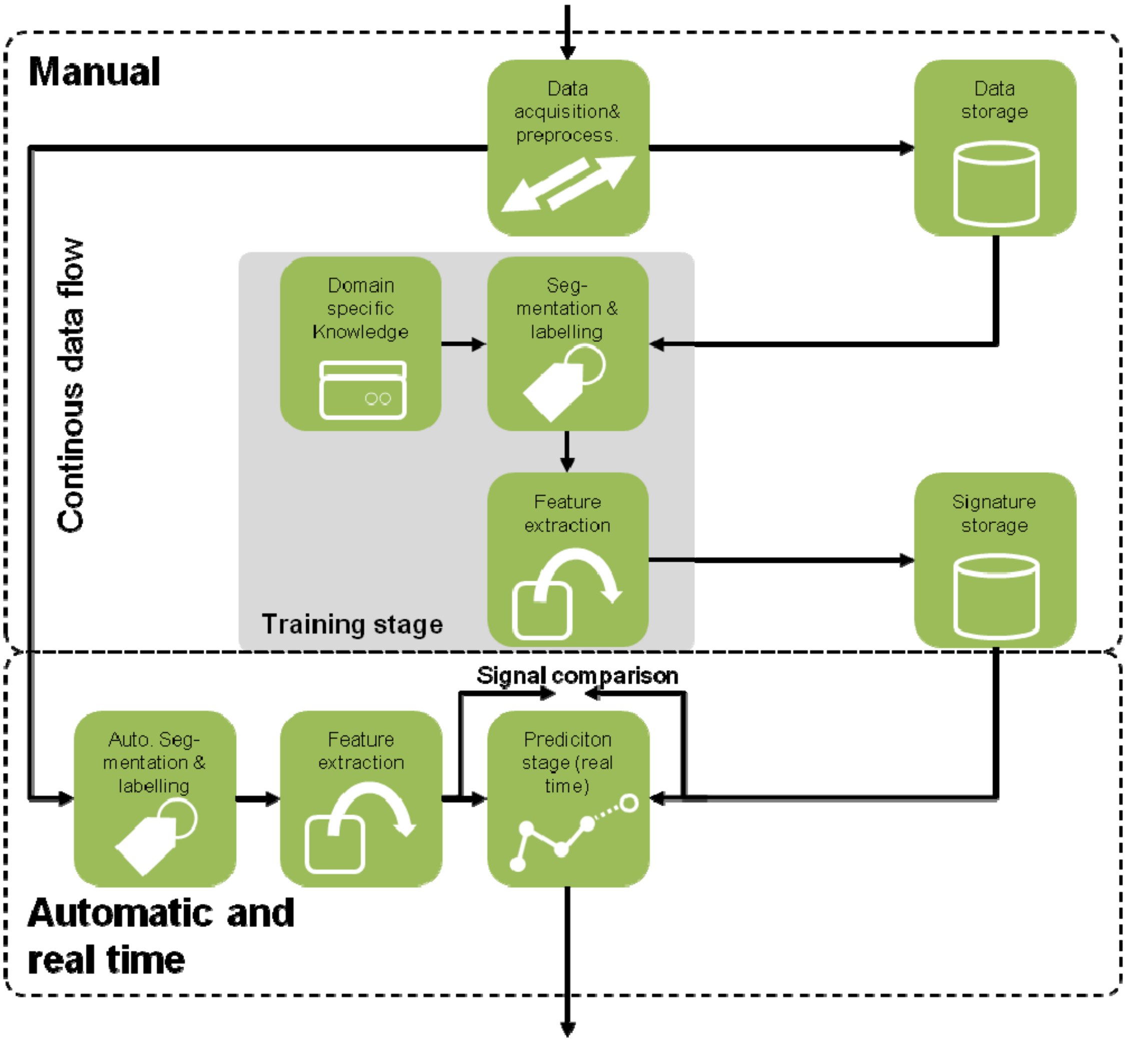


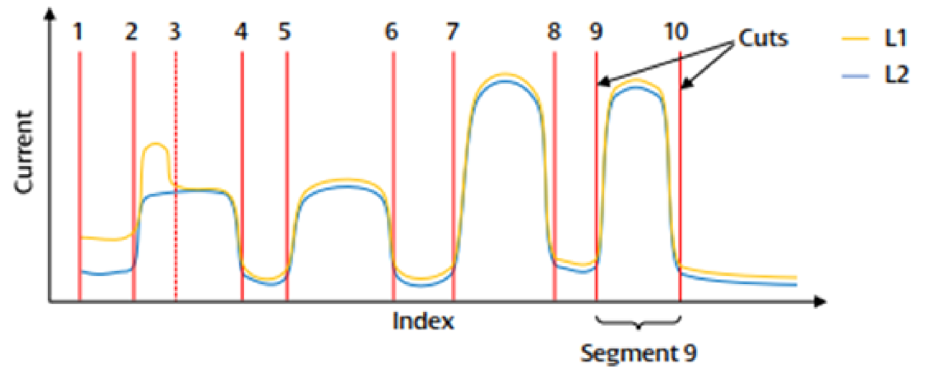
Figure 35: Data mining framework

As a first step, the data scientist has to gain **Domain Specific Knowledge** regarding the machine tool and the periphery. This includes knowledge of the electrical wiring and the structure and behaviour of components inside and outside the machine tool. The domain specific knowledge is a fundamental requirement for the following analysis and classification of the acquired data within the training and the prediction stage.

The pre-requisite for a successful data analysis implementation of the targeted machinery is the consistent and entire acquisition of relevant machine data. The **Data Acquisition module** is necessary to gain a gapless dataset of all relevant drives. It collects data from the Industrial Ethernet bus system (cf. chapter 4.4.4), performs pre-calculations and distributes the data to the data storage and the data preparation module. The data distribution takes place in event driven environment that is capable of asynchronous input and output processing. The distribution to the **Data Storage** is realised by a driver called “node-mysql” and event driven insertions using a permanent connection pool to the MySQL Database hosted on the ODROID X4U single board computer. The data storage contains the current profiles of all metering points (c.f. chapter 4.4.6) and the relating timestamps. The stored data is used to acquire the labelled signatures from the whole current profile for training stage after **preprocessing**. The conversion of analogue current transformer signals to a digital signal is always accompanied by disturbances. These disturbances are bigger if the measurement signal is very low. To gain a higher classification success of noisy signals, several approaches are suggested in the literature to preprocess the metered data [18]:

