NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
DIVISION OF INDUSTRIAL ELECTRIC DEVICES AND DECISION SYSTEMS

Proactive Computing in Industrial Maintenance Decision Making

PhD THESIS

Alexandros D. Bousdekis

Athens, July 2018

NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
DIVISION OF INDUSTRIAL ELECTRIC DEVICES AND DECISION SYSTEMS

Proactive Computing in Industrial Maintenance Decision Making

(Προδραστική Πληροφορική στη Λήψη Αποφάσεων Βιομηχανικής Συντήρησης)

PhD THESIS

Alexandros D. Bousdekis

Advisory Committee: *Gregoris Mentzas,* Professor NTUA (supervisor)
Ioannis Psarras, Professor NTUA
Rui Neves-Silva, Associate Professor
Universidade Nova de Lisboa

Approved by the seven-member examination committee in 4th July 2018.

Gregoris Mentzas Professor NTUA	*Ioannis Psarras* Professor NTUA	*Rui Neves-Silva* Associate Professor Universidade Nova de Lisboa
Dimitris Askounis Professor NTUA	*Ilias Tatsiopoulos* Professor NTUA	*Dimitris Apostolou* Associate Professor University of Piraeus
	Vrassidas Leopoulos Professor NTUA	

Athens, July 2018

……………………………….

Alexandros D. Bousdekis

Doctor of Engineering N.T.U.A.

Abstract

Proactive event-driven computing refers to the use of event-driven information systems having the ability to eliminate or mitigate the impact of future undesired events, or to exploit future opportunities, on the basis of real-time sensor data and decision making technologies. Maintenance management can benefit from these advancements in order to tackle with the increasing challenges in today's dynamic and complex manufacturing environment in the context of Industry 4.0.

To this end, the current thesis combines and brings together the research fields of Industry 4.0, Maintenance Management and Proactive Computing in order to frame maintenance management and information systems in the context of Industry 4.0. Therefore, it paves the way for the next generation of maintenance management in the frame of Industry 4.0, i.e. Proactive Maintenance. The focus of the current thesis is on proactive decision making. Consequently, it proposes proactive decision methods, capable of handling uncertainty, applicable to maintenance management and its interrelationships with other manufacturing operations, algorithms for continuous improvement of proactive decision making through the proposed Sensor-Enabled Feedback (SEF) approach and algorithms for context-awareness in proactive decision making. To do this, it utilizes methods and techniques for operational research, data analytics and machine learning.

The aforementioned algorithms have been embedded in a proactive information system for decision making which was integrated with other tools in order to implement all the steps of the Proactive Maintenance framework. The system has been deployed and evaluated in real industrial environment, while further evaluation was conducted with extensive simulation experiments. Finally, the lessons learned and the managerial implications of the proposed approaches are discussed.

Keywords: Industry 4.0, Proactive Decision Making, Maintenance Management, Proactive Maintenance, Event Processing, Uncertainty

Περίληψη

Η προδραστική πληροφορική οδηγούμενη από γεγονότα αφορά τη χρήση πληροφοριακών συστημάτων οδηγούμενων από γεγονότα που έχουν την ικανότητα να εξαλείφουν ή να αμβλύνουν την επίδραση μελλοντικών ανεπιθύμητων γεγονότων ή να αξιοποιούν μελλοντικές ευκαιρίες με βάση δεδομένα αισθητήρων πραγματικού χρόνου και τεχνολογίες λήψης αποφάσεων. Η διοίκηση συντήρησης μπορεί να επωφεληθεί από την προδραστική πληροφορική για να αντιμετωπίσει τις προκλήσεις στο πλαίσιο της Βιομηχανίας 4.0 (Industry 4.0).

Για το σκοπό αυτό, η παρούσα διατριβή συνδυάζει τους ερευνητικούς τομείς της Βιομηχανίας 4.0, της Διοίκησης Συντήρησης και της Προδραστικής Πληροφορικής. Με αυτό τον τρόπο, ανοίγει το δρόμο για την επόμενη γενιά διοίκησης συντήρησης στο πλαίσιο της Βιομηχανίας 4.0, την Προδραστική Συντήρηση (Proactive Maintenance). Το επίκεντρο της διατριβής είναι η λήψη προδραστικών αποφάσεων. Συνεπώς, προτείνει μεθόδους προδραστικών αποφάσεων για βιομηχανική συντήρηση, αλγόριθμους για συνεχή βελτίωση της λήψης προδραστικών αποφάσεων μέσω της προτεινόμενης προσέγγισης Ανατροφοδότηση Υποβοηθούμενη από Αισθητήρες (Sensor-Enabled Feedback - SEF) και αλγόριθμους για την επίγνωση πλαισίου. Για να γίνει αυτό, αξιοποιεί μεθόδους και τεχνικές επιχειρησιακής έρευνας, ανάλυσης δεδομένων και μηχανικής μάθησης.

Οι προαναφερθέντες αλγόριθμοι έχουν ενσωματωθεί σε ένα προδραστικό πληροφοριακό σύστημα για τη λήψη αποφάσεων το οποίο ολοκληρώθηκε με άλλα εργαλεία για την υλοποίηση όλων των βημάτων του πλαισίου της Προδραστικής Συντήρησης. Το σύστημα εγκαταστάθηκε και αξιολογήθηκε σε πραγματικό βιομηχανικό περιβάλλον, ενώ πραγματοποιήθηκε περαιτέρω αξιολόγηση με εκτεταμένα πειράματα προσομοίωσης.

Λέξεις κλειδιά: Βιομηχανία 4.0, Προδραστική Λήψη Αποφάσεων, Διοίκηση Συντήρησης, Προδραστική Συντήρηση, Επεξεργασία Γεγονότων, Αβεβαιότητα

Acknowledgements

This doctoral dissertation marks the completion of a very pleasant and constructive effort undertaken within the postgraduate program of the School of Electrical and Computer Engineering of the National Technical University of Athens.

This effort could not have been accomplished without the permanent assistance, the guidance, encouragement and help of Professor Gregoris Mentzas. I am grateful for the trust he showed me and for the opportunity he gave me to test my capabilities and ideas within the research activities of the Information Management System Unit (IMU). I would like also to thank the Associate Professor of the University of Piraeus Dimitris Apostolou for his valuable advice and constructive comments that helped to improve my work as well as for his participation in the seven-meber examination committee of my PhD thesis.

I would like to thank the other two members of the three-member advisory committee, Professor Ioannis Psarras and Associate Professor Rui Neves-Silva, as well as Professor Dimitris Askounis, Professor Ilias Tatsiopoulos and Professor Vrassidas Leopoulos for the honor they have done to me to participate in the seven-member committee for the dissertation.

I would like also to thank all the members of IMU and especially Babis Magoutas, Nikos Papageorgiou and Katerina Lepenioti with whom I have collaborated in research projects and publications. Our cooperation has been a key source of support and inspiration for addressing and solving research challenges.

Finally, the biggest thank deserves to the people who stood next to me throughout the time of this effort: to my parents Dimitrios Bousdekis and Vassiliki Kougioumtzi, my sister Maria Bousdeki and my partner Evgenia Karampesini.

Table of Contents

List of Figures

List of Tables

List of Acronyms

BIC	Bayesian Information Criterion
BM	Breakdown Maintenance
BN	Bayesian Network
CBM	Conditioned-Based Maintenance
CPS	Cyber-Physical Systems
DDM	Derrick Drilling Machine
DMI	Decision Method Instance
ECA	Event-Condition-Action
EDA	Event-Driven Architecture
EDI	Electronic Data Interchange
EL	Expected Loss
ELR	Expected Loss Rate
ERP	Enterprise Resources Planning
FMEA	Failure Mode and Effects Analysis
FMECA	Failure Mode, Effects and Criticality Analysis
FSO	Full Scale Output
GUI	Graphical User Interface
ICT	Information and Communications Technology
IoT	Internet of Things
IT	Information Technology
IVHM	Integrated Vehicle Health Management
MDP	Markov Decision Process
MES	Manufacturing Execution System
MPT	Markowitz Portfolio Theory
MSE	Mean Squarred Error
MVC	Model View Controller
OEM	Original Equipment Manufacturer
OT	Operational Technology

PANDDA	ProActive seNsing enterprise Decision configurator DAshboard
PDF	Probability Distribution Function
PFI	Positive Feedback Indicator
PLC	Programmable Logic Controller
PLM	Product Lifecycle Management
RAMI 4.0	Reference Architectural Model Industrie 4.0
RLD	Remaining Life Distribution
RPM	Rounds Per Minute
RUL	Remaining Useful Life
SEF	Sensor-Enabled Feedback
SOA	Service-Oriented Architecture
SVM	Support Vector Machine
SWOT	Strengths, Weaknesses, Opportunities, and Threats
TBM	Time-Based Maintenance
WCSS	Within-Cluster Sum of Squares

1 Introduction

1.1 Motivation

The emergence of the Internet of Things (IoT) has paved the way for enhancing the monitoring capabilities of enterprises by means of extensive use of physical and virtual sensors. Taking advantage of the big data generated from a large amount of sensors requires the development of event monitoring and data processing systems that are able to handle real-time data in complex, dynamic environments in order to get meaningful insights about business performance. These advancements lead to the possibility to decide and act ahead of time, i.e. to be proactive in resolving problems before they appear or realizing opportunities before they become evident.

The potential of the proactive approach is high, especially in the complex and dynamic manufacturing environment in the context of the Industry 4.0 paradigm. Manufacturing operations are driven by events, which are increasingly collected through sensors and processed via real-time operational technology and systems. Therefore, any action, activity, or monitored parameter change, which influences the operational status of a manufacturing process is potentially a trigger for proactive decision making.

Every major shifting of manufacturing paradigm has been supported by the advancement of information technology (Bi et al., 2014; Mahmood, 2018). Modern manufacturing companies have started monitoring and detecting early warning signals that machines or systems are degrading or in danger of breakdown. While this is valuable information, if organizations improve their analytics maturity, information will be fully harnessed, and the potential is much greater.

Maintenance operations are a major part of the total operating costs of manufacturing plants as they can represent up to 40 % of the production process costs (Widodo and Yang, 2011), but also, they have an impact on reliability, safety and environment (Garg, and Deshmukh, 2006). Moreover, failure of critical assets has been rated as the most significant risk to operational performance (Aboelmaged, 2015). Due to the emergence of the new technologies and computing paradigms, several approaches, frameworks and architectures for intelligent maintenance have appeared both in the academic and the industrial realms (Pistofidis et al., 2012; Fumagalli, and Macchi, 2015; Macchi et al., 2018).

However, currently, there is still a lack of services and tools capable of efficiently processing real-time big data from heterogeneous sources, implementing complex algorithms and provide meaningful insights about potential problems in an event-driven streaming infrastructure (Engel et al., 2012; Camarinha-Matos et al., 2013). Moreover, there is a large gap for the effective implementation of predictive maintenance programs extensively in industry, mainly due to the complexity of these solutions and their life cycle and thus, due to the challenges in their practical implementation (Guillen et al., 2016).

Maintenance management in the frame of Industry 4.0 can take advantage of the recent advancements in proactive computing, for fully exploiting its capabilities and supporting decisions ahead of time. Previous approaches, e.g. in the field of predictive maintenance, concluded in offline or processing batches of data (Wu et al., 2007; Elwany, Gebraeel, 2008). However, in a streaming computational environment, appropriate methods, algorithms and systems need to be developed. Consequently, there is the need for services and tools that will provide real-time proactive decision making capabilities along with adaptation and context-awareness mechanisms in order to result in reliable maintenance recommendations.

1.2 Contribution

The current thesis combines and brings together the research fields of Industry 4.0, Maintenance Management and Proactive Computing in order to frame maintenance management and information systems in the context of Industry 4.0 taking advantage of proactive event processing in enterprise systems.

The contribution of the current thesis is summarized to the following dimensions:

1. **It paves the way for the next generation of maintenance management in the frame of Industry 4.0**. To this end, it presents a framework for a new maintenance strategy, i.e. Proactive Maintenance, its definition and characteristics, as well as its generic conceptual architecture. This architecture can be seen as a blueprint for the development of Proactive Maintenance information systems.
2. **It proposes proactive decision making**. Since proactive decision making is an unexplored area, the thesis develops proactive event-driven decision methods for maintenance management and its interrelationships with other manufacturing operations. To do this, it takes advantage of the area of Operational Research. It also embeds them in an information system capable of being integrated with systems incorporating real-time detection/ diagnostic and prediction/ prognostic algorithms.
3. **It proposes continuous improvement of proactive decision making**. The thesis develops an approach for Sensor-Enabled Feedback (SEF) in order to improve the accuracy of proactive decision methods and consequently, the reliability of the generated proactive recommendations. To do this, it takes advantage of the area of Data Analytics and Anomaly Detection. This approach is embedded in the information system along with the methods and algorithms of the previous direction.
4. **It proposes context-awareness in proactive decision making**. The thesis develops an approach for context-awareness in proactive decision making utilizing machine learning techniques. In this way, it can tackle with uncertainty in intelligent decision making. To do this, it takes advantage of the area of Data Analytics and Machine Learning. The context-aware mechanism is continuously updated through SEF.

1.3 Relation to Research Projects

The current PhD thesis has been partly financially supported by the following European Commission projects:

a) **ProaSense (Proactive Sensing Enterprise)**[1], Research and Innovation Action project under Grant Agreement 612329, FP7-ICT-2013-10 (Framework Program 7 – Information and Communication Technologies), ICT-2013.1.3 - Digital Enterprise.
   - The vision of the project is to pave the way for a new class of enterprise systems, proactive enterprises, that will be continuously aware of what "might happen" in the relevant business context and optimize their behavior to achieve that what "should be the best action".
   - The main parts of the current thesis are based upon the work conducted in the context of the ProaSense project.

b) **UPTIME (Unified PredicTIve Maintenance system)**[2], Innovation action project under Grant Agreement 768634, H2020-FOF-2017 (Horizon 2020 – Factories-Of-the-Future), FOF-09-2017 - Novel design and predictive maintenance technologies for increased operating life of production systems.
   - UPTIME aims to design a unified predictive maintenance framework and an associated unified information system in order to enable the predictive maintenance strategy implementation in manufacturing industries.
   - The current thesis has been enriched based upon the work conducted in the context of the UPTIME project. More specifically, the literature review was further extended and the framework for Proactive Maintenance was finalized.

[1] http://www.proasense.eu/
[2] https://www.uptime-h2020.eu/

1.4 *Research Design and Structure of the Dissertation*

The research design and methodology of the current thesis is depicted in Figure 1-1. Table 3-1 presents the contents of each step of the adopted research methodology as well as the Chapters of the thesis to which they correspond.

The first step deals with a **Literature Review** on the background concepts: Industry 4.0, Maintenance Management and Proactive Enterprise with the aim to identify their interrelationships. To this end, a synthesis of the literature review is conducted and the research area and focus is identified. The second step deals with posing the research questions and outlining the proposed solution of the thesis aiming to pave the way **Towards Proactive Maintenance Management**. The third step deals with the development of a **Framework for Proactive Maintenance**. In this step, Proactive Maintenance is defined and the conceptual architecture is developed. The Proactive Maintenance conceptual architecture is compatible with RAMI 4.0.

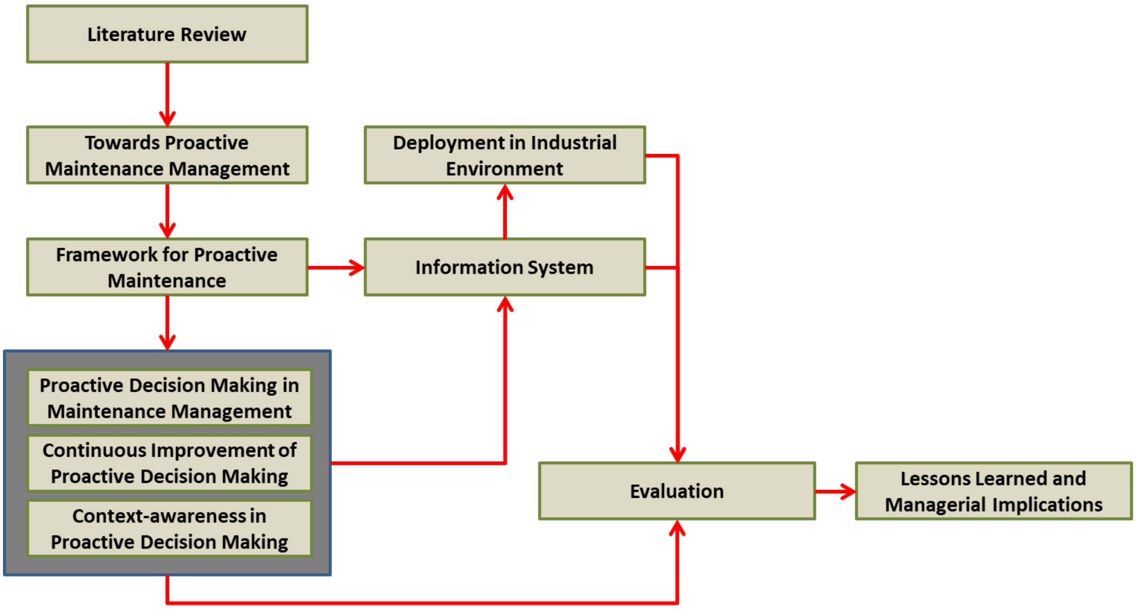

Figure 1-1: The Research Design and Methodology

Table 1-1: The Contents of each Research Methodology Step

Research Methodology Step	Contents
Literature Review *(Chapter 2)*	• Identification of research area and focus • Review of literature regarding the background concepts: o Industry 4.0 o Maintenance Management o Proactive Enterprise • Synthesis of the literature review
Towards Proactive Maintenance Management *(Chapter 3)*	• Research Questions • The Thesis
Framework for Proactive Maintenance *(Chapter 4)*	• Definition of Proactive Maintenance • Concept of Proactive Maintenance • Conceptual Architecture of Proactive Maintenance
Proactive Decision Making in Maintenance Management *(Chapter 5)*	• Motivation • State-of-the-art analysis • Development of methods and functionalities
Continuous Improvement of Proactive Decision Making *(Chapter 6)*	• Motivation • State-of-the-art analysis • Development of methods and functionalities
Context-awareness in Proactive Decision Making *(Chapter 7)*	• Motivation • State-of-the-art analysis • Development of methods and functionalities
Information System *(Chapter 8)*	• The PANDDA system for proactive decision making o Architecture o Design and Development o User Interface and Walkthrough
Deployment in Industrial Environment *(Chapter 9)*	• The MHWirth Business Case o Description o Deployment use cases • The HELLA Business Case o Description o Deployment use cases
Evaluation *(Chapter 10)*	• System evaluation by users • System performance evaluation • Sensitivity Analysis and Comparative Analysis of the implemented functionalities • Discussion of evaluation results
Lessons Learned and Managerial Implications *(Chapter 11)*	• Lessons Learned • Managerial Implications

The fact that the focus of the thesis is on proactive decision making (which is the least explored area of the aforementioned framework) triggers jointly the subsequent three steps of the research methodology, i.e. **Proactive Decision Making, Continuous Improvement of Proactive Decision Making, Context-awareness in Proactive Decision Making**. For each one of them, a more focused state-of-the-art analysis is conducted in order to reveal the research gaps. On the basis of these research gaps, the approaches, methods, models and functionalities are developed.

The step incorporating the **Information System** takes place based upon both the Framework for Proactive Maintenance and the Proactive Decision Making, Continuous Improvement of Proactive Decision Making, Context-awareness in Proactive Decision Making steps. In this step of the research methodology, the ProActive seNsing enterprise Decision configurator DAshboard (PANDDA) system for proactive decision making is designed, developed and implemented incorporating the previously mentioned functionalities. Finally, it is integrated with an overall system implementing the Proactive Maintenance framework.

The next step of the adopted research methodology deals with the **Deployment in Industrial Environment**. More specifically, the PANDDA system which incorporates the aforementioned functionalities is deployed in two business cases as part of an overall Proactive Maintenance information system. These two business cases are: MHWirth, an oil drilling manufacturing company, and HELLA Saturnus Slovenija, an automotive lighting equipment manufacturing company. The **Evaluation** step takes place through system evaluation by users, system performance evaluation and extensive simulation experiments for conducting sensitivity and comparative analyses of the adopted functionalities. On the basis of these results, a discussion of results is presented. Finally, in the last step of the research methodology, the **Lessons Learned and Managerial Implications** of adopting a Proactive Maintenance strategy in Industry 4.0 are discussed.

2 Literature Review

In this Chapter, the literature review on the background concepts of Industry 4.0, Maintenance Management and Proactive Enterprise is presented. These three background concepts are reviewed with the aim to identify their interrelationships. The research area and focus of the current thesis is the intersection of all the three terms.

2.1 Scope of the Literature Review

The literature review presents a qualitative analysis of research done to date on the background concepts of "Industry 4.0", "Maintenance Management" and "Proactive Enterprise" and their interrelationships with two basic objectives: (i) to identify the topic set studied; and (ii) to discuss the available empirical evidence, detecting contradictions and inconsistencies in the literature as well as the research gaps that should be fulfilled. The literature review on the aforementioned background concepts aims to identify the interrelationships found to date among these concepts, as shown in Figure 2-1. This Figure actually shows the scope of the current literature review. Each circle is further analyzed in the course of the literature review and enables identifying the background concepts' interrelationships, their overlaps and their gaps based on Figure 2-1.

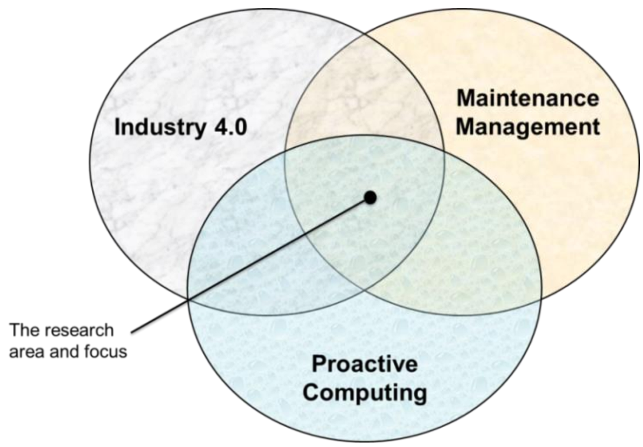

Figure 2-1: The scope of the literature review

2.2 Industry 4.0

2.2.1 Overview

In the manufacturing realm, the advances of science and technology continuously support the development of industrialisation all around the world (Belvedere, et al., 2013). From a technological evolution perspective, there are four industrial revolutions commonly identified (Kagermann, et al., 2013). The first three industrial revolutions took around two centuries, and are the result of, respectively: (1) the introduction of water and steam-powered mechanical manufacturing facilities; (2) the application of electrically-powered mass production technologies through the division of labour; and (3) the use of electronics and information technology (IT) to support further automation of manufacturing (Drath and Horch 2014). In recent years, along with the increased research attention on the Internet of Things (IoT) (Atzori, et al., 2010) and Cyber-Physical Systems (CPS) (Khaitan, and McCalley, 2015), governments and industries worldwide have noticed this trend and acted to benefit from what this new industrial revolution wave could provide (Ridgway, et al., 2013; Siemieniuch, et al., 2015; Liao et al., 2017):

(i) From the government plans perspective,

- Since 2011 the United States (US) government began a series of national-level discussions, actions and recommendations, titled 'Advanced Manufacturing Partnership (AMP)', to ensure the US to be prepared to lead the next generation of manufacturing (Rafael, et al., 2014).
- In 2012, the German government passed the 'High-Tech Strategy 2020' action plan, which annually sets billions of euros aside for the development of cutting-edge technologies. As one of the ten future projects in this plan, the 'Industrie 4.0' represents the German ambitions in the manufacturing sector (Kagermann, et al., 2013).
- The French government initiated a strategic review in 2013, named the 'La Nouvelle France Industrielle', in which 34 sector-based initiatives

are defined as France's industrial policy priorities (Conseil national de l'industrie, 2013).

- In 2013, the United Kingdom (UK) government presented a long-term picture for its manufacturing sector until the year of 2050, called the 'Future of Manufacturing'. It aims to provide a refocused and re-balanced policy for supporting the growth and resilience of UK manufacturing over the coming decades (Foresight, 2013).
- The European Commission lunched the new contractual Pubic-Private Partnership (PPP) on 'Factories of the Future (FoF)' under the Horizon 2020 programme in 2014 (European Commission, 2016).
- In 2014, the South Korea government announced the 'Innovation in Manufacturing 3.0' that emphasised four propulsion strategies and assignments for a new leap of Korean manufacturing (Kang et al., 2016).
- The Chinese government issued the 'Made in China 2025' strategy alongside the 'Internet Plus' plan in 2015. It prioritises ten fields in the manufacturing sector to accelerate the informatization and industrialisation in China (Li, 2015).
- In 2015, the Japanese government adopted the 5th Science and Technology Basic Plan, where particular attentions have been paid to the manufacturing sector for realising its world-leading 'Super Smart Society'. (Cabinet Office, 2015)
- The Singapore government has committed $19 billion to its RIE 2020 Plan (Research, Innovation and Enterprise) in 2016. Eight key industry vertical have been identified within the advanced manufacturing and engineering domain (National Research Foundation, 2016).

(ii) From the industrial plans perspectives,

- AT&T, Cisco, General Electric, IBM and Intel founded the 'Industrial Internet Consortium (IIC)' in 2014 to catalyse and coordinate the priorities and enabling technologies of the Industrial Internet (Evans and Annunziata, 2012).

- Meanwhile, other big firms like Siemens, Hitachi, Bosch, Panasonic, Honeywell, Mitsubishi Electric, ABB, Schneider Electric and Emerson Electric have also already invested heavily in IoT and CPS related projects.

The Industrial Revolution is a concept and a development that has fundamentally changed our society and economy. The term “development” may seem to indicate some tardiness in the context of a “revolution,” which really signifies a rapid and fundamental change, but there is no doubt that major alterations occurred within a relatively short period. Industries arose and replaced small-scale workshops and craft studios. Textile and pottery factories were the first to recognize the new dawn, and a new infrastructure of canals and railway lines enabled efficient distribution. It was the transition from industrious to industrial, and the start of a boom for both. From the first mechanical loom, dating from 1784, 234 years ago, we can distinguish four stages in the ongoing process called the Industrial Revolution.

That is the way we currently look at it. The first “acceleration” occurred toward the end of the 18th century: mechanical production on the basis of water and steam. The Second Industrial Revolution is placed at the beginning of the 20th century: the introduction of the conveyor belt and mass production, to which the names of icons such as Henry Ford and Frederick Taylor are linked. Number three is the digital automation of production by means of electronics and IT.

The fourth industrial revolution, known as Industry 4.0, includes intelligent production, IoT technologies and CPS aiming to bring together (Information Technology (IT) and Operational Technology (OT). The four industrial revolutions are shown in Figure 2-2. Currently, industry is found at the edge of the third and the fourth industrial revolutions. However, a lot of aspects should be explored in depth as shown in Figure 2-3 and Figure 2-4.

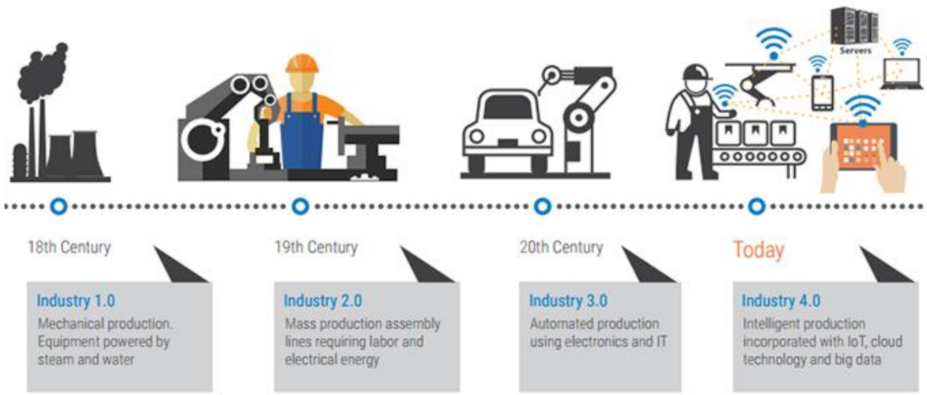

Figure 2-2: The four industrial revolutions (Source: BCM Advanced Research)

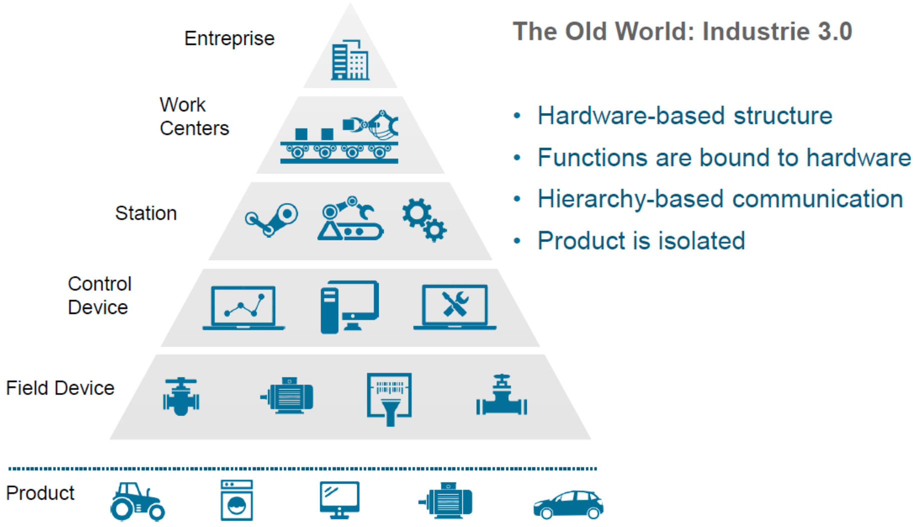

Figure 2-3: From Industry 3.0… (Source: Platform Industrie 4.0)

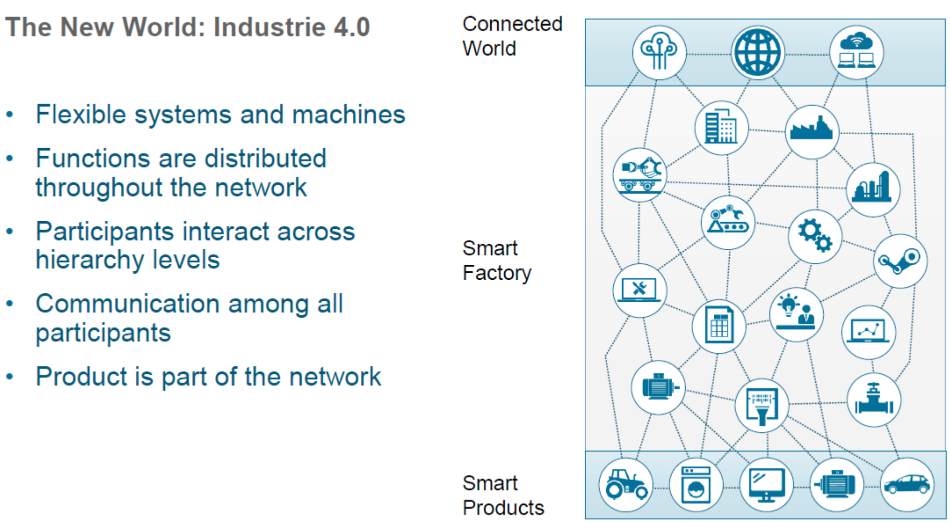

Figure 2-4: … to Industry 4.0 (Source: Platform Industrie 4.0)

The basic principle of **Industry 4.0** is the core of IoT and smart manufacturing: work in progress products, components and production machines will collect and share data in real time. This leads to a shift from centralized factory control systems to decentralized intelligence (Shrouf et al., 2014). The German Federal Ministry of Education and Research defines Industry 4.0 as “the flexibility that exists in value-creating networks is increased by the application of CPS. This enables machines and plants to adapt their behavior to changing orders and operating conditions through self-optimization and reconfiguration. The main focus is on the ability of the systems to perceive information, to derive findings from it and to change their behavior accordingly, and to store knowledge gained from experience. Intelligent production systems and processes as well as suitable engineering methods and tools will be a key factor to successfully implement distributed and interconnected production facilities in future Smart Factories”. Industry 4.0 core elements are depicted in Figure 2-5.

The intense employment of IoT technology and cyber-physical systems across the industrial value chain leads to huge amounts of heterogeneous data. Perceiving information and extracting business insights and knowledge from these data along with the knowledge storage gained from experience is one of the major challenges in Industry 4.0 (Gölzer et al., 2015; Gröger et al., 2016; Gröger, 2018). Intelligent production systems and processes as well as suitable engineering methods and tools (e.g. data analytics techniques) will be a key factor to successfully implement distributed and interconnected production facilities in future Smart Factories. To represent these issues, RAMI 4.0, a 3D model representing all different manually interconnected features of the technical – economical properties has been developed[3].

Industry 4.0 combines production methods with state-of-the-art information and communication technology. The driving force behind this development is the rapidly increasing digitisation of the economy and society. In the world of Industrie 4.0, people, machines, equipment, logistics systems and products communicate and cooperate with each other directly. Production and logistics processes are integrated

[3] http://www.control.lth.se/media/Education/EngineeringProgram/FRTN20/2016/ZVEI-Industrie-40-RAMI-40-English.pdf

intelligently across company boundaries to make manufacturing more efficient and flexible. At the same time, manufacturing costs can be reduced despite the individualised manufacturing. Networking the companies in the supply chain makes it possible to optimise not only individual production steps, but the entire value chain (Platform Industrie 4.0).

The discovery of new technologies has escorted industry development from the early adoption of mechanical systems, to support production processes, to today's highly automated assembly lines, in order to be responsive and adaptive to current dynamic market requirements and demands. This requires establishing the factory with capabilities of self-awareness, self-prediction, self-comparison, self-reconfiguration, and self-maintenance (Platform Industrie 4.0). Moreover, optimized decision-making is a key characteristic of a smart factory. Taking the right decisions at anytime is a key to succeed in the market (Shrouf et al., 2014). To this end, new improvements and value can be provided by the analysis of large quantities of collected data by IoT devices (i.e. big data).

Figure 2-5: Industry 4.0 Core Elements (Source: BCM Advanced Research)

According to BCG, the potential impact of Industry 4.0 in Germany will be obvious in four main areas (Rüßmann et al, 2015):

- **Productivity**. During the next ten years, Industry 4.0 will be embraced by more companies, boosting productivity across all German manufacturing sectors by €90 billion to €150 billion. Productivity improvements on conversion costs, which exclude the cost of materials, will range from 15 to 25 percent. When the materials costs are factored in, productivity gains of 5 to 8 percent will be achieved. These improvements will vary by industry. Industrial-component manufacturers stand to achieve some of the biggest productivity improvements (20 to 30 percent), for example, and automotive companies can expect increases of 10 to 20 percent.

- **Revenue Growth**. Industry 4.0 will also drive revenue growth. Manufacturers' demand for enhanced equipment and new data applications, as well as consumer demand for a wider variety of increasingly customized products, will drive additional revenue growth of about €30 billion a year, or roughly 1 percent of Germany's GDP.
- **Employment**. The growth it stimulates will lead to a 6 percent increase in employment during the next ten years. And demand for employees in the mechanical-engineering sector may rise even more—by as much as 10 percent during the same period. However, different skills will be required. In the short term, the trend toward greater automation will displace some of the often low-skilled laborers who perform simple, repetitive tasks. At the same time, the growing use of software, connectivity, and analytics will increase the demand for employees with competencies in software development and IT technologies.
- **Investment**. Adapting production processes to incorporate Industry 4.0 will require that German producers invest about €250 billion during the next ten years (about 1 to 1.5 percent of manufacturers' revenues).

2.2.2 Big Data and Internet of Things in Industry 4.0

Intelligence is the key enabler to facilitate work and in a broad sense and consists of two parts. Algorithmic intelligence describes how to reach a goal via a process (e.g., driving your car to a destination) and tactical intelligence describes how to reach the destination with consideration to changing factors (e.g. checking the car tire pressure to compensate for changing road conditions). Industry 4.0 in the simplest form concerns enabling manufacturing with the element of tactical intelligence using techniques and technologies such as IoT, cloud computing and big data (Trappey et al., 2016). IoT is considered to be a paradigm shift for Internet technologies. Estimations show that as of 2014 the number of IoT-enabled devices has exceeded the world's human population. IoT is used by consumers as well as by manufacturers that rely on cyber (software, data systems) and physical (devices, machinery, equipment) connectivity to function effectively (Trappey et al., 2017). While Industry 4.0 was initially considered a technology experiment, it is now a necessity to maintain competitiveness in a constantly changing industry environment. IoT is a core enabling technology that enables industries to move from Industry 3.0 to Industry 4.0 by inserting intelligence into products and processes across the supply chain. Industry 4.0 also represents the aggregation of IoT, CPS, cloud computing and big data analytics to improve the goal of a near zero defect state (Cheng et al., 2016).

Various aspects of IoT technology have been reviewed in the academic realms (Da Xu et al., 2014; Trappey, et al., 2017). IoT can be considered as a global network infrastructure composed of numerous connected devices that rely on sensory, communication, networking, and information processing technologies (Da Xu et al., 2014). So far, IoT has been gaining attraction in industry, in manufacturing enterprises having installed sensors generating real-time big data, such as logistics, manufacturing, retailing, and pharmaceutics. With the advances in wireless communication, smartphone, and sensor network technologies, more and more networked things or smart objects are being involved in IoT. As a result, these IoT-related technologies have also made a large impact on new information and communications technology (ICT) and enterprise systems technologies (Da Xu et al., 2014).

Identifying and structuring an architecture or model is a long process with much negotiation to abstract from specific needs and technologies in order to fulfill the following requirements (Weyrich, and Ebert, 2016):

- Connectivity and communications either one-to-one (unicast) or data collection and information dissemination to multiple partners (multicast and anycast).
- Device management must provide solutions once a device is added or a device configuration changes and must be propagated to other devices.
- Data collection, analysis, and actuation are relevant for extracting information and knowledge for offering services.

- Scalability is important to handle increased processing volumes for different installation sizes.
- Security features are necessary to provide trust and privacy and are required for all aspects of the IoT.

Many market researchers such as Gartner and Cisco consider the industrial IoT as the IoT concept with the highest overall potential, although it has not gathered yet the interest that smart homes or wearables have gathered, due to the high investments required and the long periods of implementation needed. Modern manufacturing companies collect and store operations-related data or even utilize technological infrastructures and information systems for monitoring and detecting early warning signals that machines or systems are degrading or in danger of breakdown. While this is valuable information that can reduce risk of unplanned downtime and potentially save the enterprise money, if organizations improve their analytics maturity, information will be fully harnessed, and the potential is much greater.

This fact depends on the level of maturity a manufacturing enterprise has reached, in terms of data processing capabilities, ICT advancements and maintenance management development. According to Gartner[4] and PwC[5], to seize near-term opportunities, capitalize on the long-term structural shift and accelerate the overall development of the Industrial IoT, technology providers need to focus on brownfield innovation to support existing equipment in the field, and raise the market awareness on successful use cases and implementations, while technology adopters should reorient their overall business strategy to take full advantage of the latest developments in the Industrial IoT by also identifying their new ecosystem partners.