- Robust learners
- Data polishing methods
- Noise filters

The module **Segmentation and Labelling** plays a distinct role in the training process and in the prediction process as it will be one of the main pre-requisites to get distinguished component based patterns to extract features on. The metered data is available as time-series of the whole test run pattern (c.f. chapter 4.4.6). It is necessary to break down the whole test pattern to single component based time series associated with the different operation periods. The segmented time series and the resulting distinguishable operation periods are then labelled for an automatic recognition. Domain specific knowledge about the different consumers is needed for a correct labelling of the different segments and enables signature creation for later classification processes. Due to randomly activated non-controllable components during the training pattern (c.f. chapter 4.4.6), a class imbalance problem occurs. **Figure 36** depicts the segmentation process. In this example between cut 2 and 3 a non-plannable state occurs because of the random activation of a non-controllable component.



Cut No.	1	2	3	4	5	6	7	8	9	10
Cut Index	1	6	11	16	21	31	36	46	51	56
Nominal	1	0	0	1	1	1	1	1	1	1
Class	12	14	2	0	3	0	4	0	5	0

Figure 36: Segmentation process

A minimum number of valid segments are needed to fulfil the minimum requirement to gain the necessary classifier accuracy. The creation of synthetic samples with algorithms such as SMOTE [19] generates poor classification results working on current samples.