The need for increased **data analytics maturity** has been identified in the industrial and research realms. However, taking advantage of the sensor-generated big data requires the development of data processing systems that are able to handle real-time data in complex, dynamic environments. In this way, the effective increase of data analytics maturity can facilitate predictions and decisions ahead of time,

[4] https://www.gartner.com/doc/3065317/using-advanced-analytics-predict-equipment
[5] http://www.pwc.com/gx/en/industries/industry-4.0.html

leading to increased operational intelligence. According to Gartner[6], there are **four levels of data analytics maturity**, each one building on the previous one: ***Monitor, Diagnose and Control, Manage, Optimize***. In the first level, companies monitor through sensors or other measuring devices and report on asset behaviour. In the second level, on the basis of monitoring, enterprises diagnose issues and sometimes respond to the sensing issue. For instance, monitoring and controlling equipment enables the throttling back or shutting down of an expensive piece of equipment when temperatures or pressures exceed certain thresholds that could lead to downtime. The third level enables organizations to manage the performance of asset and processes by creating predictive models in order to enable shifting from unplanned maintenance to predictive maintenance, resulting in less downtime, better quality and reduced costs. Finally, level 4 is about optimizing decisions, processes and systems on the basis of the real-time predictive models. Organizations can do this to determine the optimal production schedule/product mix or predictive maintenance schedule across assets to optimize asset life and profit, and to minimize downtime while meeting customer demand (business outcome).

2.2.3 Cyber Physical Systems

Cyber-Physical Systems (CPS) is defined as transformative technologies for managing interconnected systems between its physical assets and computational capabilities (Baheti and Gill, 2011). With recent developments that have resulted in higher availability and affordability of sensors, data acquisition systems and computer networks, the competitive nature of today's industry forces more factories to move toward implementing high-tech methodologies and to converge IT and automation as shown in Figure 2-6. Consequently, the ever growing use of sensors and networked machines has resulted in the continuous generation of high volume data (Lee, et al., 2013). In such an environment, CPS can be further developed for managing Big Data and leveraging the interconnectivity of machines to reach the goal of intelligent, resilient and self-adaptable machines (Lee, et al., 2015).

[6] https://www.gartner.com/doc/2826118/industrial-analytics-revolutionizes-big-data

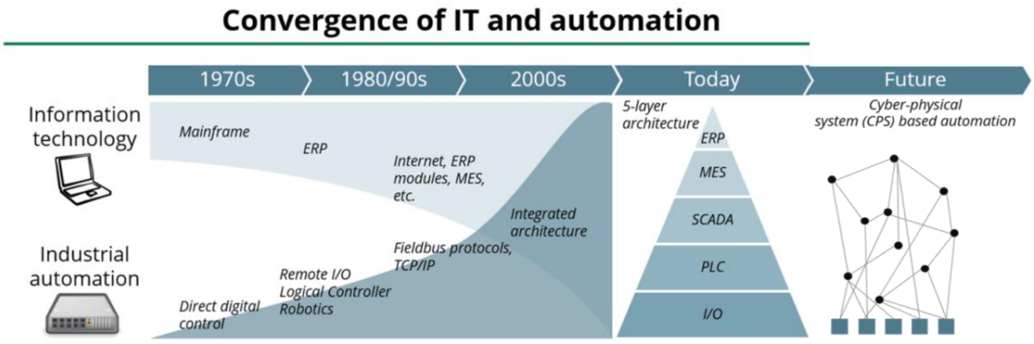

Figure 2-6: Convergence of IT and automation (Source: iot-analytics.com)

The 5C CPS architecture provides a step-by-step guideline for developing and deploying a CPS for manufacturing application. In general, a CPS consists of two main functional components: (1) the advanced connectivity that ensures real-time data acquisition from the physical world and information feedback from the cyber space; nd (2) intelligent data management, analytics and computational capability that constructs the cyber space. However, such requirement is very abstract and not specific enough for implementation purpose in general. In contrast, the 5C architecture clearly defines, through a sequential workflow manner, how to construct a CPS from the initial data acquisition, to analytics, to the final value creation (Lee et al., 2015), as shown in Figure 2-7. Figure 2-8 shows applications and techniques associated with each level of the 5C architecture

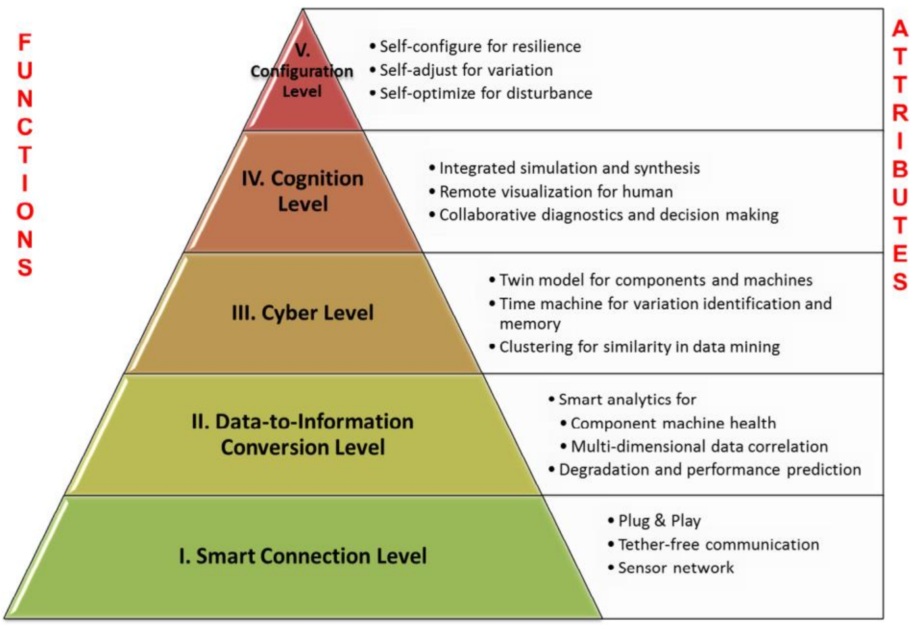

Figure 2-7: 5C architecture for implementation of Cyber-Physical System (Lee et al., 2015).

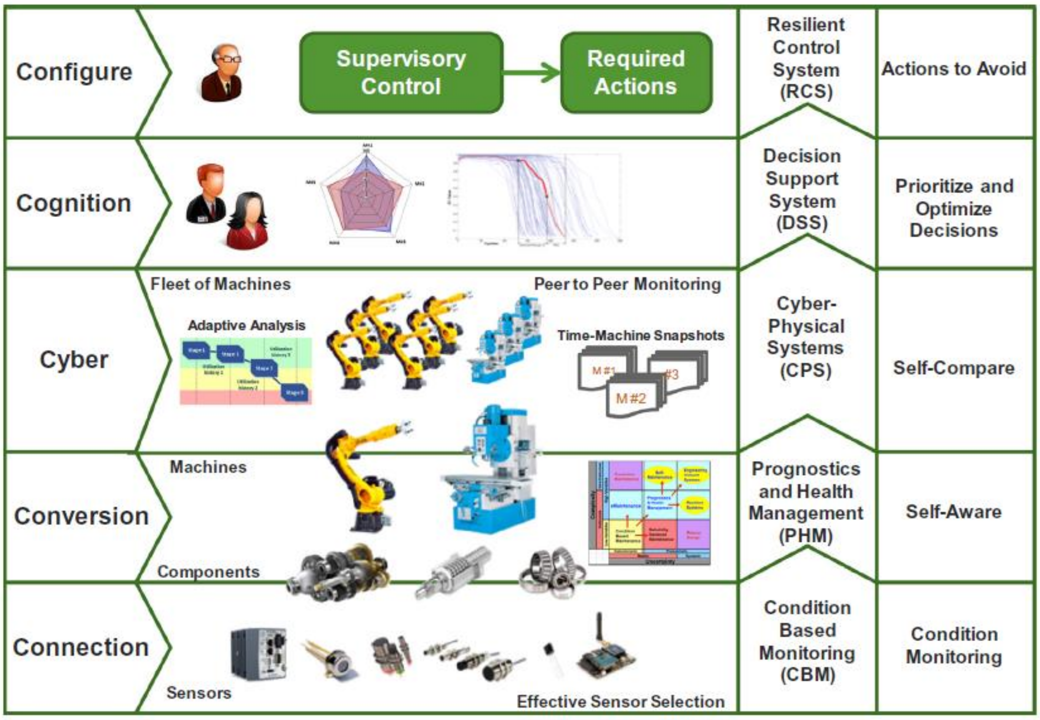

Figure 2-8: Applications and techniques associated with each level of the 5C architecture (Lee et al., 2015).

2.2.4 RAMI 4.0

Authors of the RAMI 4.0 model are BITCOM, VDMA and ZWEI. They decided to develop a 3D model because the model should represent all different manually interconnected features of the technical – economical properties, as shown in Figure 2-9. Very important criterion in the modern engineering is the product life cycle with the value stream which it contains. The left – hand horizontal axis displays this feature. There are expressed e.g. constant data acquisition throughout the life cycle. Even the totally digitization of the whole development – market chain offers great potential for improvement of products, machines, and other layers of the Industry 4.0 architecture throw-out the all life cycle (Zezulka et al., 2016). This look corresponds well with the IEC 62890 draft standard. The next model axis (right in the horizontal level) describes function position of the components in the Industry 4.0. In this axis, there is specified the functionality of the components, no any specification for implementation but the function assignment only. The axis respects both IEC 6224 and the 61512 standards. But the IEC 6224 and the 61512 standards are intended for specification of components in a position in one enterprise or works unit only. Therefore the highest level in the axis horizontal right is the Connected world.

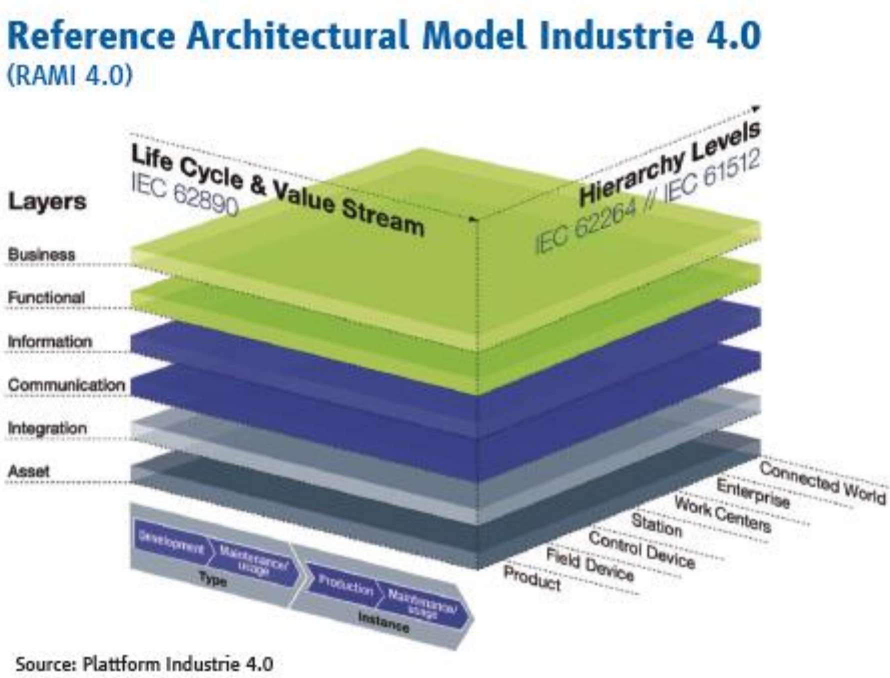

Figure 2-9: The Reference Architectural Model Industrie (RAMI) 4.0 (Source: Platform Industrie 4.0)

The individual layers and their interrelationships, shown in Figure 8, are described as follows (Platform Industrie 4.0, 2014):

Function of layers in vertical axis:

- *Asset Layer:* This layer represents reality, e.g. physical components such as ideas, archives, documents, linear axes, metal parts, diagrams. Also human being is a part of the Asset Layer. They are connected with the virtual reality world by the Integration layer. Passive connection of the assets to the higher Integration Layer is done by for instance means of QR codes.
- *Integration Layer:* This layer makes provision of information on the assets (HW/SW, components) in a form which is available for computer processing. It makes also computer control of the process, generation of events from assets and it contains elements, which are connected with IT (RFID readers, sensors, HMI, actuators, etc.). Integration of persons is a part of Integration layer functions as well – (via HMI).

- *Communication Layer:* This layer provides standardization of communication by means of uniform data format in the direction of the Information Layer. It provides also services for control of the integration Layer.
- *Information Layer:* Provides run time for preprocessing of events, execution of event-related rules. It enables formal description of the rules and event pre – processing. Next functions of the Information layer are: Ensuring data integrity, consistent integration of different data, obtaining new, higher quality data (data, information, knowledge) provision of structured data by means of service interfaces. It also receives events and transforms them to match the data which are available for the higher layer.
- *Functional Layer:* Functional Layer enables formal description of functions and creates platform for horizontal integration of various functions. It contains run time and modeling environment for services for support of business processes and a run time environment for applications and technical functionality. Rules and decision – making logic are generated in the Functional Layer. Some use case can be executed in lower layers as well. But remote access and horizontal integration can take place within the Functional layer only because of the necessity of data integrity.
- *Business Layer:* The layer ensures the integrity of functions in the value stream, enables mapping business models and the resulting of the overall process. It contents legal and regulatory Framework conditions, enables modeling of the rules which the system has to follow. The layer creates also a link among different business processes.

Function of layers in the horizontal left axis:

The left – hand side horizontal axis represents the life cycle & and value stream of industrial production. This axis is divided to Type and Instance. A type of any product, machine or SW/HW represents the initial idea. This covers the placing of design orders, development and testing up to the prototype of production. After all tests and validation, the type is prepared for serial production. On the other hand, the type of any component, machine or HW/SW etc. creates a basis for the serial production. Each manufactured product represents an instance of that type, for example has a unique serial number. The instances are sold and delivered to customers. For customers are the products initially once again only types. They become instances when they are installed in a particular system. The change from type to instance may be repeated many times. The fine structure of the life cycle and value stream look in the RAMI 4.0 over the axis left hand horizontal shows a division of the Type to Development and Maintenance/ usage, but due to the physical character of the problem – instances consist from Production and Maintenance/usage. The function of layers in the horizontal left axis can be explained in following simple example: The development of a new electrical drive represents creation of a new type of an engine. The drive (controlled engine) is developed, initial samples are set up and tested and a first prototype series is manufactured and validate. After successful testing, the new drive type is released for sale (product designation in sales catalogue of the producer). In this moment a first serial production can be started. Each drive in the serial production has its serial number (a unique identification) and is an instance of the previously developed electric drive. Feedback from customers to instances of the type may lead to corrections in the mechanical part of the drive and correction in the control SW. Such modifications are modification in the type, i.e. they are applied as amendments to the type documentation and new instances of the modified type are produced. The left hand side of the RAMI 4.0 model represents the value stream as well.

Digitization and linking of the value stream (in the Industry 4.0 idea and praxis) big potential for improvement of produced types. Logistic data can be used in assembly, purchasing sees inventories in real time and know were parts from suppliers are at any moment, customers sees the completion status of the product during production etc. The value stream in the totally digitized production enables linking of purchasing, order planning, assembly, logistic, maintenance, the customer and suppliers and so on. It provides great improvement potential .The life cycle can therefore be viewed together with the value- adding processes which it contains and not in isolation as it is in the present production (Platform Industrie 4.0. (2014)).

2.2.5 Industry 4.0 Component Model

The second very important model for purposes of the Industry 4.0 that has been developed by BITCOM, VDMA and ZWEI during the last one year is the Industry 4.0 components model. It is intended to help producers and system integrators to create HW and SW components for the Industry 4.0. It is the first and the only specific model which goes out from the RAMI 4.0 model. It enables better description of cyber – physical features and enables description of communication among virtual and cyber – physical objects and processes (Zezulka et al., 2016). The HW and SW components of future production will be able to fulfil requested tasks by means of implemented features specified in the Industry 4.0 components model. The most important feature is the communication ability among the virtual objects and processes with real object and processes of production while this model specifies the conform communication. Physical realization of it is that any component of the Industry 4.0 system takes an electronic container (shell) of secured data during the all life cycle. The data are available to all entities of the technical – production chain. Therefore this model goes out from a standardized, secure and safety real time communication of all components of production. The electronic container (shell) of data and the all Industry 4.0 component model is specified in Figure 2-10 (Adolphs et al., 2015).

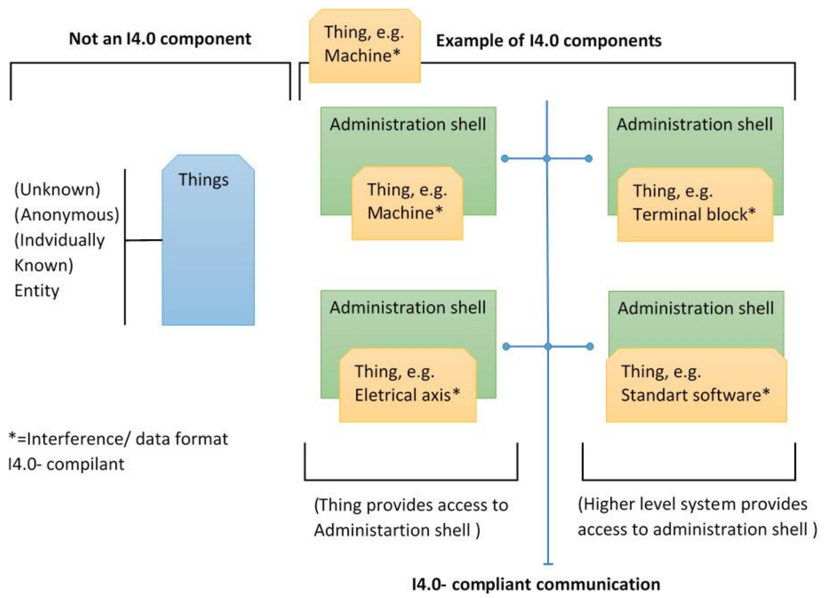

Figure 2-10: The I4.0 component model (Source: Platform Industrie 4.0)

2.3 Maintenance Management

2.3.1 Overview

Over the last few decades, maintenance functions have drastically evolved with the growth of technology (Ahmad, and Kamaruddin, 2012). Maintenance is defined as a set of activities or tasks used to restore an item to a state in which it can perform its designated functions (Duffuaa, et al., 1999; Dhillon, 2017). The modern industry is increasingly demanded to work at high reliability, low environmental risks, and human safety while operating their processes at maximum yield (Peng et al., 2010). Industrial maintenance, is gaining significance (Cannata et al., 2010; Ruschel et al., 2017) both within the academic and industrial community, as it develops from being considered a minor activity, towards a strategic task in operation management (Pinjala et al., 2006), thus being called asset lifecycle management.

Technological development has resulted in increased complexity in both industrial machinery and production systems. The economical consequences from an unexpected 1-day stoppage in industry may become as high as up to 100,000 to 200,000 euros (Peng et al., 2010). Operational reliability of industrial machinery and production systems has a significant influence on the profitability and competitiveness of industrial companies. This emphasizes the increasing importance of effective maintenance strategies of machinery, production processes, and systems in industry (Peng et al., 2010). Maintenance strategies can be broadly classified into Breakdown Maintenance (BM), Time-Based Maintenance (TBM) and Condition-Based Maintenance (CBM) (Duffuaa, et al., 2001). Other terms such as “preventive maintenance”, “planned maintenance”, “predictive maintenance” are also used.

Breakdown Maintenance, also known as run-to-failure, corrective or reactive maintenance, is a strategy that is used to restore (repair or replace) some equipment to its required function after it has failed (Blanchard, et al., 1995) implementing corrective actions. This strategy leads to high levels of machine downtime (production loss) and maintenance (repair or replacement) costs due to sudden failure (Tsang, 1995).

Time-Based Maintenance, also known as **Preventive Maintenance** or planned maintenance, involves the performance of a set of certain maintenance activities prior to the failure of equipment in specific time intervals (Lofsten, 1999). It has replaced the Breakdown Maintenance and is still widely used in manufacturing firms. This strategy contributes to minimizing failure costs and machine downtime (production loss), and increasing product quality (Usher, et al., 1998). In the industry, application of the TBM strategy can be generally performed through either experience or original equipment manufacturer (OEM) guidelines and it is performed at regular time intervals (Sheu, et al., 1995). TBM will also encounter some minor or major planned shutdowns of systems for predetermined overhaul or repair activities on still functioning equipment. System overhaul and critical item replacement at fixed intervals are widely adopted in automated manufacturing and control systems. Although TBM can reduce the probability of system failures and the frequency of unplanned emergency repairs, its intervals based on OEM recommendations may not be optimal because actual operating conditions may be very different from those considered by the OEM (Labib, 2004; Tam, et al., 2006). On the economy aspect, TBM tends to be too conservative that results in very high maintenance costs.

TBM is widely used in industry; however, companies are increasingly turning to CBM, with manufacturing companies considering the use of condition monitoring. CBM is becoming essential for every manufacturing business as products have become more and more complex due to the evolution of technology and thus, quality and reliability have become issues of high significance (Jardine et al., 2006; Peng et al., 2010; Hashemian, and Bean, 2011). Consequently, the costs of time-based preventive maintenance have increased and CBM has started to be evolved as a novel lever for maintenance management (Jardine et al., 2006; Guillen et al., 2016).

Condition-Based Maintenance (CBM) is a maintenance strategy where the decision to perform maintenance is reached by observing the “condition” of the system and its components (Guillen et al., 2016). CBM attempts to avoid unnecessary maintenance tasks by taking maintenance actions only when there is evidence of abnormal behaviors of a physical asset (Jardine et al., 2006). Generally, CBM can be treated as a method used to reduce the uncertainty of maintenance activities and is carried out according to the requirements indicated by the equipment condition (de Jonge et al., 2017).

Condition monitoring has been significantly enabled by the development of appropriate technologies and sensing equipment measuring various parameters. In this way, the engineers are able to monitor in real-time the actual health state of equipment and to decide about maintenance actions. Condition monitoring is increasingly realized with equipment-installed sensors, which have the capability of measuring with high frequency a multitude of parameters (Jardine et al., 2006; ISO 2012a; ISO 2012b) leading to processing and storage of a huge amount of data (big data) that pose challenges to the subsequent processing pipeline of data analysis, knowledge extraction and decision making.

Predictive Maintenance goes a step beyond the mere real-time monitoring of the manufacturing system. It indicates the use of detection and prediction algorithms about the current and the future health state of the manufacturing system respectively with the use of Prognostics and Health Management (PHM) methods and techniques. In this way, maintenance decision making is facilitated. The concept of Predictive Maintenance evolved almost in parallel with the concept of CBM, however with a different meaning. At the beginning, Predictive Maintenance did not consider condition monitoring, but it dealt with predictions based on expert knowledge and manufacturer's specifications of equipment. Even today, there are several research works dealing with Predictive Maintenance without considering sensor-generated real-time data.

The classical industrial view of CBM and predictive maintenance is mainly focused on the use of condition monitoring techniques such as vibration analysis, thermography, acoustic emission or tribology (ISO 2011). The recent developments of maintenance management lead to a new predictive maintenance approach, providing powerful capabilities for physical understanding of the useful life of a system through dynamic pattern recognition in various available data sources, Remaining Useful Life (RUL) or Remaining Life Distribution (RLD) prediction and providing maintenance-related recommendations in order to exploit the full potential of the predictive maintenance framework and the advances in ICT.

2.3.2 Condition Based Maintenance and Predictive Maintenance

Sometimes predictive maintenance is used as an alternative term of CBM. Other terms that are used in literature are “online monitoring”, “risk-based maintenance” (Hashemian, and Bean, 2011) and Prognostics and Health Management (PHM) (Sheppard et al., 2008; Lee et al., 2014; Guillen et al., 2016).

CBM relies on diagnostic and prognostic models and uses them to support decisions about the appropriate maintenance actions based on the current health state of a system and/or its predicted performance and remaining lifetime. It is performed after one or more indicators show that equipment is going to fail or that equipment performance is deteriorating. CBM was introduced to try to maintain the correct equipment at the right time and is based on using real-time data to prioritize and optimize maintenance resources (Jardine et al., 2006; Peng et al., 2010; Voisin et al., 2010; Guillen et al., 2016).

The term “Predictive maintenance” focuses on techniques that help determine the condition of in-service equipment in order to predict when maintenance should be performed. This approach offers cost savings over routine or time-based preventive maintenance, because tasks are performed only when warranted. In most cases, the term “predictive maintenance” does not necessarily include real-time condition monitoring through sensors, while the term “Condition Based Maintenance” does not necessarily include predictions (Liu et al., 2016; Nguyen et al., 2017), since it may refer to (near) real-time diagnostic outcomes, i.e. detection of the current condition, and actions upon them (Garcia et al., 2006; Lindström et al., 2017).

CBM has a long history. From visual inspection, which is the oldest method yet still one of the most powerful and widely used, it has evolved to automated methods that use advanced signal processing techniques based on pattern recognition, including neural networks, fuzzy logic, and data-driven empirical and physical modeling (Hashemian, and Bean, 2011; Guillen et al., 2016). However, nowadays, nearly 30%

of industrial equipment does not benefit from predictive maintenance technologies (PwC, 2017). Predictive maintenance is the preferred maintenance method in 89% of cases, compared to time-based maintenance, which is prudent in only 11% of cases (Hashemian, and Bean, 2011).

Several maintenance frameworks have been proposed in the literature outlining the steps involved in performing CBM. Lee et al. (2004) describes three core steps: (i) data acquisition, to collect the data; (ii) data processing, to handle the data; and (iii) maintenance decision making, to decide about the optimal maintenance policy. Peng et al. (2010) focused on the third step (maintenance decision making), further detailing it into diagnosis and prognosis. The authors also indicated the need for historical data and for the development of a model for representing system behavior. Irigaray et al. (2009) focused on supporting CBM by storing relevant data and information and utilizing them so that the most appropriate decisions are drawn and are updated dynamically by means of a platform based on web services and a systematic process consisting of four layers: condition monitoring, assessment of the health state, prognosis and decision making.

Peng et al. (2010) described in detail a maintenance decision support framework consisting of five main steps: (i) feature selection, which is conducted with the aid of historical data as well as several methods such as Principal Component Analysis, Genetic Algorithms and Support Vector Machine (SVM); (ii) data training (analysis); (iii) diagnostics and prognostics, by using real-time data; (iv) reliability and Remaining Useful Life (RUL) where the result is verified and its precision is assessed in order to give feedback to steps (ii) and (v); and (v) maintenance schedule, which considers the cost function which is extracted from the relationship between the maintenance cost, RUL and reliability of the system.

A generic conceptual framework for CBM decision support has been proposed by Voisin et al. (2010). This framework considers the interactions of prognosis with the whole business environment and represents the business processes which are integrated with prognosis. A simplified version is shown in Figure 2-11 (Iung et al., 2009; Voisin et al., 2010). Moreover, it separates the decision support step from diagnostics and prognostics by combining and updating two earlier frameworks (Léger and Morel, 2001; Muller et al., 2008a; Lebold and Thurston, 2001). Diagnosis and Prognosis in sensor-driven environments are well studied areas in literature; however, decision making in this context is still underexplored area.

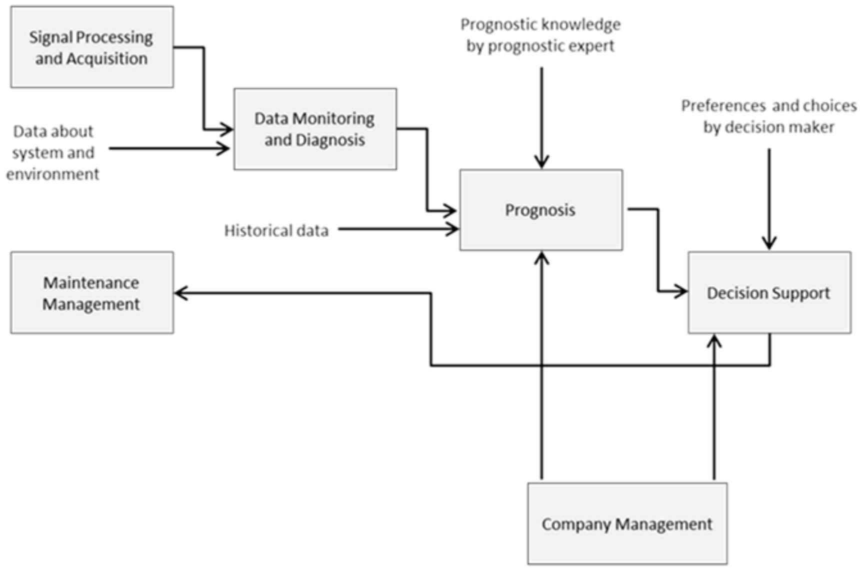

Figure 2-11: The role of diagnosis and prognosis in CBM (adapted from Voisin et al., 2010)

The **MIMOSA OSA-CBM** specification[7] is a standard architecture for moving information in a condition-based maintenance system. It has already been implemented in several industries, such as aerospace industry within the framework of Integrated Vehicle Health Management (IVHM) (Lebold and Thurston, 2001; Dunsdon and Harrington, 2008; Benedettini et al., 2009). A more in depth look reveals a way to reduce costs, improve interoperability, increase competition, incorporate design changes, and further cooperation in the realm of condition-based maintenance. The OSA-CBM provides a means to integrate many disparate components and eases the process by specifying the inputs and outputs between the components. It is an implementation of the ISO-13374 functional specification. OSA-CBM adds data structures and defines interface methods for the functionality blocks defined by the ISO standard. According to ISO-13374, the **six blocks of functionality** are: ***Data Acquisition, Data Manipulation, State Detection, Health Assessment, Prognostics Assessment, Advisory Generation***.

[7] http://www.mimosa.org/mimosa-osa-cbm

Data Acquisition deals with converting an output from the transducer to a digital parameter representing a physical quantity and related information. Data Manipulation performs signal analysis, computes meaningful descriptors, and derives virtual sensor readings from the raw measurements. State Detection facilitates the creation and maintenance of normal baseline “profiles”, searches for abnormalities whenever new data are acquired, and determines in which abnormality zone, if any, the data belong (e.g. “alert” or “alarm”). The final three blocks normally attempt to combine monitoring technologies in order to assess the current health of the machine, predict future failures, and provide recommended action steps to operations and maintenance personnel. More specifically: Health Assessment diagnoses any faults and rates the current health of the equipment or process, considering all state information. Prognostic Assessment determines future health states and failure modes based on the current health assessment and projected usage loads on the equipment and/or process, as well as remaining useful life predictions. Finally, Advisory Generation provides actionable information regarding maintenance or operational changes required to optimize the life of the process and/or equipment.

Guillen et al. (2016) studied CBM with the aim to provide a framework bringing together the managerial and the technical perspective based on international standards. This framework introduces three complementary points of views of this same process simultaneously: (i) CBM basic concepts (detection, diagnosis, prognosis) within the basic CBM flow. These concepts are reinterpreted using two views: (ii) The Data-processing view: CBM flow and concepts reinterpretation within the Data-Processing technical requirements. (iii) The Maintenance information view: maintenance requirements translation.

In maintenance, Industry 4.0 find its application in designing of self-learning and smart system that helps predict failures, diagnose and trigger maintenance schedules (Kumar, and Galar, 2018). In order to extract specific and relevant information, these smart systems are highly demanded for data access, for quality and also for the use of multiple sources of data (Lee et al., 2013; Lee et al., 2015). Development of intelligent maintenance systems based on cyber-physical approach, for failure detection, providing diagnostics and prognostics, has been the core focus on several research projects (Syed et al., 2012, Sankavaram et al., 2013; Kroll et al., 2014).

PwC proposes the Predictive Maintenance 4.0 concept, i.e. predictive maintenance in the frame of Industry 4.0 (PwC, 2017). PwC defined the application of big data analytics in maintenance as the fourth level of maturity in predictive maintenance, namely Predictive Maintenance 4.0 (PwC, 2017). Based on their definition, Predictive Maintenance 4.0 is about predicting future failures in assets and ultimately prescribing the most effective preventive measure by applying advanced analytic techniques on big data about technical condition, usage, environment, maintenance history, similar equipment elsewhere and in fact anything that may correlate with the performance of an asset. The four levels are described below:

- Level 1 Visual inspections: periodic physical inspections; conclusions are based solely on inspector's expertise.
- Level 2 Instrument inspections: periodic inspections; conclusions are based on a combination of inspector's expertise and instrument read-outs.
- Level 3 Real-time condition monitoring: continuous real-time monitoring of assets, with alerts given based on pre-established rules or critical levels.
- Level 4 PdM 4.0: continuous real-time monitoring of assets, with alerts sent based on predictive techniques, such as regression analysis.

2.3.3 Decision Making in Predictive Maintenance

2.3.3.1 The Role of Decision Making in Predictive Maintenance

Decision making in predictive maintenance indicates the phase which is triggered by (near) real-time predictions (e.g. about a future failure) in order to generate proactive recommendations about maintenance actions that eliminate or mitigate the impact of the predicted failure. In addition, decision making incorporates domain knowledge related to maintenance management of the specific industry.

Although there is a rich literature on diagnostic and prognostic models, automated decision making in the context of predictive maintenance is an underexplored area. The evolution of Internet of Things (IoT) and the emergence of Industry 4.0 pave the way for an extensive use of sensors in the manufacturing environment measuring a multitude of parameters. Efficient processing of all this data and providing meaningful business insights is of outmost importance. To this end, the level of big data analytics maturity can be increased by generating recommendations ahead of time on the basis of (near) real-time predictions. In this way, manufacturing firms can optimize their performance and obtain a significant competitive advantage. Currently, there are many conceptual papers regarding Industry 4.0, but decision making for predictive maintenance in this context has not been examined yet.

The current literature review focuses on decision making algorithms for predictive maintenance. In this sense, it investigates decision making algorithms that are triggered by predictions that have been derived through condition monitoring. Condition monitoring is the process of monitoring the condition in order to identify a significant change which is indicative of a developing fault (Han, and Song, 2003). It is a major component of predictive maintenance (Márquez et al., 2012). During the last years, due to the emergence of Industry 4.0 and IoT, condition monitoring techniques have evolved from visual inspections and manual analysis of data sets to high-frequency sensors generating real-time big data about several parameters (e.g. vibration, temperature, thermography, etc.). On the basis of this data, advanced data analytics techniques can be applied in order to handle the uncertainty due to the stochastic manufacturing operations.

Since the dynamicity and complexity of the manufacturing environment make decision making a challenging task, there is an increasing interest on maintenance decision making algorithms. However, existing literature reviews on maintenance decision making algorithms have usually the following limitations: (i) they do not distinguish between static and dynamic models (through condition monitoring), e.g. between offline and real-time models; (ii) they get involved with various maintenance strategies without focusing on predictive maintenance; (iii) Decision making is not necessarily executed on the basis of predictions; (iv) They focus on specific categories of decision making methods (e.g. optimization) and/ or maintenance aspects (e.g. maintenance policy).

An important and well-established principle of predictive maintenance is the P-F curve, which is shown in Figure 2-12. P-F curve indicates how a part of equipment starts being degraded to the point at which the forthcoming failure can be predicted (the potential failure point "P"). Thereafter, if it is not predicted and no suitable action is taken, it continues to deteriorate - usually at an accelerating rate - until it reaches the point of functional failure (Point "F"). The amount of time which elapses between the point where a potential failure occurs and the point where it deteriorates into a functional failure is known as the P-F interval (Veldman et al., 2011). This interval can be seen as an opportunity window during which actions can be taken with the aim to eliminate the anticipated functional failure or mitigate its effect. In an Industry 4.0 context, decision making algorithms can be triggered by real-time predictions within this interval in order to generate proactive recommendations.

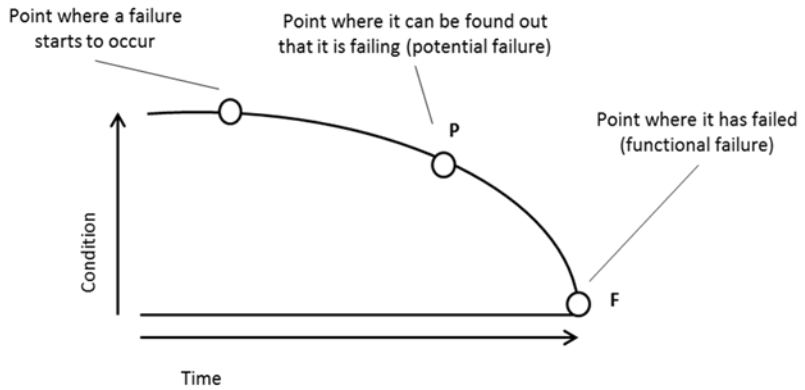

Figure 2-12: P-F curve

Such decision making algorithms should take into account several constraints and objectives and provide the best maintenance plan, i.e. the one that minimizes the maintenance costs and optimizes overall business performance. Figure 2-13 depicts the relationship among the predicted time-to-failure, the equipment reliability and the maintenance costs. It shows that while time-to-failure is approaching zero, reliability is decreasing (Peng et al., 2010). When time-to-failure becomes zero, a failure (e.g. breakdown of equipment) occurs. The best time to do maintenance is when the maintenance cost is minimum and reliability has started to decrease significantly.

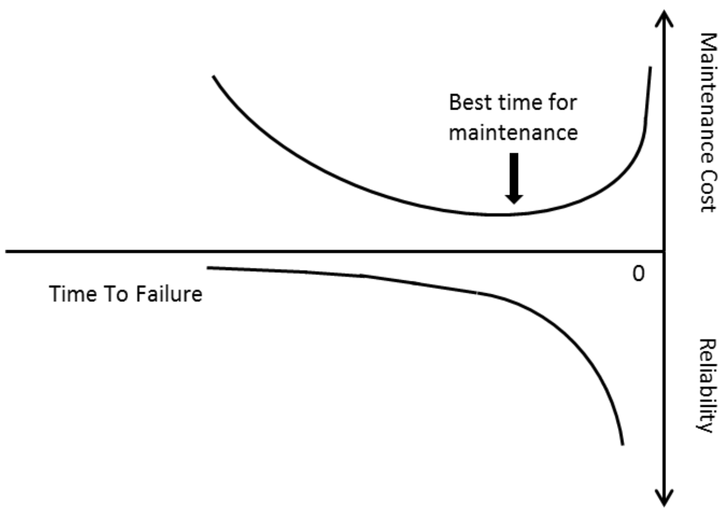

Figure 2-13: Relationship among time-to-failure, reliability and maintenance cost

The increasing complexity and uncertainty of the manufacturing environment has leveraged the emergence of several algorithms aiming to better support decision making, e.g. for maintenance planning (Ruschel et al., 2017). Effective and automated (i.e. by providing recommendations) decision making in predictive maintenance leads to higher reliability of equipment and reduced losses (Ruschel et al., 2017). Smart decision making is at the heart of Industry 4.0 since the ultimate goal of deploying widespread sensors is to achieve smart decision making through comprehensive data collection (Zheng et al., 2018). However, the uncertainty existing in predictive analytics but also in the degradation process itself and the time constraints under which a decision should be taken pose significant challenges in the applicability of the decision making algorithms. Such decision making algorithms should be able to provide courses of actions with the aim to improve equipment operating life at maximised performance. During the last years, with the emergence of Predictive Maintenance as a novel lever of maintenance management, there is an increasing interest in decision making algorithms aiming to better support maintenance decisions.

2.3.3.2 Decision Making Algorithms in Literature

This Section investigates decision making algorithms for predictive maintenance in literature. In order to facilitate the comprehension and the investigation of the existing algorithms, we structured the literature in 5 areas of decision making problems in predictive maintenance based on similar classifications existing in literature (e.g. Alaswad, and Xiang, 2017; Ruschel et al., 2017; Chemweno et al., 2018). Most of papers belong to more than one area. However, the categorization to these 5 areas was formulated based on the focus as well as the main contribution and novelty of each work to the specific area. The 5 areas of decision making algorithms in predictive maintenance along with the associated references are shown in Table 2-1. These areas are the following:

- **Maintenance Planning and Scheduling**: This area includes algorithms that enable defining the maintenance actions based on the adopted policies and the information about the impacts and risks.
- **Reliability- and Degradation- based Decision Making**: This area includes algorithms incorporating the degradation rates in order to minimize long-run costs, and thus, to enable the scheduling of mitigating maintenance actions. They may also utilize information from the equipment conditions, in order to support the modelling and balancing between costs and reliability objectives, e.g. using probabilistic methods.
- **Joint Optimization**: This area includes algorithms aiming to optimize maintenance results, considering the objectives of the production system in order to ensure overall business improvements. Optimized maintenance cost reduction does not always lead to optimization of equipment availability, which can lead to delays in the production and delivery of the final product. To this end, there are algorithms achieving the balance between maintenance with production, logistics and quality objectives.
- **Multi-State and Multi-Component Systems Optimization**: This area includes algorithms that allow the identification of intermediate stages of their condition. Therefore, on the basis of this, optimization models lead to intermediate decision making. Although there is a high amount of algorithms in the literature, we selected the ones that are applied in a more explicit and direct way.