The **Training Stage** includes the selection of relevant time series segments. Due to the electricity grid-synchronous update time of the metering hardware [20], varieties in speed in the recorded segments appear accordingly to the actual power line frequency during the metering. These varieties can be adjusted by use of algorithms such as Dynamic Time Warping (DTW) [21]. DTW is used to measure similarity between two signals varying in speed. DTW is applied on the metered time series by using a pair wise comparison to find a minimum distance dimension.

To generate signatures out of the training stage **feature extraction** on the segmented, labelled and consolidated data takes place. The feature extraction and the resulting feature vector are completely machine and process dependent and contain time and frequency related features.

4.4.6. Predetermined test run

In order to create a dataset that contains all relevant component and machine behaviour information a test run containing predetermined patterns is needed. In this context a distinction between controllable and non-controllable components is necessary. The controllable components are directly controllable by the machine's programmable logical controller. All non-controllable components are connected to an own closed loop control or switched on and off time-controlled and are therefore independent from the machines programmable logical controller. According to **Figure 40** an exemplary breakdown of the controllable and non controllable components was carried out for the Siemens Carnaghi AC32 machine. With the information of controllable and non-controllable components in mind a test run can be designed. During the test run all controllable machine tool components are switched on and off in a known sequence to enable a component specific allocation within the recorded current-signature. The non-controllable components have to be metered separately to detect overlapping activity and the related characteristic behaviour. Due to a high number of occurring states, not every combination of controllable and non-controllable states can be considered. States with a very low number of samples will be neglected.

4.4.7. Application at Siemens

The Min-Max Data Mining Toolbox is a software and hardware solution to detect abnormal machine and machine component behavior by monitoring the current consumption of a machine tool. Currently, it is under implementation as a demonstrator within the Siemens use case. The advantage of the proposed solution installed in the Siemens turbo compressor factory is the use of electricity as the solely needed data input. Additional capacity can be achieved by the use of maintenance reporting information from the shop floor. Further and more expensive sensors, e.g. vibration sensors, are not necessary. The solution consists of two different main parts. On the one hand, a hardware solution to measure the actual current consumption based on commercially available power meter terminals. On the other hand, a software solution to analyze and assign gathered data real time on the fly. The custom-built hardware solution, depicted in **Figure 37**, is directly connected to the electrical cabinet of the three Carnaghi machines considered in the Siemens use case. All relevant consumers and components connected to the machine tool are comprised by the machine’s main electrical distribution. The metering of the main electrical connection is sufficient for the operation in prediction mode. Data recording for training purposes takes place with a particular training box to measure controllable and non-controllable components.

Figure 37: Hardware boxes to measure the main electrical machine connection

Figure 38 depicts a temporarily installed metering box in the electrical cabinet of the Carnaghi AC32. Visible is the connection to the current transformers (current metering) and the CEE plug for the voltage metering. Every connection is designed for the metering of 3 phases.

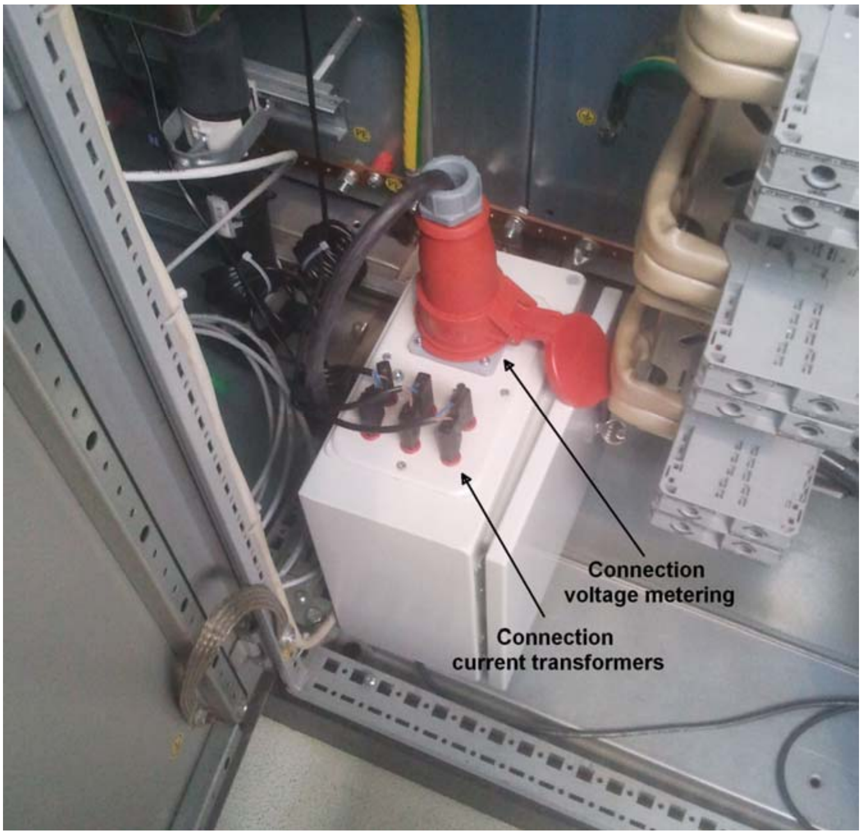


Figure 38: Temporarily installed metering box in the electrical cabinet of the AC32 with connected plugs for current and voltage metering

At the main electrical connection the actual voltage and current via current transformers of all three phases is measured with a frequency of 50 Hz (local power grid frequency) and saved to a local data-base. The used frequency is sufficient to conduct analysis in time and frequency domain. Also, short time process changes can be detected. The software solution is

1. used to query the measurement bus system and transform the digital signals into physically understandable data,
2. to analyze the measured data on the fly,
3. to save the data into a local designated data-base and
4. to communicate with the PERFoRM Middleware.

The commercial measurement system based on the bus protocol EtherCAT is queried by C libraries (EtherCAT master) running under a Linux environment. Further transfer of the signal into physically understandable values is carried out by a C program. Further data distribution and analysis is realised by node.js. Node.js allows to apply a data analysis on the fly. Program modules automatically detect machine operating patterns during a predetermined test run of the machine and can then detect anomalies regarding the reference condition of the machine and the machines components. **Figure 39** depicts all elements of the proposed system connected to the PERFoRM middleware. The "TUBS Data Acquisition" module is connected directly to the machine. The previously described hardware and the data acquisition software are located here. Data acquisition is realised by current transformers at the machine's electrical main connection. The "TUBS Data Mining" module is the heart of data processing. Under consideration of previously recorded "fingerprints" of the machine conditions, the deviation of the current status of components is calculated. These “fingerprints” and information of failure behaviour regarding the machines components are kept within the storage.

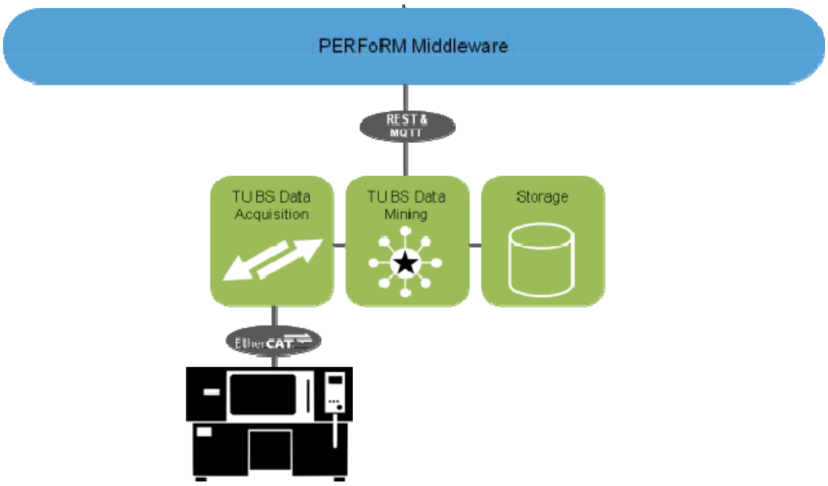


Figure 39: Connection of the TUBS Min-Max Data Mining Toolbox to the machine tool and the middleware