- **Maintenance Cost and Risk Estimation and Optimization**: This area includes algorithms dealing with cost issues in order to facilitate the decision making of maintenance actions with the optimal cost. Sometimes, there is the possibility of estimating and calculating maintenance costs for different situations and scenarios. There are also algorithms that map the impact of failures with financial issues. Such algorithms also provide a ranking of the critical components of the system along with a priority of suitable maintenance actions.

Table 2-1: Areas of decision making algorithms in predictive maintenance.

Area of Contribution	Number of references	References
Maintenance Planning and Scheduling	23	Yu et al., 2003; Carnero, 2006; Muller et al., 2007; Su, and Tsai, 2010; Li, and Gao, 2010; Martorell et al., 2010; Varnier, C., and Zerhouni, 2012; Al-Najjar, and Jacobsson, 2013; Xu et al., 2013; Duarte et al., 2013; Guo et al., 2013; Mendes et al., 2014; de Jonge et al., 2015; Gopalakrishnan et al., 2015; Xu et al., 2015; Terkaj et al., 2015; Wan et al., 2015; Nadj et al., 2016; Yildirim et al., 2016a; Yildirim et al., 2016a; Said et al., 2016; Fitouri et al., 2016; Ghosh et al., 2017
Reliability- and Degradation- based Decision Making	24	Sun et al., 2007; Wu et al., 2007; Elwany, and Gebraeel, 2008; Muller et al., 2008; Xu, and Hu, 2008; Islam, and Khan, 2010; Zhu et al., 2010; Besnard, and Bertling, 2010; Zhu et al., 2011; Tian et al., 2012; Castro et al., 2012; Moghaddass et al., 2014; Le et al., 2014; Hong et al., 2014; Song et al., 2014; Tang et al., 2015a; Tang et al., 2015b; Do et al., 2015; Lin et al., 2015; Park et al., 2016; Drumheller et al., 2017; He et al., 2017; Animah, and Shafiee, 2017; Zan et al., 2018
Joint Optimization	16	Zhou et al., 2007; Njike et al., 2009; Rausch, and Liao, 2010; Gulledge et al., 2010; Nodem et al., 2011; Wang, 2011; Portioli-Staudacher, and Tantardini, 2012; Lee, and Ni, 2013; Kouedeu et al., 2015; Gan et al., 2015; Jafari, and Makis, 2015; Jiang et al., 2015; Van Horenbeek, and Pintelon, 2015; Cinus et al., 2016; Gu et al., 2017
Multi-State and Multi-Component Systems Optimization	11	Le, and Tan, 2013; Zhou et al., 2013; Xia et al., 2013; Van Horenbeek, and Pintelon, 2013; Sheu et al., 2015; Azadeh et al., 2015; Jiang et al., 2015b; Huynh et al., 2015; Nguyen et al., 2015; Li et al., 2016; Keizer et al., 2016
Maintenance Cost and Risk Estimation and Optimization	23	Fouladirad et al., 2008; van der Weide et al., 2010; Nordgård et al., 2010; Sharma, and Sharma, 2010; Vaurio, 2011; van der Weide et al., 2011; Fouladirad et al., 2014; Cheng et al., 2012; Dandotiya, and Lundberg, 2012; Emde, and Boysen, 2012; Sharma, and Sharma, 2012; Sinkkonen et al., 2013; Faccio et al., 2014; Susto et al., 2014; Haroun, 2015; Susto et al., 2015; Wu et al., 2015; Wang, 2016; Chen, and Kezunovic, 2016; Bumblauskas et al., 2017; Li et al., 2017; Si et al., 2017

In the papers investigated, the decision making algorithms are based on the following categories of methods:

- **k-out-of-n structure** refers to a n-components system that works if at least k of the n components work, or fail if at least k of the n components fail.
- **Dempster-Shafer** is a mathematical theory of evidence that allows combining evidence from different sources and represents them by a reliability function.
- **Genetic Algorithms** are metaheuristic algorithms inspired by the process of natural selection. Genetic algorithms are commonly used to generate high-quality solutions to optimization, multi-objective optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection.
- **Simulated annealing** is a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a metaheuristic to approximate global optimization in a large search space. It is often used when the search space is discrete.
- **Probabilistic relational model** is a representation language for statistical models that combine a frame-based logical representation with probabilistic semantics such as Bayesian networks.
- **Markovian processes** (Markov chains, Markov Decision Process, Semi-Markov Decision Process, Partially Observable Markov Decision Process) is a stochastic process with discrete states in which the probability distribution of the next state depends only on the current state and not on the sequence of events that preceded.
- **Case-based Reasoning** is a technique that seeks to solve new problems by adapting solutions used to solve previous problems.
- **Logical Analysis of Data** is a data analysis methodology that integrates Boolean functions and optimization concepts.
- **Probabilistic Safety Assessment** is a technique for numerically quantifying risk measures.
- **Fuzzy Logic and Inference** is the actual process of mapping from a given input to an output using fuzzy logic.

- **Collaborative Planning** enables the management of information and knowledge to support maintenance decision making.
- **Rule-based Systems** (e.g. Event-Condition-Action rules) are an approach that incorporates the domain knowledge in an expert system. It may incorporate probabilities or fuzzy sets.
- **Mathematical Programming/ Optimization** (Linear, Non-linear, Stochastic Dynamic, Mixed Integer Optimization) aims to deal with optimization problems formulated in a respective objective function.
- **Multi-objective Optimization** deals with mathematical optimization problems involving more than one objective function to be optimized simultaneously.
- **Regression Analysis** is a set of statistical processes for estimating the relationships among variables. It includes many techniques for modelling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables.
- **Cost Risk Analysis** involves the trade-off between the cost risk of a failure and the cost risk of a maintenance action and is usually based on the predicted reliability distribution.
- **Proportional Hazards Models** are a class of survival models in statistics. Survival models relate the time that passes before some event occurs to one or more covariates that may be associated with that quantity of time. In a proportional hazards model, the unique effect of a unit increase in a covariate is multiplicative with respect to the hazard rate.
- **Bayesian Networks** is a probabilistic graphical model (a type of statistical model) that represents a set of variables and their conditional dependencies via a directed acyclic graph.
- **Statistical Process Control** is a method of quality control which employs statistical methods to monitor and control a process. This helps ensure the process operates efficiently, producing more specification-conforming product with less waste (rework or scrap).

- **Collaborative Filtering** is a method originated from recommended systems aiming to enable decision making in order to support the user based on the identified requirements.
- **Artificial Neural Networks** are systems that learn (i.e. progressively improve performance on) tasks by considering examples, generally without task-specific programming.

2.3.3.3 Discussion and Limitations

Despite the high amount of works regarding decision making algorithms for predictive maintenance, there are still several aspects that concern both industry and academia. The literature review reveals a large gap between academic approaches and industrial applications. Despite the high amount of related research works in literature, there are not many research and industrial platforms incorporating decision making algorithms. It seems that it is difficult for manufacturing companies to deploy and adapt the decision making algorithms existing in literature to their own specific business context, data types and proprietary models. To this end, the need for shifting from theoretically-based research to applied-based research has been outlined (Ruschel et al., 2017).

From the literature review, it seems that predictive maintenance capabilities have not been sufficiently examined in the context of Industry 4.0 and big data technologies. By means of a collaborative environment, pertinent knowledge and intelligence become available at the right place and time, in order to facilitate reaching the best maintenance decisions. However, the formalization of knowledge, information as well as preferences of the decision maker added to the high amount of data generated by sensing equipment concerns both academia and industry (Ruschel et al., 2017). This fact reveals that there are still difficulties in handling and analysing this multitude of data in an efficient and meaningful way.

Consequently, manufacturing companies are still reluctant to adopt novel technologies and information systems to a large extent for improving their maintenance operations. This fact has led technology providers to narrow their development efforts to real-time condition monitoring software or, less often, to diagnostic and prognostic algorithms for specific use cases. In this way, they do not provide full exploitation of the large amounts of data generated by sensors in the direction of developing algorithms, methods and techniques for (near) real-time decision making.

Although there are lots of research works regarding decision making algorithms for predictive maintenance, they are usually limited to specific problems, domains and industries requiring **Maintenance Planning and Scheduling** and are applicable under several assumptions. Consequently, their applicability in real industrial environments is limited, while they cannot be transferred to a different production process with similar challenges in a straightforward way.

A large amount of works deals with optimization of inspection intervals according to the actual reliability of the equipment. In this way, the aim is to conduct **Reliability- and Degradation- based Decision Making** for the definition of the inspection intervals. Although this is useful information, it does not exploit the availability of historical and real-time data and the information that can be extracted in order to recommend specific actions that should be applied by the engineers and the operators in order to significantly facilitate **Maintenance Planning and Scheduling**.

Moreover, rarely the **Reliability- and Degradation- based Decision Making** algorithms take into account real-time prognostic information (e.g. prediction about a future failure) for generating (near) real-time recommendations. There is a loose integration between predictive analytics and decision making algorithms. The common practice is to utilize the current level of degradation that is derived from the analysis of the indicators measured by sensors along with expert knowledge. In addition, they rely on processing of batches of data at specific sampling times. To this end, there is the need for scalable and efficient (near) real-time decision making algorithms. This aspect has both a technological (use of appropriate technologies, e.g. for streaming data) and a functional (use of appropriate decision models, e.g. recursive and computationally efficient) perspective.

Moreover, several decision making algorithms for predictive maintenance are based upon model-based prognostic algorithms instead of data-driven. Therefore, the associated decision making algorithms are mainly knowledge-based due to the lack of data analytics exploitation. The stochastic nature of the degradation process makes decision making for predictive maintenance highly uncertain and complex. For this reason, a large amount of existing decision making algorithms utilize simulation models or iterative solution procedures. Only the simple models get usually involved with exact solutions (e.g. single machine). Moreover, sometimes simulation is combined with more advanced optimization techniques (e.g. genetic algorithms, simulated annealing) in order to decrease the computational effort and provide more reliable results.

There is a gap in literature regarding generic decision models representing the decision making process instead of the physical process. Moreover, there is a gap regarding the use of probabilistic methods in a streaming context with the aim to tackle with uncertainty. However, there is a clear trend in literature, currently mainly at a conceptual level, towards less human intervention in decision making by conducting advanced analytics for big data with self-learning capabilities by observing the largest number of data and information extracted directly from the machinery, in relation to the information based on the expert judgment.

Several algorithms aim to reduce maintenance costs using **Maintenance Cost and Risk Estimation and Optimization** approaches, however without taking into account other parameters such as the availability of the equipment, the total cost of production, the available inventory of maintenance spare parts, the transportation costs, the quality defects, the safety of operations, etc. In this way, it is possible to achieve the costs reduction in the maintenance actions, but negatively impacting on other objectives. This fact has led to the emergence of **Joint Optimization** approaches aiming to improve the overall business performance or at least optimize certain objectives apart from maintenance costs. For example, the downtime of the system might be influenced by logistical delays and a reduction in inventory cost can have an indirect benefit to predictive maintenance operations.

More sophisticated decision making algorithms have also been developed in order to represent the actual manufacturing systems, i.e. **Multi-State and Multi-Component Systems Optimization**. These research works are far less than the ones regarding single component systems; however, the decision making algorithms for single-component systems cannot be properly applied to multi-component systems. Multi-component systems aim to consider various categories of dependencies among components. However, the fact that they are specific to the equipment or to the manufacturing process as well as their increased complexity pose challenges in their implementation in the context of a sensor-driven manufacturing environment.

The majority of the **Maintenance Planning and Scheduling, Joint Optimization** and **Maintenance Cost and Risk Estimation and Optimization** algorithms rely on the assumption of perfect maintenance or replacement, without considering various degrees of maintenance, e.g. recommending maintenance actions with different cost functions and impacts on equipment lifetime.

In the dynamic, sensor-driven manufacturing environment of Industry 4.0, a problem setting normally changes rapidly. This is a crucial step towards reliability of information, since an increase in reliability of these algorithms also leads to more accurate recommendations for maintenance actions. Although feedback mechanisms for continuous improvement and learning of the diagnostic and prognostic algorithms have been well perceived by academia and industry, mechanisms for tracking the suggested recommendations and for continuously improve the decision algorithms is an underexplored area. Currently, machine learning methods for updating the decision models have not been widely investigated in literature.

2.3.4 E-maintenance

E-maintenance has been increasingly used in many organizations in recent years, particularly in the USA and Europe not only because it reduces business risks, but also as a value-adding process in today's competitive business environment (Aboelmaged, 2015). Information and Communication Technologies (ICTs) are transforming the way systems are maintained, they provide the support to generate more systems behaviour knowledge and to introduce new tools and processes for a more proactive maintenance (Guillen et al., 2016). This maintenance support, has been defined as E-Maintenance (Muller et al., 2008): "Maintenance support which includes the resources, services and management necessary to enable proactive decision process execution. This support includes e-technologies (i.e. ICT, Web- based, tether-free, wireless, infotronics technologies) but also, e-maintenance activities (operations or processes) such as e-monitoring, e-diagnosis, e-prognosis, etc.” E-maintenance provides a new working context extending the service maintenance to a knowledge-driven organization, where the information flows integrating diverse processes (especially those related with monitoring and CBM), knowledge providers (technicians of the service provider, machinery builder/engineers/ technicians, and operators on field), and expert/decision support systems (intelligent systems). This includes the intelligent maintenance systems concept (Espindola et al., 2013; Guillen et al., 2016).

E-maintenance refers to the convergence of emerging information and communication technologies with information systems which take into account the resources, services and management to enable decision making in a proactive way (Muller et al, 2008a). E-maintenance has become important in the last years due to the emergence of technologies which are able to optimize maintenance-related workflows and the integration of business performance, which enable openness and interoperation of e-maintenance with other components of e-enterprise (Iung et al., 2009). This support does not include only technologies, but also operations and processes related to maintenance such as condition monitoring, diagnostics, prognostics, etc. (Muller et al., 2008a; Muller et al., 2008b; Irigaray et al., 2009; Levrat and Iung, 2007). E-maintenance is considered not only in production and operation stages but also as an integral part of the whole lifecycle management (Takata et al., 2004; Iung et al., 2009).

Despite the potential applications of e-maintenance, a number of issues need to be considered to successfully implement e-maintenance system in various contexts (Aboelmaged, 2015). Though e-maintenance research has grown rapidly over the past decade, there has been lack of emphasis on developing conceptual frameworks that integrate fragmented key themes within e-maintenance research stream. In the same vein, Kajko-Mattsson et al. (2011) indicated that e-maintenance research is still immature and suffers from lack of common definitions, lack of sound and widely accepted underlying theories, vague usage scope, and lack of commonly defined components inherent in the e-maintenance. Although literature on e-maintenance has debated the concept from various views with little consensus, careful content analysis of e-maintenance definitions reveal two key perspectives; managerial and engineering. The managerial perspective focuses e-maintenance as a strategy (e.g. Hausladen and Bechheim, 2004; Lee et al., 2006; Levrat et al., 2008; Muller et al., 2008b) or a set of supporting activities and processes such as monitoring, diagnosis, and prognosis of real-time system health data (e.g. Candell et al., 2009; Ucar and Qiu, 2005). Alternatively, engineering perspective emphasizes the role of information and communication technologies such as intelligent sensors, channels, software solutions, and e-collaboration methods that configure e-maintenance system (e.g. Han and Yang, 2006, Bangemann et al., 2006; Tao et al., 2003; Pistofidis et al., 2012).

Consequently, common characteristics of e-maintenance approach can be synthesized as follows (Aboelmaged, 2015):

- e-maintenance is a strategy
- e-maintenance supports decision making at different organizational levels
- e-maintenance has great opportunities for cost-effective decisions to be made
- e-maintenance integrates maintenance principles with e-business or e-technologies applications (e.g. telecommunications, web services, mobile, wireless and portable devices, and other means of electronic collaboration)
- e-maintenance monitors and manages systems and assets over the internet
- e-maintenance integrates production and maintenance operations systems
- e-maintenance collects feedback from remote customer sites and integrates it to upper level enterprise applications
- e-maintenance generates dynamic and real-time maintenance information that enables knowledge application for assets and production systems

- e-maintenance includes scientific approaches and methodologies that prognosis system's well-being and increases its productivity for better competitiveness.

To deal with the challenges arising out of high volume of data generated by machines in Industry 4.0 scenario, big data and advanced tools are developed and implemented so that data can be systematically processed into information and facilitate decision-making with more information in real time. However, the design of e-maintenance solutions remains a task with several challenges:

- ***Organizational Challenges***: These challenges mainly focus on enterprise resource management related aspects like (1) organizations restructuring for those involved in maintenance, (2) resource planning (e.g. spare part, material, etc.), (3) information management, (4) management of heterogeneous organizations and (5) knowledge management.
- ***Architectural Challenges***: Challenges dealing with the issues of the architecture of eMaintenance solutions, like (1) developing framework for eMaintenance development, (2) developing models for distributed processing and analysis of data, (3) service model developmentfor distributed data analysis, (4) developing prognostic tool-based models, (5) model development for visualization of relevant data that supports interaction between human and machine, and (6) developing model for dispersed data storage capability.
- ***Infrastructural Challenges***: When services, according to SOA, are developed and implemented, infrastructural challenges arise to address to the issues pertaining to providing necessary tools and technologies required to meet the service needs and requirements. Some of these challenges include (1) wired/wireless network infrastructure, (2) service and user authentication, (3) mechanism for safety and security, (4) maintainability of eMaintenance services, (5) availability performance management and tracing and tracking mechanism and (6) establishment of documentation and archiving mechanism.

- ***Content and Contextual Challenges***: There are those challenges that are connected with the data sourced from eMaintenance services, like (1) establishing appropriate ontology through which data from data sources (e.g. process, product, condition monitoring and business data) are integrated smoothly, 2) providing quality assurance mechanism so as to increase decision-making quality, (3) providing mechanism to establish user's current situation so as to adapt information to user's context, (4) mechanism to manage uncertainty in data sets, (5) mechanism for describing various context and (6) for pattern recognition.
- ***Integration Challenges***: Coordination, integration and orchestration of services and data managed by eMaintenance solution raises integration challenges like (1) service management, interaction and interactivity, (2) management of configurations and (3) enablement of integration capability across a multiplatform and technologies.

Failure of critical assets was rated as the most significant risk to operational performance. This fact has led to an increasing demand for predictive maintenance information systems and technologies for preventing asset failure, detecting quality issues, improving operational processes, etc. To this end, several software companies have developed systems for predictive maintenance-related aspects (e.g. IBM: Predictive Maintenance, SAP: Predictive Maintenance and Service, Software AG: IoT Predictive Maintenance, BOSCH: Predictive Maintenance, SAS: The Early Warning Project on Predictive Maintenance). Moreover, several research projects resulted in e-maintenance prototypes. The development of e-maintenance prototype systems can be distinguished in two chronological periods which also have different characteristics. The first wave of appearance, development and deployment of e-maintenance concepts and prototypes in the context of research projects was during the period 2003-2008, see e.g. Watchdog Agent (Djurdjanovic et al., 2003) in the international research project "Embedded Watchdog Agent/ Lifecycle Unit (EWA/LCU)", TELMA Platform (Iung, 2003; Levrat and Iung, 2007; Levrat et al., 2008) in "DYNAMITE" EU project and PROTEUS (Szymanski et al., 2003; Bangemann et al., 2006) in the PROTEUS project. The second wave of emerging e-maintenance paradigms appeared in 2014 and is still evolving in the context of national and EU projects (e.g. SUPREME, iMain, RepAIR, MANTIS, preInO, UPTIME). The second wave appeared due to the emerging opportunities of the industrial IoT, big data infrastructures and communication devices, but also due to the increasing financial pressures which have led to a significant demand of eliminating maintenance costs by optimizing business performance.