As described in Chapter 4.4.4 a connection to the PERFoRM Middleware was implemented by use of a REST API and a MQTT publisher-subscriber based messaging protocol in Node.js. The GET method is used for transfer of the PERFoRM-ML (PML) file containing the MQTT topics for data transfer to and from the Min-Max Data Mining Toolbox. The actual state allows sending measurement data from the TUBS Data Acquisition module to the PERFoRM Middleware and receiving data from the Siemens MCIS and LHnet system via the middleware. In the later development the data from the Siemens maintenance reporting system (LHnet) can be used to verify the findings from the classification process of the Min-Max Data Mining Toolbox.

As described before, the domain knowledge – in this case a well known machine architecture – is the requirement to apply the presented methodology. For the three vertical lathes within the Siemens turbo compressor factory a detailed analysis of the machine setup was conducted. Circuit diagrams are used for instance to have a detailed view on all components connected via the main electricity supply of the machines. The identification and classification of the existing components according to **Figure 40** is an essential step for the successful creation of the proposed predetermined test run (c.f. chapter 4.4.6). Every controllable component (e.g. Querbalken CR_CHANG(1)) has a label on the left side with the corresponding G-Code for activation by the machine’s programmable logical controller. The label on the right tags the distribution point and the number of phases in the electrical cabinet of the machine (e.g. Pumpe Kettenmagazin 4 kW 3~ 22Q4 161 K3). The words off and on label a steady state of the component.