2.4 Proactive Computing

2.4.1 Proactivity in Information Systems

Most applications currently supported by event processing platforms are reactive by nature. There have been various research efforts reported on proactive event-driven computing providing promising results in terms of processes optimization in the areas of compliance (Thullneret al., 2011), network management (Fu, and Xu, 2010), task execution (Hocheol et al., 2010), traffic management (Artikis et al., 2014; Wang, and Cao, 2014), healthcare (He et al., 2017), logistics (Feldman et al., 2013), credit card fraud management (Artikis et al., 2014) and industrial maintenance applications (Sejdovic, and Kleiner, 2016). However, they developed conceptual or ad-hoc models that are not easily reused for other purposes. The underlying motivation of proactive computing stems from social and economic factors, and is based on the fact that prevention is often more effective than cure.

The term proactive computing was first introduced by Tennenhouse (2000). However, anticipatory systems can be considered as the origin of proactive computing. Rosen in 1985 defined the anticipatory system as “A system containing a predictive model of itself and/or its environment, which allows it to change state at an instant in accord with the model’s predictions pertaining to a later instant” (Rosen, 2012). Although the contexts in which questions such as “what should we do now?” are posed are different, they are all alike in their fundamental concern with the making of policy, and the associated notions of forecasting the future and planning for it (Louie, 2010; Nadin, 2016). A reactive system can only react, in the present, to changes that have already occurred in the causal chain, while an anticipatory system's present behavior involves aspects of past, present, and future (Louie, 2010). The presence of a predictive model serves precisely to pull the future into the present; a system with a ''good'' model thus behaves in many ways as if it can anticipate the future. Model-based behavior requires an entirely new paradigm, an ''anticipatory paradigm'', to accommodate it. This paradigm extends – but does not replace – the ''reactive paradigm'' which has dominated the study of natural systems (Nadin, 2016).

According to Tennenhouse, proactive computing describes the evolution away from interactive computing, i.e., from classical human-centered workstation settings to human-(un)supervised pervasive computing scenarios. Tennnehouse's two principles were: getting human above the loop (instead of in the loop) of computing, and respond to human stimuli faster than human abilities. Tennenhouse's proactive term overlaps with the term autonomic computing that emerged later, however it characterizes both systems that exhibit reactive behavior, in the sense that they react to event that already happened; proactive behavior is about dealing with events before they happen. Tennenhouse conducted a fundamental re-examination of the boundary between the physical and virtual worlds; changes in the time constants at which computation is applied; and movement from human-centered to human-supervised (or even unsupervised) computing. He identified three main requirements for proactive systems:

- Getting physical. Proactive systems will be intimately connected to the world around them, using sensors and actuators to both monitor and shape their physical surroundings. Research into "getting worked systems to their environments.
- Getting real. Proactive computers will routinely respond to external stimuli at faster-than-human speeds. Research in this area must bridge the gap between control theory and computer science in the form of software- and network-enabled control (e.g. control regimens that tolerate statistical variations in component availability and connectivity).

- Getting out. Interactive computing deliberately places human beings in the loop. However, shrinking time constants and sheer numbers demand research into proactive modes of operation in which humans are above the loop.

Want et al. (2003) further discuss proactive computing as well as the differences to autonomic computing. The aim of proactive computing is unobtrusive systems that connect to the physical world and require as little human interaction as possible. Further, they should anticipate the user's needs and act on his/ her behalf. The authors identify seven principles as foundations of proactive systems: connecting with the physical world, deep networking, macro-processing, dealing with uncertainty, anticipation, closing the control loop, and making systems personal. Despite leading to similar techniques, autonomic computing, in contrast, describes the discipline of managing the complexity of a heterogeneous system through appropriate system design principles. Salovaara, and Oulasvirta (2004) discuss the general concept of proactive computing. They suggest that a system can act proactively, if it has a hypothesis about what its user's goals are. In order to achieve these goals, the system makes use of different resources. The authors present a classification of six different types of proactive resource management in order to become a proactive system: preparation, optimization, advising, manipulation, inhibition, and finalization of user's resources.

Handte et al. (2012) describe proactivity from an adaptation perspective as modifications of an application performed before an application can no longer be executed. Vansyckel et al. (2013) further included context adaptation as a necessity, in order to be able to avoid having to adapt the application itself. As an example, in (Vainio et al., 2008), the system automatically adjusts the lighting of the environment based on what it anticipates the users desire is. Hence, it acts on the users behalf. However, it does so after it notices a change, i.e., in a reactive manner from an adaptation standpoint. Hence, proactive can refer to before the user acts, or before the triggering event happens, respectively. The main difference is that in order to act before an event takes place, the system must have knowledge of that event and, hence, requires prediction.

The evolution from responsive to reactive computing was achieved with the development of models and tools to express and execute reactive systems in an easy way. This major breakthrough turned event-driven applications pervasive and part of the main-stream computing (Engel et al., 2012). A similar evolution is necessary in order to enable pervasive use of proactive computing. Building on EDA, proactive event-driven computing is an evolving paradigm where a decision is neither made due to explicit requests nor as a response to events, but is triggered by real-time predictions about a future predicted event and is made under time constraints by exploiting large amounts of historical and streaming data (Engel et al., 2012).

Proactivity refers to the ability to avoid or eliminate the impact of undesired future events, or to exploit future opportunities, by applying predictive models combined with real-time sensor data and automated decision making technologies (Engel et al., 2012). Consequently, proactivity in terms of information systems is driven by predictions, leading to in-creased situation awareness and decision making capabilities ahead of time (Engel et al., 2012). In proactive event processing, a proactive situation deals with the prediction of a future undesired event based on real-time data streams and with decision making on the basis of the predicted event before it occurs. Therefore, proactive event pro-cessing must include the notion of a future event, the identification of predictive event patterns, and possible courses of actions (Engel et al., 2012).

The proactivity principle extends the reactive one underlying the Sensing Enterprise, referred in literature as sense-and-response (Elwany, and Gebraeel, 2008) or detect-and-act (Tao et al., 2014), to a new model of situational awareness, consisting of four phases: ***Detect, Predict, Decide, Act*** (Engel et al., 2012; RTInsights, 2016). Detect deals with monitoring the universe; a detection of the current indicators. Predict utilizes the current indicators in order to forecast that the system is going to a state outside the admissible state in the future if nothing changes. The Decide phase results in a real-time decision about the best way to eliminate or mitigate the problem and stay within the admissible states. The Act phase has to do with the actual implementation of the action.

According to Lundberg (2006), companies that are capable of analyzing their business operations based on the rapidly growing mass of data, of predicting the best proceeding process sequence, and proactively controlling their processes based on this knowledge will be a decisive step ahead of their competitors. This kind of company sketches the vision of a ‘‘Predictive Enterprise’’ as the next stage in the evolution of real-time enterprises within the age of data as a crucial competitive asset.

PwC (2016) introduced the concept of “proactive organization”, mainly focusing on the services offered to customers. According to PwC, a proactive organization aims to identify and capture digital signals, to identify the right moment to offer services and to identify the right mode of sevice delivery. Therefore, proactive organisations recognise the critical value of data and are continuously looking for new sources of data and ways of gaining meaningful insights from it. This data treatment should be subjected to privacy and confidentiality regulations and should be used for better decisions.

Krumeich et al. (2016) proposed the concept of “prescriptive enterprise” and concluded in an architectural paradigm consisting of five layers: Integration Layer, dealing with Events, Transactions, Process Data and Big Data; Descriptive Analytics Layer, dealing with In-Memory Data Management and Connectivity; Predictive Analytics Layer, dealing with streaming analytics; Prescriptive Analytics Layer, dealing with real-time decisions; Adaptation Layer, dealing with intelligent actions and adaptation mechanisms.

Several factors in today’s computing infrastructure open the door for this breakthrough: (i) the growing availability of affordable and pervasive sensor technology, (ii) the spreading of broadband connectivity, and (iii) the developments in predictive analytics technology. The latter highlights a different angle to this process. Analytics has evolved from being merely descriptive (understanding of historical data), to being predictive (providing forecasts of future behavior). The next step is prescriptive analytics, a term which stands for the use of data to prescribe the best course of action to realize the best outcome (Russel, and Norvig, 2016). We can view the proactive idea as the event-driven variation of prescriptive analytics; reactive computing, coupled with predictive analytics, yields the ability to react to events before they occur, which is the essence of proactive event-driven computing (Engel, and Etzion, 2011).

Proactive computing can enhance the concept of “Sensing Enterprise”. “Sensing Enterprise” refers to the ability of the enterprise to process information captured by sensors and to provide added value insights (Camarinha-Matos et al., 2013) by taking advantage of IoT advances such as advanced sensor fusion, faster wireless connectivity and real-time predictive analytics (Li et al., 2015). The sensing enterprise incorporates reactive behaviors, providing direct links between “stimuli” and actions (Santucci et al., 2012). To this end, EDAs are able to close the business - ICT gap by delivering appropriate business functionality and enabling interconnectivity at an object level (Potocnik, and Juric, 2014). However, most applications currently supported by event processing platforms are reactive by nature. Reactive event processing deals with detection of situations and reaction to them. A reactive situation is an event occurrence that might require a reaction (Engel et al., 2011).

On the contrary, a proactive event-driven architecture combines advanced event processing with dynamic forecasting capabilities leveraged towards online optimisation and decision-making. The decisions are made in real time and require swift and immediate processing of Big Data, that is, extremely large amounts of noisy data flooding in from various locations, as well as historical data. The implementation of proactive event-driven computing in an enterprise context is shown in Figure 2-14. Achieving this vision requires novel research in three different directions (Artikis et al., 2012; Artikis et al., 2014; Fournier et al., 2015):

- **Dealing with large quantities of data**. Massive volumes of historical data and massive streaming data have to be analyzed to forecast events. Most systems are not capable of handling big data in real-time because of scalability problems, the need to cleanse noisy data offline, or the difficulty in fusing different types of data coming from different sources online. The result is that most analyses are done on offline data, while online data is not leveraged for immediate operational decisions.

- **Extending the state-of-the-art in event processing to deal with future events and uncertainty due to incomplete and noisy streaming data**. The ability to process past events and forecast future ones makes proactive systems a compelling application area. But, the uncertain nature of future events requires a major leap in event processing systems.
- **Devising methods for making near-optimal decision within time constraints**. The decision about which is the best action to take in proactive computing has two properties that differ from most contemporary decision support systems: (1) the decision should be taken on-line and under real-time constraints, which may dictate the use of approximation techniques and (2) The decision often entails autonomic actions, rather than providing only recommendations for human decision makers.

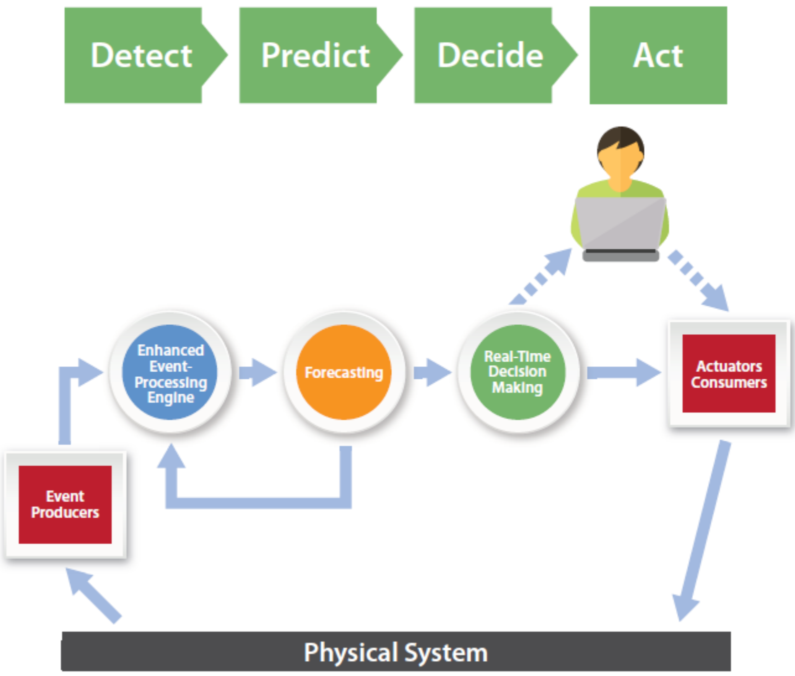

Figure 2-14: Implementation of proactive event-driven computing in an enterprise context (RTInsights, 2016)

2.4.2 Event Processing

In the context of the “Sensing Enterprise”, physical and virtual sensing devices such as sensors, actuators and controllers are able to observe changes in terms of the systems examined and their condition and to generate data in the form of events, which are then further processed by an appropriate information system (Da Xu et al., 2014). In addition, web-service communication technologies can be exploited in order to effectively integrate sensors into a multi-layered real-time big data architecture. To do this, the EDA paradigm is appropriate for closing the ICT-business gap.

In recent years, there are several attempts of coupling event processing architecttures with proactive computing for overcoming challenges of efficiency and scalability. IoT aims to connect different things over the networks. As a key technology in integrating heterogeneous systems or devices, Service-Oriented Architecture (SOA) can be applied to support IoT. The architectural design of IoT is concerned with architecture styles, networking and communication, smart objects, Web services and applications, business models and corresponding process, cooperative data processing, security, etc. (Da Xu, et al., 2014). From the technology perspective, the design of an IoT architecture needs to consider extensibility, scalability, modularity, and interoperability among heterogeneous devices.

As things might move or need real-time interaction with their environment, an adaptive architecture is needed to help devices dynamically interact with other things. The decentralized and heterogeneous nature of IoT requires that the architecture provides IoT efficient event-driven capability. Thus, SOA is considered an effective approach to achieve interoperability between heterogeneous devices in a multitude of way (Xu, 2011; Miorandi, et al., 2012; Da Xu, et al., 2014). Designing a SOA for IoT is a big challenge, in which service-based things might suffer from performance and cost limitations. In addition, scalability issues often arise as more and more physical objects are connected to the network. When the number of things is large, scalability is problematic at different levels including data transfer and networking, data processing and management, and service provisioning (Miorandi et al., 2012).

2.4.2.1 Service-Oriented Architecture (SOA) and Event-Driven Architecture (EDA)

In recent years there has been much use of the terms Event-Driven Architecture (EDA) and SOA. SOA is a distributed software architecture where self-contained applications expose themselves as services, which other applications can connect to and use. To reach its full potential, SOA applications should be self-describing, discoverable, and platform- and language-independent (Papazoglou, 2008). This leads to loose coupling and high flexibility. The adoption of SOA in a company typically starts as an IT initiative to improve infrastructure efficiency and can then mature into optimised use for business purposes. One of the most common ways to implement SOAs are web services (De Prado et al., 2017). Web services are self-descriptive software modules which can be accessed through a net and which develop a task facilitating machine to machine interoperability (Papazoglou, 2008).

REST web services emerged as an alternative to more traditional SOAP web services. REST is an architectural style for distributed hypermedia systems where services provide resources identified by URLs (Fielding, and Taylor, 2000). Communications between REST services and their clients take place using HTTP main operations, mainly GET, POST, PUT and DELETE. With the growth of service components and processes in service oriented applications, a new service infrastructure is required for maintaining applications in a flexible way. This infrastructure must support well-known web service standards and provide support for a message middleware (Papazoglou, 2008). These requirements are ful lled by an ESB. An ESB provides services to complex architectures through a messaging system, supplying interoperability among diverse applications and components through standard interfaces; that allows applications to be offered as services in the ESB (De Prado et al., 2017).

Event-based programming, also called EDA is an architectural style in which one or more components in a software system execute in response to receiving one or more event notifications (Etzion et al., 2011). An event is an indication of something that has already happened, whereas a request, expresses the requestor's wish that something specific should happen in the future (Luckham, 2002). In a decoupled event processing system, an event producer does not depend on a particular processing or course of action being taken by an event consumer (Etzion et al., 2011). Moreover, an event consumer does not depend on processing performed by the producer other than the production of the event itself. In a decoupled system there can be more than one consumer of an event, and the action taken can vary significantly among consumers (Etzion et al., 2011). It can also vary during the lifetime of the application. As an event producer does not know what an event consumer is going to do with an event, or even how many consumers there are, it usually does not make sense for the event producer to expect a response to its events (Etzion et al., 2011).

SOA and EDA were considered to be different architectures. However, there is a consensus during the last years. In this way, they are not considered to indicate alternative architectures but it is possible to use event processing within an overall SOA. In other words, EDA can complement SOA because services can be activated by triggers fired on incoming events (Luckham, 2012). For this reason, there have been several attempts for the development of SOA 2.0 (also called advanced SOA or event-driven SOA) that focuses on events, inspired by EDA (De Prado et al., 2017).

Even though SOA conceptually offers loose coupling and is intended to be distributed, service orchestration is typically done centrally, with the orchestrator taking control of the involved services. EDA is extremely loosely coupled and highly distributed by design. An event creator only needs to know that the event occurred, it does not need to know anything about who is interested in the event or how it will be processed (Engel et al., 2011). With EDA, applications turn from synchronised and blocking to asynchronous and non-blocking (Engel et al., 2011).

In fact the term event-driven SOA is now used by some analysts and vendors to denote the combination of EDA and SOA. An event-based programming approach can be mixed with request-response components in a SOA in two ways (De Prado et al., 2017): (i) It is possible for a component to implement both approaches. In other words, it can provide or consume a request-response interface and also be an event producer or event consumer. (ii) The SOA infrastructure that hosts the SOA components can provide instrumentation that produces events on behalf of request-response style services.

2.4.2.2 Main concepts of event processing

EDAs and conceptual models that support them have evolved in the last several years, departing from the traditional computing architectures which employ synchronous, request-response interactions between client and servers. This is a paradigm shift in two senses (Engel et al., 2012): first, event driven architectures support applications that are reactive in nature, in which processing is triggered in response to events, contrary to traditional responsive applications, in which processing is done in response to an explicit request. Second, event driven architecture adhere to the decoupling principle, in which there are event producers, event consumers and event processing agents that are mutually independent. Figure 2-15 shows an illustration of such architecture, showing the logical separation of event processing logic from the event producers and event consumers (Etzion, and Niblett, 2010).

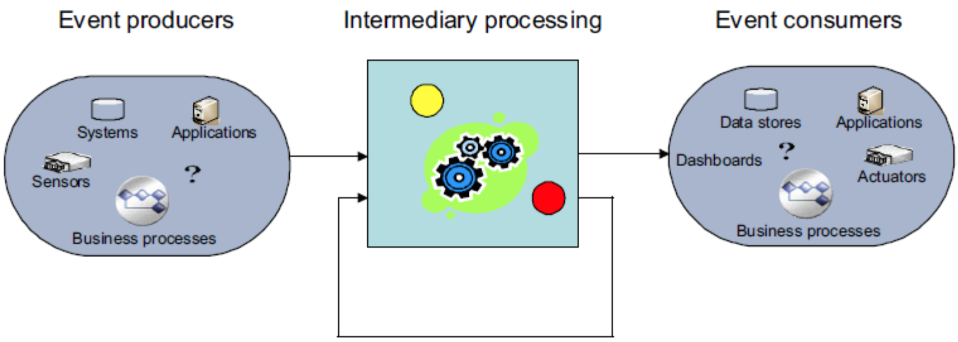

Figure 2-15: The major architectural components of event processing architecture (Etzion, and Niblett, 2010)

An EDA consists of event producers and event consumers, while it also incorporates event processing agents and an event processing network (Etzion, and Niblett, 2010). An **event producer** is an entity at the edge of an event processing system that introduces events into the system. An **event consumer** is an entity at the edge of an event processing system that receives events from the system. An **event processing agent** is a software module that processes events. An **event processing network (EPN)** is a collection of event processing agents, producers, consumers, and global state elements connected by a collection of channels. An example showing the event processing components is depicted in Figure 2-16.