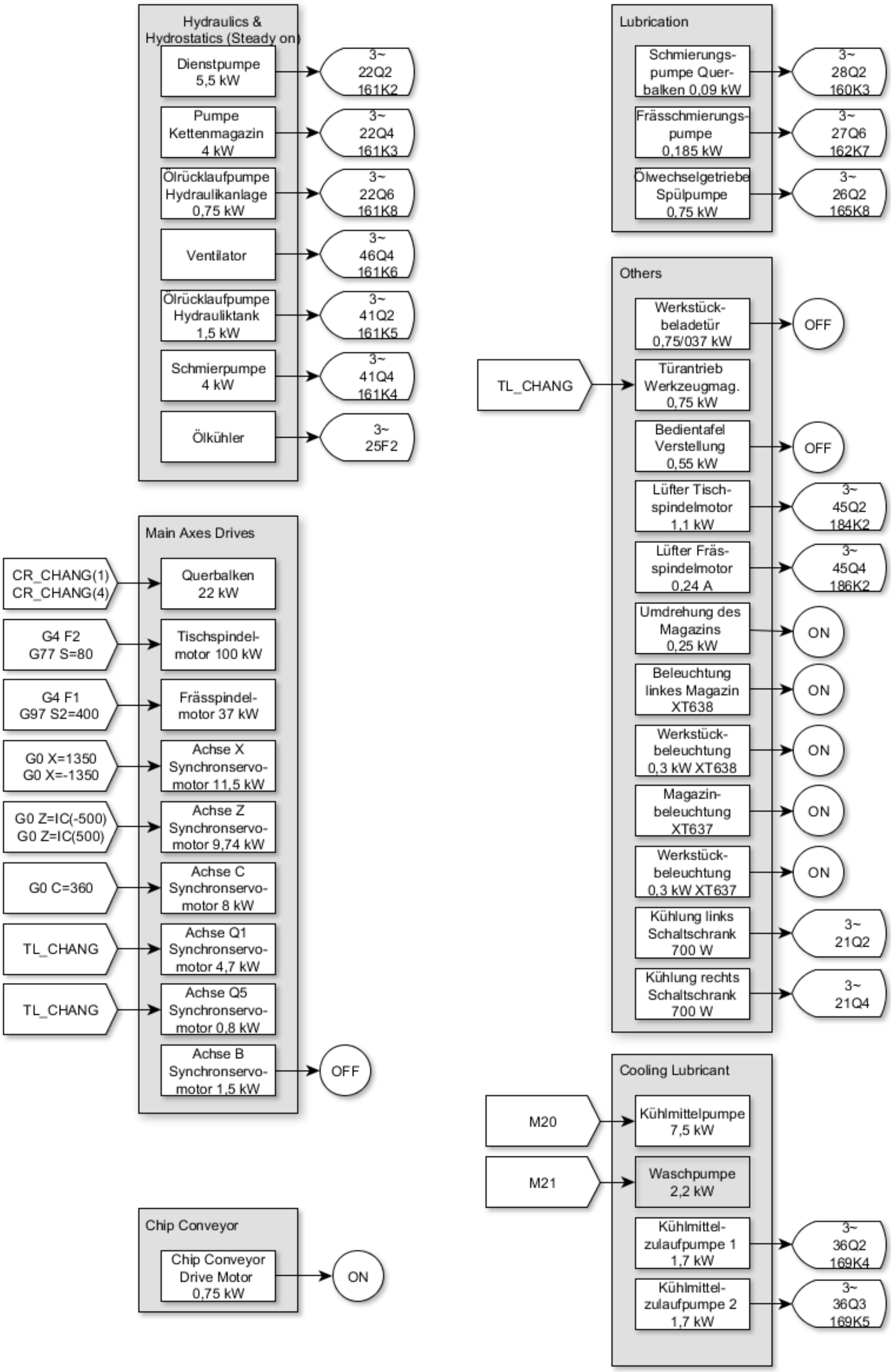


Figure 40: Overview and assignment to super-ordinate categories of the Carnaghi AC32 components

5. Conclusion

In the following chapter a short summary of the document, the next steps and the industrial impact and benefits of the solutions are presented.

This deliverable contains solutions regarding tools for visualisation and decision support with the aim to enable flexible and reconfigurable production systems. The presented solutions focus mainly on data-driven systems for maintenance support and on KPI and KBF guided decision support. Additionally web-based visualisation modules enable the production engineer to evaluate implemented solutions. .All solutions pursue the objective of implementation into the industrial use cases with due regard to the PERFoRM architecture.

Although most of the solutions presented in this document are in a highly advanced development stage and are ready for implementation or already in an implementation stage, the adoption to a highly reliable working system for industrial application needs to be accomplished. Therefore, methodological preliminary work, testing and a transition into a demonstration phase are successfully completed in WP4. For later application the developed solutions are case-specific and will be finalised by reference to the requirements of the designated use cases. The validation of the applicability of developed tools and methods are performed in WP5. Therefore the developed methods and tools have been discussed or are currently in discussion with Siemens, Whirlpool, GKN and the SmartFactory. The following practical implementation of the presented solutions smooth the way to more flexible and predictable production systems. The industrial potential and impact will be revealed within the demonstration WPs.

The presented solutions and the application enable opportunities in particular in the industrial sector. In the following the issues and the expected benefits for the technology-driven industry are presented. With more sophisticated products and digitalisation processes the complexity and maintenance frequency in manufacturing increases steadily. Additionally, the trend towards customer-oriented lot-size 1 and the resulting increased need for flexibility have to be handled. In order to make the increasing complexity controllable smart manufacturing assessment and maintenance system have to be established. Also, to reduce resulting follow-up costs. According to the research already carried out, the ability for flexible rescheduling and reconfiguration of production chains can be achieved with analysis of existing and additional factory data. This data is not always easy to access, and it is often not available with the required accuracy. For proper comprehension of available data, knowledge about the target domain and the technical circumstances is essential. The capability of identifying relationships between technical occurrences and specific machine components is therefore fundamental and enables the successful evaluation of present data. For example, the knowledge about interactions between the probability of failure occurrence and data-driven analytics is fundamental for failure predictions in the future. In order to assess the increased customisation and flexibilisation opportunities, a homogenous data pooling and the use of a common visualisation interface is essential. This enables the production scheduler on the one hand to choose between different scheduling possibilities and to select the most promising option. On the other hand, the most efficient plant capacity utilisation can be achieved.

References

[1] S. Gulabani, „Developing RESTful Web Services with Jersey 2.0," *Packt Publishing,* 2013.

[2] OASIS, „MQTT Version 3.1.1 Plus Errata 01," [Online]. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.doc. [Zugriff am December 2015].

[3] F. Boschi, C. Zanetti, G. Tavola, M. Taisch, P. Leitão, J. Barbosa und A. Pereira, „From key business factors to KPIs within a reconfigurable and flexible Cyber-Physical System".*Submitted to ICE 2017 IEEE TMC Europe Conference.*

[4] D3.js, „D3.js - Data-Driven Documents," [Online]. Available: https://d3js.org/. [Zugriff am 23 May 2017].

[5] D. C. Montgomery, Introduction to Statistical Quality Control, John Wiley & Sons, Inc., 2009.

[6] W. Electric, Statistical Quality Control Handbook, Western Electric Corporation, 1956.

[7] P. Leitão, N. Rodrigues und J. Barbosa, „What-if game simulation in agent-based strategic production planners," *20th Conference on Emerging Technologies & Factory Automation (ETFA'15),* 2015.

[8] M. A. Djeziri, „Robust fault diagnosis by using bond graph approach," *IEEE/ASME Transactions on Mechatronics,* pp. 599-611, 2007.

[9] R. Milne, M. Drummond und P. Renoux, „Predicting paper making defects on-line using data mining," *Knowledge-Based Systems,* Bd. 11, pp. 331--338, 1998.

[10] M. N. Tomlinson, „What is the ideal maintenance strategy? A look at both MoD and commercial shipping best practive.," *INEC,* 2016.

[11] P. Bastos, L. Isabel und C. M. Pores, „Application of data mining in a maintenance system for failure prediciton," *Safety, Reliability and Risk Analysis* , pp. 933-940, 2014.

[12] C. Siemens AG, „Siemens description and requirements of architectures for retrofitting production equipment," PERFoRM project, 2016.

[13] MTC, „Report on Self-Adaptive Machines Demonstrator Design and Set-up," PERFoRM project, 2016.

[14] M. M. Rahman und D. D.N., „Addressing the Class Imbalance Problem ind Medical Datasets," *International Journal of Maachine Learning and Computing,* pp. 224-228, 2013.

[15] N. V. Chawla und K. W. Bowyer, „SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research, vol. 16,* pp. 321-357, 2002.

[16] R. Barros, „A Survey of Evolutionary Algorithms for Decision-Tree Induction," *, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 42, no. 3,* pp. 291-312, 2012.

[17] Apache, „Apache Lucene," [Online]. Available: https://lucene.apache.org/. [Zugriff am 10 May 2017].

[18] S. Garcia, J. Luengo und F. Herrera, Data Preprocessing in Data Mining, Berlin: Springer, 2015.

[19] N. V. Chawla und K. W. Bowyer, „SMOTE: Synthetic Minority Over-sampling Technique".

[20] Beckhoff Automation , [Online]. Available: http://www.beckhoff.de/default.asp?ethercat/el3403.htm.

[21] H. Sakoe und S. Chiba, „Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING,* February 1978.