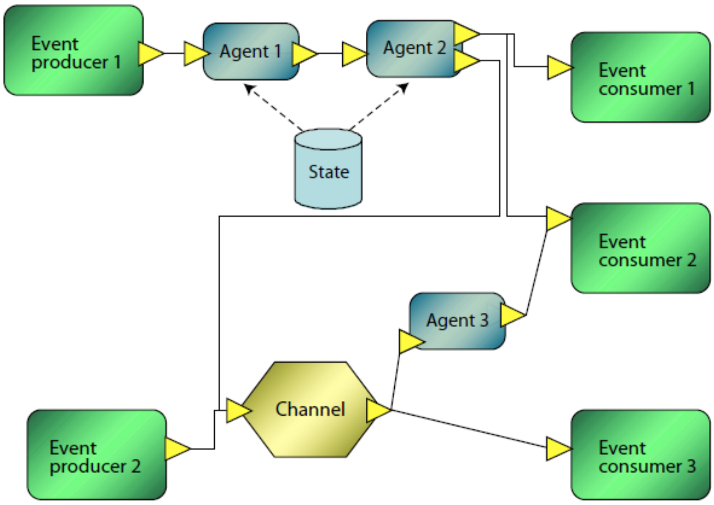

Figure 2-16: An example of an Event Processing Network (EPN). (Etzion, and Niblett, 2010)

An **event stream** (or stream) is a set of associated events. It is often a temporally totally ordered set (that is to say, there is a well-defined timestamp-based order to the events in the stream). A stream in which all the events must be of the same type is called a homogeneous event stream; a stream in which the events may be of different types is referred to as a heterogeneous event stream. Streams can be a convenient way to think of and model an event processing application. Some event processing systems make the stream their major abstraction. It can be more natural to think of an event processing agent as operating on an entire stream of events, rather than as operating on each event one by one.

An event processing building block represents an event processing concept and is used to create platform-independent definition elements, which are implementation neutral instances of this building block (Etzion, and Niblett, 2010). For example, we can use the event type building block to create implementation-neutral representations of the event types needed by an application such as the Delivery Request event type used in the Fast Flower Delivery application. Each application is made up of a collection of these definition elements, customized to perform a particular role and connected together to form an event processing network. When the application is implemented, these platform-independent definition elements have to be translated into one or more platform-specific runtime artifacts, using platform-specific tools.

Any event-driven application will involve one or more different types of events and, as its name suggests, the event type building block allows us to describe these types (Etzion, and Niblett, 2010). This building block defines the structure of an event (this is sometimes called an event schema) along with some of its semantics. The event producer and event consumer building blocks are used to represent the concepts of the same name. The event producer represents an application entity that emits events into the EPN, and the event consumer an application entity that receives them. These building blocks model only those bits of the behavior of the event producer or consumer that are visible to other components of an event processing network. So the event producer building block does not specify how an event producer instance actually comes to emit an event, and the event consumer building block does not specify what an event consumer instance does when it consumes an event. The event producer or event consumer definition element can represent either a single producer or consumer instance, or a whole class of such instances. In some applications there might be just one instance of the producer (for example, if the producer is a firewall router raising alert events); in other cases there might be many instances (for example, smoke detectors in a building). Where there are many instances it would be tedious to require every one to be represented by a separate definition element.

The event processing agent building block represents a piece of intermediary event processing logic inserted between event producers and event consumers. In contrast to the event producer and event consumer, the event processing agent building block models the behavior of the agents built from it (Etzion, and Niblett, 2010). EPAs are further refined into types, as shown in Figure 2-17.

An event channel's principal job is to route events between event producers and event consumers. Apart from the five building blocks already introduced, there are two further building blocks (Etzion, and Niblett, 2010): the context building block and the global state element building block. A context element collects a set of conditions from various dimensions (temporal, spatial, segmentation-oriented, and state-oriented), giving us a way to categorize event instances so that they can be routed to appropriate agent instances. For example, you can use a segmentation-oriented context to make sure that events relating to different customers are handled by different event processing agent instances. A global state element refers to data that is available for use both by event processing agents and by contexts. This data may be system-wide global variables, reference data used to enrich events, and event stores that hold past events. The seven fundamental building blocks of event processing are shown in Figure 2-18.

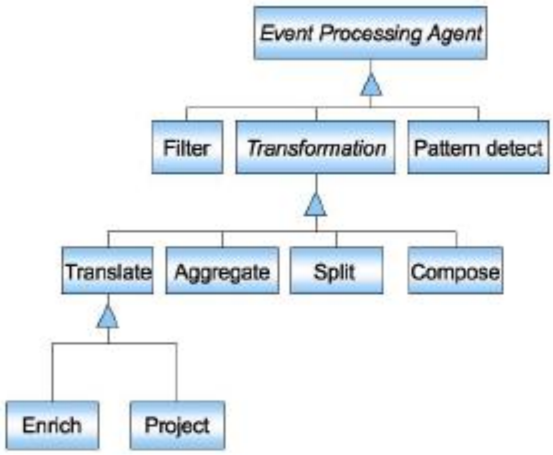

Figure 2-17: Event Processing Agent Types (Etzion, and Niblett, 2010)

It is not surprising that there is a pressing need for real-time recognition of events in the multitude of data that is being recorded and processed. This requirement may be addressed by employing recognition systems that detect situations or events of special significance within an organization, given streams of `low-level' information that are very difficult to be utilized by humans. The vast majority of today's event processing systems focus on the efficiency of reasoning algorithms. However, these don't take into account the various types of uncertainty that exist in most applications (Engel et al., 2012). As big data applications, many of the emerging event processing systems are required to process events that arrive from sources such as sensors and social media, which have inherent uncertainties associated with them. In these cases, the streams of events may be incomplete or inaccurate, for example, regarding the time and location of events.

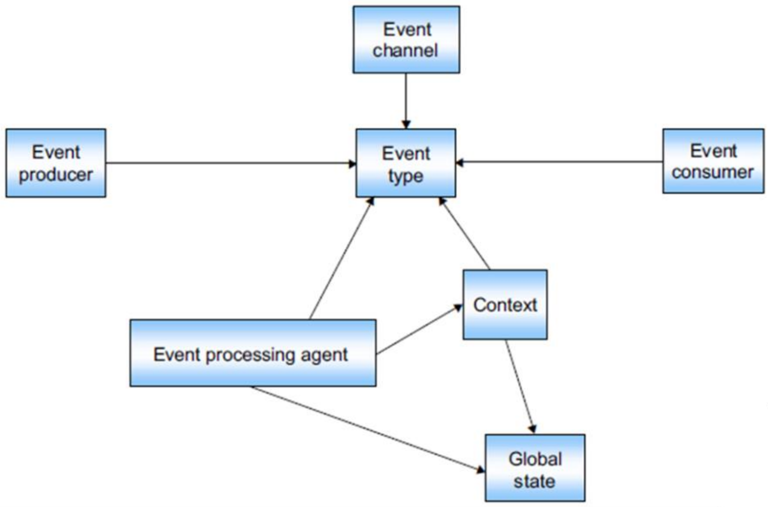

Figure 2-18: The fundamental building blocks of event processing (Etzion, and Niblett, 2010)

2.5 Synthesis of Literature Review

The emergence of the Internet of Things (IoT) has paved the way for enhancing the monitoring capabilities of enterprises with the extensive use of physical and virtual sensors. Taking advantage of the big data generated from a large amount of sensors requires the development of event monitoring and data processing systems that are able to handle real-time data in complex, dynamic environments in order to get meaningful insights about business performance and increase data analytics maturity. The EDA paradigm strongly contributes to the development of monitoring and control architecture, enabling interconnectivity at an object level, and consequently impacting e-maintenance platforms (Karim, 2009).

To this end, proactive event-driven computing leads to the possibility to decide and act ahead of time, i.e., to be proactive in resolving problems before they appear or realizing opportunities before they become evident and be able to recover and support continuity according to the “Detect, Predict, Decide, Act” proactive pattern (Engel et al., 2012). The need for a business turning from reactive to proactive is increasing. Proactive enterprise leads to increased situation awareness capabilities ahead of time. This leads to a new class of enterprise systems, proactive systems, that will be continuously aware of that what “might happen” in the relevant business context and optimize their behavior to achieve what “should be the best action” even during stress and balancing on demanding margins. The manufacturing domain, and especially the maintenance operations, can significantly benefit from the “Proactive Enterprise” concept.

Maintenance operations are a major part of the total operating costs. Studies show that approximately 60% of all the manufacturing equipment fails prematurely after the implementation of corrective maintenance actions (Karim et al., 2009). Insufficient maintenance management can result in equipment deterioration and quality defects which correspond to financial losses due to delays, customer complaints, and purchasing of new equipment spare parts (Ollila and Malmipuro, 1999).

Since manufacturers increasingly see maintenance as a strategic business function for maintenance costs, downtime reduction and asset lifecycle increase, it is no longer viewed as a "necessary evil". Manufacturers now have more alternatives than ever to employ a costly "run until it breaks" maintenance strategy, or an inefficient "fix it regardless" maintenance approach. To this end, Predictive Maintenance has been emerging during the last years in conjunction with the use of IoT-based condition monitoring technology and data analytics capabilities. Predictive maintenance is an evolving maintenance strategy that is increasingly gathering the interest of modern manufacturing companies. Various predictive maintenance frameworks have been proposed in both the industrial and academic realms. Based on the existing frameworks, predictive maintenance consists of four main steps: Signal Processing, Diagnosis, Prognosis, Decision Making.

Moreover, since automation over available predictive maintenance services is crucial to build manufacturing value-driven solutions (Macchi et al., 2014; Camarinha-Matos et al., 2013; Aboelmaged, 2015; Guillén et al., 2016), the e-maintenance paradigm has become important in the last years due to the emergence of technologies which are able to optimize maintenance-related workflows and the integration of business performance, which enable openness and interoperation of e-maintenance with other components of e-enterprise (Iung et al., 2009). E-maintenance facilitates a higher degree of proactivity, supporting greater control and capacity to act on the systems, including efficiency and effectiveness of maintenance plans monitoring (Muller et al., 2008; Guillen et al., 2016). This fact leads to the need of adopting "more proactive strategies" in maintenance management (Guillen et al., 2016). Although e-technologies provide several advantages, optimization of e-maintenance benefits with the aim to improve the production system performance requires not only technology, but also appropriate models, methods and methodologies (Muller et al., 2008b; Irigaray et al., 2009).

The future factory will take advantage of new capabilities and will enable the realization of sophisticated approaches based on the collaboration of devices, network services within the single enterprise and among enterprises (Cannata et al., 2010). This is a key issue especially for the maintenance; however, two main challenges should be overcome (Cannata et al., 2010): (i) Interoperability: several e-maintenance platforms are based on proprietary technologies, which implies higher costs and slow market adoption, since implementation costs and time are required; (ii) Scalability and flexibility: due to rapidly changing market and to on-going trend towards flexible and adaptive factories, there is a need for scalable platforms in order to effectively support the changing conditions.

Currently, there is still a lack of services and tools capable of efficiently processing real-time big data from heterogeneous sources, implementing complex algorithms and provide meaningful insights about potential problems along with a continuous self-improvement approach (Camarinha-Matos et al., 2013). The capabilities of proactive event-driven decision making have not been examined in manufacturing operations, due to several challenges associated to large scale, big data-driven enterprise environments as well as due to the lack of appropriate algorithms. Moreover, there is a large gap for the effective implementation of predictive maintenance programs extensively in industry, mainly due to the complexity of these solutions and their life cycle and thus, due to the challenges in their practical implementation (Guillen et al., 2016).

Existing solutions suffer from several limitations: (i) Most of them focus on product maintenance, i.e. on the service stage of the Product Lifecycle Management (PLM) (e.g. warranty failures) and not on industrial maintenance, i.e. on the manufacturing stage of the PLM; (ii) They are mainly based upon physical, domain-specific models that are not easily extensible for other equipment or for other industries; (iii) They rarely exploit big data processing infrastructures for real-time, sensor data, since they usually use batches of data, while the level of data analytics maturity is usually low; (iv) Each one of them focuses on a specific aspect (e.g. condition monitoring, diagnostics, etc.) instead of having a unified approach for covering all the phases and industrial operations-related aspects.

3 Towards Proactive Maintenance Management

In this Chapter, the research questions are formulated and the thesis is presented. More specifically, the research questions along with their constituting parameters are described and an outline of the proposed solution is presented.

3.1 Introduction

Maintenance strategies have been evolving throughout the years towards more efficient ones by taking advantage of the development of technologies and information systems. Figure 3-1 shows the evolution maintenance strategies which lead to a higher positive impact on business performance, but also to a higher demand for increasing data analytics maturity.

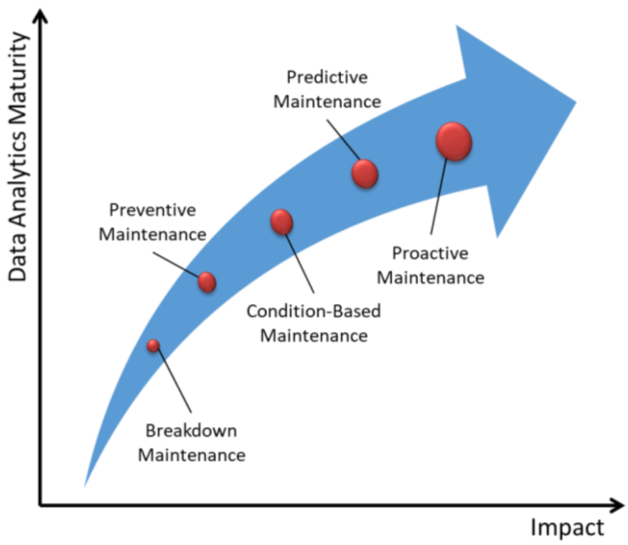

Figure 3-1: The evolution of maintenance strategies

Proactive Maintenance, which is the proposal of the current thesis, incorporates condition monitoring equipment, sensors and actuators generating huge amounts of real-time data. This IoT-based environment requires scalable and efficient methods, algorithms and systems for big data processing. It also takes advantage of the e-maintenance concept in order to enable proactive decision process execution with appropriate decision models and algorithms. In this way, there is an increased level of data analytics maturity, since they result in specific recommendations for maintenance optimization. In addition, the methods and algorithms used in Proactive Maintenance information systems can take into account other manufacturing operations as well in order to result in an optimized business performance.

3.2 Research Questions

This Section presents the four research questions of the current thesis. Table 3-1 outlines these research questions and their constituting parameters.

Table 3-1: Research Questions and their Parameters

Research Questions	Parameters
What is the next generation of maintenance in an IoT-based industrial environment?	• How to support a novel lever of maintenance management in an IoT-based industrial environment? • How to develop a generic maintenance framework taking into account the most recent advancements in maintenance management and computer science? • What new aspects should be investigated in order to enable the aforementioned framework's implementation?
How to support proactive decision making in maintenance operations?	• How to support real-time, event-driven proactive decision making in maintenance operations? • What are the interactions of maintenance management with other industrial operations? • What decision methods and technical requirements are needed?
How to conduct continuous improvement of proactive decision making?	• How to improve the accuracy of the cost-related input parameters of the decision methods and thus, the reliability of the generated recommendations themselves in an event-driven infrastructure? • How to provide meaningful visualization and real-time monitoring of the actual cost performance?
How to incorporate context-awareness in proactive decision making and actions implementation?	• How to consider operations-related context in maintenance optimization? • How to consider context-aware costs under uncertainty in a real-time, event-driven computational environment?

3.2.1 *Research Question 1: What is the next generation of maintenance in an IoT-based industrial environment?*

Existing modern maintenance solutions suffer from several limitations: (i) Most of them focus on product maintenance, i.e. on the service stage of the Product Lifecycle Management (PLM) (e.g. warranty failures) and not on industrial maintenance, i.e. on the manufacturing stage of the PLM; (ii) They are mainly based upon physical, domain-specific models that are not easily extensible for other equipment or for other industries; (iii) They rarely exploit big data processing infrastructures for real-time, sensor data, since they usually use batches of data, while the level of data analytics maturity is usually low; (iv) Each one of them focuses on a specific aspect of maintenance (e.g. condition monitoring, diagnostics, etc.) instead of having a unified approach for covering all the phases and industrial operations-related aspects; (v) They have been described at an abstract conceptual level with no practical applications as a result of the high complexity of maintenance solutions; (vi) They have not been validated in an industrial environment as a result of manufacturing companies' reluctancy or aversion to change.

To this end, the current thesis aims to explicitly define the next generation of maintenance management by converging and synthesizing predictive maintenance, proactive computing, Industry 4.0, IoT, Big Data and the ISO 13374 as implemented to MIMOSA OSA-CBM. For this reason, it will examine its key characteristics, the advancements in technologies and information systems engineering to be exploited and will conclude to a generic conceptual architecture that can be seen as blueprint for maintenance applications in a sensor-driven, big data-rich industrial environment. Finally, it will identify the most developed aspects of Proactive Maintenance and will identify the gaps that should be addressed. The answer to this research question will enable to have a common understanding and will facilitate Proactive Maintenance implementation in modern manufacturing firms.

3.2.2 Research Question 2: How to support proactive decision making in maintenance operations?

Automation of maintenance decisions on the basis of real-time sensor-driven prognostic information is an unexplored area. Existing works provide only a diagnostic or a prognostic output, while they rely on processing of batches of data and not on real-time, event-driven information. In addition, several research works in proactive computing have only been described conceptually and have not been embedded in a real-time, event-driven environment. The convergence of maintenance management and proactive event-driven computing in the frame of Industry 4.0 can significantly enable overcoming the aforementioned challenges.

Following RQ1, the current thesis aims to fulfil the research gaps existing at the Decide phase of the “Detect- Predict-Decide- Act” proactivity principle in the context of maintenance decisions in conjunction with other interrelated operational decisions. The Decide phase is still an unexplored area in terms of methods, models and technologies. Consequently, the current thesis aims to investigate and develop proactive decision methods capable of addressing maintenance-related aspects in a real-time, event-driven infrastructure in order to provide proactive recommendations that can lead to expected losses minimization and improvement of the overall business performance. Therefore, RQ2 can be analysed to the following questions:

- How to support real-time, event-driven proactive decision making in manufacturing operations such as maintenance, spare parts inventory and supplier selection?
- What decision methods and technical specifications are required?
- What are the interactions of maintenance management with other industrial operations?

3.2.3 Research Question 3: How to conduct continuous improvement of proactive decision making?

Proactive event-driven decision making is highly sensitive to its input parameters, especially to those related to cost. Even slightly different action cost values compared to their actual ones may lead to the recommendation of a wrong (not optimal) action and/or timing for its implementation. Since cost related information may be either estimated by humans or measured through sensors, these deviations may occur due to user input's inaccuracies or the quality of collected data (e.g. due to sensor noise), respectively.

To overcome the aforementioned problems associated with the inaccuracy of manually inserted cost-related information and the resulting inaccurate recommendations, there is the need for continuous learning of cost parameters by considering the actual costs incurred because of the action during the time period it is implemented. Moreover, the user should be able to monitor in real-time the actual operational performance in terms of costs. Therefore, RQ3 can be analysed to the following questions:

- How to improve the accuracy of the cost-related input parameters of the decision methods and thus, the reliability of the generated recommendations themselves?
- How to provide meaningful visualization and real-time monitoring of the actual cost performance?

3.2.4 Research Question 4: How to incorporate context-awareness in proactive decision making?

The large amount of sensor-generated data leads to a strong demand for data-driven, real-time systems capable of efficiently processing them, in order to get meaningful insights about potential problems. Proactive decision making requires context-awareness (Engel et al., 2011); however, the high frequency of the real-time events and the high uncertainty pose challenges to the efficient handling of context-awareness.

Context-awareness has been considered in detection (Detect phase) and prediction (Predict phase) algorithms (Feng et al., 2009; Wan et al., 2014; Thaduri et al., 2014; Galar et al., 2015; Schmidt et al., 2016), but not in decision making algorithms and especially in proactive event-driven decision methods, where there is uncertainty about the future state of the system examined. In this sense, they have focused on reactive applications rather than proactive ones. Consequently, RQ4 can be analysed to the following questions:

- How to consider operations-related context in maintenance optimization?
- How to consider context-aware costs under uncertainty in a real-time, event-driven computational environment?

3.3 The Thesis

The proposed solution aims to fulfill the identified research gaps and thus, to answer to the aforementioned research questions. The way with which the current thesis addresses the research questions are described in the following sections. Table 3-2 shows the alignment of research questions with the thesis propositions along with the related publications and chapters (Chapters 4-7) of the current thesis. Chapter 8 deals with the development of the associated information system. Chapter 9 presents the deployment of the information system in industrial environment. Chapter 10 presents the evaluation results, while Chapter 11 the lessons learned and the managerial implications of adopting the proposed approaches.

The answers to the four Research Questions are outlined in Section 3.3.1.1, Section 3.3.1.2, Section 3.3.1.3 and Section 3.3.1.4 respectively. It should be noted that Section 3.3.1.1 deals with the overall framework for Proactive Maintenance, while Section 3.3.1.2, Section 3.3.1.3 and Section 3.3.1.4 zoom in the proactive decision making aspects, after the generation of predictions. On the basis of the aforementioned framework for Proactive Maintenance, the literature review reveals there are various research works dealing with real-time data-driven diagnostic and prognostic algorithms and information systems. However, the appropriate decision making algorithms are still unexplored providing a high potential for research with a high-value impact. To this end, the current thesis deals with proactive decision making making in maintenance management, continuous improvement of proactive decision making and context-awareness in proactive decision making. Proactive decision making is addressed with the functionalities presented at the conceptual architecture of Figure 3-2 and explained in the respective answers of the research questions.

Table 3-2: Alignment of Research Questions and Thesis Propositions

Research Questions	Thesis Proposition	Related Publications	Chapter
What is the next generation of maintenance in an IoT-based industrial environment?	Framework for Proactive Maintenance	j1, c1, c4, c5, c9, c13, c14	4
How to support proactive decision making in maintenance operations?	Proactive Decision Making	j3, j4, c2, c3, c8, c10, c11, c12	5
How to conduct continuous improvement of proactive decision making?	Continuous Improvement of Proactive Decision Making	j3, c6, c10, c11	6
How to incorporate context-awareness in proactive decision making and actions implementation?	Context-awareness in Pro-active Decision Making	j3, c7, c11	7

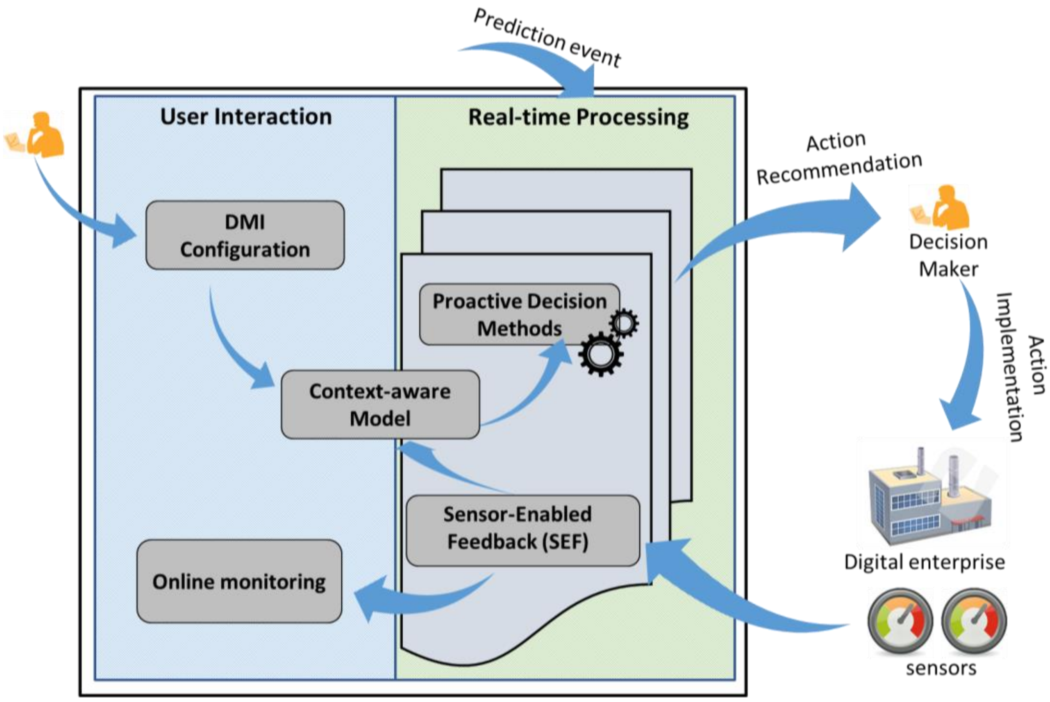

Figure 3-2: The Conceptual Architecture for Proactive Decision Making

3.3.1.1 Framework for Proactive Maintenance

To exploit the capabilities that IoT, big data processing technologies and proactive computing provided in the context of Industry 4.0, there is the need for a framework that will be able to facilitate the implementation of maintenance in an IoT-based industrial environment. To this end, a new generation of maintenance management is defined, i.e. Proactive Maintenance. Proactive Maintenance consists of the following phases: SENSE, DETECT, PREDICT, DECIDE, FMECA and ANALYZE.

Proactive Maintenance goes beyond traditional definitions and specifications of CBM and predictive maintenance and outlines the maintenance operations in the frame of Industry 4.0. To this end, the concept of Proactive Maintenance is presented. Moreover, the conceptual architecture of Proactive Maintenance is developed and its phases along with their inputs and outputs are defined. The Proactive Maintenance conceptual architecture is compatible with RAMI 4.0.

3.3.1.2 Proactive Decision Making in Maintenance Management

The aim is to address proactive decision making with **Proactive Decision Methods** capable of being embedded in a real-time, event-driven infrastructure triggered by sensor-generated data. At design-time, the users are able to select the most appropriate decision method for mitigating predicted undesired events, on the basis of functional and non-functional requirements, as well as to enter domain knowledge with the aim to define the **Decision Method Instance (DMI) Configuration**. DMIs are specific instances of decision methods, corresponding to a predicted undesired event (e.g. a business failure). The concept of DMI allows the extension of the system with more decision methods addressing different problems. For each DMI, domain knowledge entered by users may include the alternative mitigating actions, their cost functions, the cost of the undesired event as well as the decision horizon.

These decision methods should be able to provide real-time recommendations about optimal action(s) and optimal time for action(s) implementation on the basis of prediction events. Therefore, there is the need for development of such methods that deal with various maintenance-related operations in a manufacturing firm.

Moreover, they should be able to be incorporated in appropriate information systems that will be integrated with IoT devices through Detect and Predit systems. This thesis tackles with decisions about maintenance actions, joint maintenance and logistics actions and selection of maintenance spare parts' suppliers.

3.3.1.3 Continuous Improvement of Proactive Decision Making

The aim is to address the continuous improvement of proactive decision making with an adaptation mechanism. Proactive event-driven decision making is highly sensitive to its input parameters. The proposed approach, called **Sensor-Enabled Feedback (SEF)**, gathers and processes sensor-generated data during actions implementation in order to improve the accuracy of the input parameters required by the proactive decision methods and thus, the reliability of the generated recommendations. At design-time, through the **DMI Configuration**, the user is able to add additional parameters dedicated to the SEF mechanism. To this end, the user inserts the cost factors which each cost function consists of and maps each cost factor with the relevant sensor (e.g. a cost factor about cost due to production loss in a manufacturing enterprise is mapped to a sensor measuring productivity).

This approach is capable of being embedded in a big data infrastructure where sensors generate large amounts of data in the form of events.The SEF mechanism removes noise from sensor data and applies analytics techniques (i.e. curve fitting, anomaly detection) in order to derive the update parameter value. The role of SEF is twofold: (a) The user is informed online about the real-time estimated parameter (e.g. cost) along with the associated context before, during and after action implementation through the **Online Monitoring** functionality, and (b) The updated parameter value and the actual context within which it was obtained feed into the context model in order to update it with the new knowledge for the next time a prediction event is received and a recommendation is provided.

3.3.1.4 Context- awareness in Proactive Decision Making

The aim is to address context-awareness in proactive decision making with a probabilistic **Context-aware Model** that is updated through the SEF mechanism. Research on context-aware systems has focused on reactive applications rather than proactive ones, while it has focused on detection (Detect phase) and prediction (Predict phase) algorithms but not in decision making algorithms and especially in proactive event-driven decision methods. The context-awareness mechanism incorporates a machine learning approach for estimating the uncertain input parameters of proactive decision methods.

At design-time, the context-aware model is enacted as soon as the user inserts the domain knowledge required during **DMI Configuration** along with the associated affecting context in order to create the constraints and the causal relationships between the contextual elements and the affected input parameters of the proative decision methods. The output feeds into the Proactive Decision Methods block, which is triggered by real-time prediction events. The machine learning methods are suitable for intelligent context-aware systems in order to handle uncertainty (Thaduri et al., 2017) about the future context as well as about the values of input parameters. The use of a probabilistic context-aware model overcomes the challenge of the uncertainty regarding the context at the recommended time for the action imple-mentation, since it is used for the input parameter risk estimation (e.g. cost risk).

4 Framework for Proactive Maintenance

In this Chapter, the proposed framework for Proactive Maintenance is presented. This Chapter includes the definition and the description of the Proactive Maintenance concept, as well as the overall conceptual architecture for Proactive Maintenance.

4.1 Definition of Proactive Maintenance

Proactive Maintenance indicates the next generation of maintenance management with the aim to contribute to the digital transformation of manufacturing enterprises from reactive to proactive. Alternatively, it could be defined as Condition-based Predictive Maintenance in the frame of Industry 4.0. The term “predictive maintenance” that is often used does not necessarily include real-time condition monitoring through sensors. The term “Condition Based Maintenance” that is usually used does not necessarily include predictions, since it may refer to (near) real-time diagnostic outcomes, i.e. detection of the current condition.

Proactive Maintenance is based upon four technological pillars: **Industry 4.0**, **IoT**, **Big Data** and **Proactive Computing**. To this end, Proactive Maintenance is a new maintenance strategy that is based to a large extent on the IoT technologies and real-time information systems. It takes advantage of the industrial IoT infrastructure, sensor-generated big data processing technologies and e-maintenance services with the aim to provide real-time monitoring, detections, predictions and proactive recommendations about maintenance actions. The aim is to eliminate or mitigate the impact of future failures in order to maximize reliability of operations and improve the business performance. The e-maintenance services interact with e-operations services in order to also consider the global impact of maintenance-related changes to the manufacturing operations, while self-learning mechanisms are applied to all the phases of Proactive Maintenance in order to continuously improve the generated information.

Proactive Maintenance has a managerial and a technological perspective. From a managerial point of view, its implementation requires the identification of the need for a different maintenance strategy through feasibility studies as well as the radical change of maintenance-related business processes and operations in all the enterprise organizational levels (operational, management, strategic). From a technological point of view, it requires appropriate technologies and information systems for effectively supporting the Industry 4.0 principles. Therefore, Proactive Maintenance should include the following characteristics:

- **IoT-based Condition Monitoring**. Condition monitoring is applied with sensors at a component, machine or production process level. The decisions about their type and their distribution (placement) are affected by the manufacturing system examined. These hardware and/or software sensors generate huge amounts of real-time data (big data) in the context of IIoT which are further processed through appropriate infrastructures.
- **Event-Driven Architecture**. Event processing is used to process massive primitive events and get valuable high level information from them by continuously monitoring the event flow. Therefore, through the event triggers, event-driven infrastructures are able to handle big data in a scalable and efficient way.
- **Prognosis Lifecycle**. Prognostic lifecycle covers all the maintenance phases, through which information is processed; from signal processing and diagnostics till prognostics and maintenance decision making along with continuous improvement during actions implementation. Predictions about the future equipment condition, on the basis of which mitigating actions can be applied ahead of time, constitute the backbone of Proactive Maintenance. They can be realized with associated predictive event processing agents.

- **Proactive Computing**. Proactive event processing makes it possible to anticipate potential issues during process execution and thereby enables proactive process management, i.e. to decide and act on the basis of real-time predictions. The proactive event-driven applications are subjected to the proactive principle. A proactive situation includes a future event, a predictive pattern, the probability distribution function of the event occurrence, a list of mitigating actions and costs (e.g. the cost of the future event, the costs of actions as function of implementation time).
- **E-maintenance Support**. The e-maintenance concept is linked to the Proactive Maintenance framework, since it provides the communication and technological background for real-time data processing and information exposure to the users and thus, it can support all the phases of the proactive principle. E-maintenance applications and platforms can facilitate proactivity and further advance to a greater value with the development of Cyber-Physical Systems, while they are able to utilize an event-driven architecture for scalable sensor-generated big data processing.
- **Interaction with other Industrial Operations**. Since, every change in industrial operations affects the others, maintenance operations should be considered along with its interactions with the other operations. A reduction in production, quality and inventory costs is considered as one of the most important indirect benefits of Proactive Maintenance. For instance, due to the available real-time prognostic information, predictive maintenance actions along with quality improvement and production activities can be recommended and spare parts can be ordered just in time.

4.2 The Concept of Proactive Maintenance

Proactive Maintenance strategy implementation requires a complete methodology as well as appropriate information systems capable of processing information captured by sensors in order to provide added value insights by taking advantage of IoT advances for handling failure uncertainty. Failure uncertainty is derived from the stochastic nature of the degradation processes of manufacturing equipment and leads to high uncertainty in decision making (Van Horenbeek et al., 2013; Li et al., 2015).

Although the use of sensors with high monitoring capabilities within the modern enterprises’ network and across different levels is a reality, the strategic value of data analysis should be increased. The current status is the use of sensors and information systems for monitoring various parameters that are known to affect equipment condition from expert knowledge. Although this is valuable information, this approach does not enable the maintenance strategy transformation. Currently, there is still lack of services and tools capable of efficiently processing real-time big data from heterogeneous sources, implementing complex algorithms and provide meaningful insights about potential problems along with a self-improvement approach. Novel IT technologies and e-maintenance systems are still not well perceived by the industry due to the high consulting costs (since vendors are selling closed/proprietary solutions) and the projects’ long duration. Even in the case of open source solutions, the consulting costs are very high and the projects last long.

Proactive Maintenance maximizes the expected utility and exploits the full potential of predictive maintenance management, sensor-generated big data processing, e-maintenance, proactive computing and industrial data analytics. It is able to be applied in the context of the production process of any manufacturing company regardless their processes, products and physical model used. To this end, the concept of Proactive Maintenance converges and synthesizes predictive maintenance, proactive computing, the Gartner’s levels of industrial analytics maturity and the ISO 13374 as implemented to MIMOSA OSA-CBM in order to create a consistent basis for a generic Proactive Maintenance architecture in an IoT-based industrial environment. In this way, the convergence of Operational Technology and Information Technology can also be achieved. Figure 4-1 depicts the relationships among these concepts and their aggregation to SENSE, DETECT, PREDICT, DECIDE, ANALYZE, and FMECA phases.

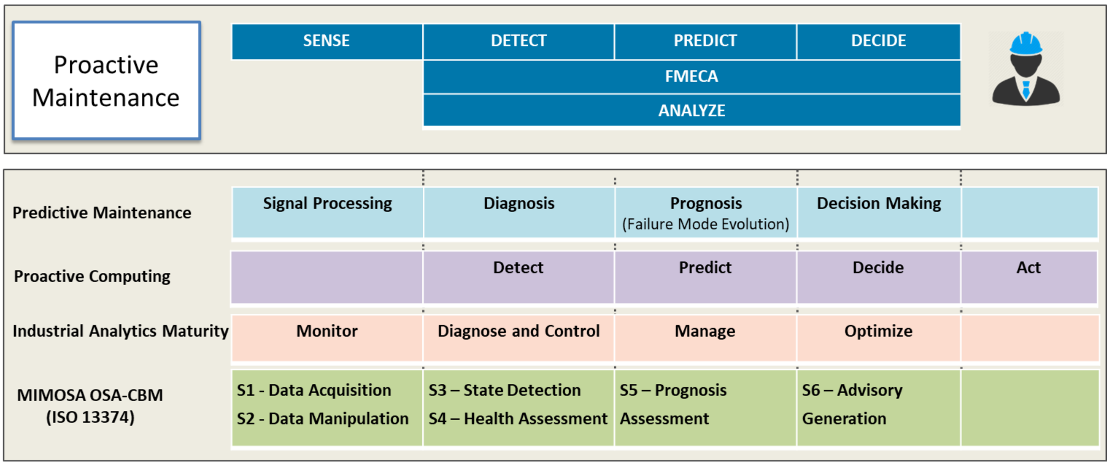

Figure 4-1: The Proactive Maintenance Concept

4.3 The Conceptual Architecture of Proactive Maintenance

The large scale and complexity of modern software projects result in several challenges for software architectural designers. The inception phase of the architecture design introduces several canonical architectural elements providing a basic functional decomposition of the envisaged system (Kruchten, 2004). The goal of Proactive Maintenance software design is to transform the real world problem into software solution using conceptual modelling for accurately describing the real world problem, capturing and facilitating analysis of the system in the context of its environment, incorporating key product features, requirements, and essential domain knowledge, including information about the structural, behavioural, and functional characteristics (Liu, and Gluch, 2004; ISO/IEC/IEEE42010, 2011; Mahmood, and Montagna, 2012; Bernus et al., 2015; Pelliccione et al., 2017).

Due to the increasing number of networked components, a level of complexity has been reached which is difficult to handle using traditional development processes (Fennel et al., 2006). During the last years, there is a paradigm shift from a hardware-, component-driven to a requirement- and function-driven development process, and a stringent standardization of infrastructure elements (Pelliccione et al., 2017) in order also to ensure reusability and knowledge transfer (Gröger, 2018). Architectural practices need to keep a clear focus on scalable and flexible systems architectures providing ubiquitous, interoperable and networked solutions to realise smarter systems, platforms and ICT infrastructures for all entities operating in a common business ecosystem (Romero, and Vernadat, 2016).

In the context of Industry 4.0, standards are essential for ensuring the reliable and efficient interaction of various different systems. DIN, the German Institute for Standardization and its partner institute DKE, has presented an updated Roadmap Industry 4.0[8]. The Roadmap gives an overview of existing standards in this area and identifies the need for new standards along with appropriate recommendations. A major outcome of these processes is the RAMI 4.0 model which is described in DIN SPEC 91345[9]. The Proactive Maintenance architecture was designed according to the Proactive Maintenance concept and the ISO/IEC/IEEE 42010 “System and software engineering – Architecture description” (ISO/IEC/IEEE42010, 2011) which defines the architecture as: *“< system > fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution”*.

In this Section, the proposed Proactive Maintenance conceptual architecture is presented. Since the Proactive Maintenance architecture should be compatible with RAMI 4.0, Section 4.3.2 presents Proactive Maintenance architecture in the frame of RAMI 4.0.

4.3.1 The Proactive Maintenance Architecture

The Proactive Maintenance conceptual architecture forms the basis for the development of a unified information system capable of covering the whole prognostic lifecycle and linking maintenance with other industrial operations, i.e. production, logistics, quality. The Proactive Maintenance system is able to be applied in the context of the production process of any manufacturing company regardless their processes, products and physical model used. It is applicable at the level of component, machine and production system, depending on the placement of sensors

[8] https://www.din.de/blob/65354/f5252239daa596d8c4d1f24b40e4486d/roadmap-i4-0-e-data.pdf

[9] https://www.din.de/en/din-and-our-partners/press/press-releases/updated-german-standardization-roadmap-on-industry-4-0-110576

throughout the production lifecycle and the data availability in the manufacturing company's legacy data systems (e.g. Enterprise Resources Planning- ERP, Manufacturing Execution System- MES). Within the Proactive Maintenance system, there are interactions between the various e-maintenance services and the e-operations data and information from the manufacturing companies' systems in order to synchronise maintenance with production, quality and logistics management. The interrelationship between the e-maintenance and the e-operations services allow the exchange of the appropriate information. The functional / high level view of the conceptual architecture is depicted in Figure 4-2.

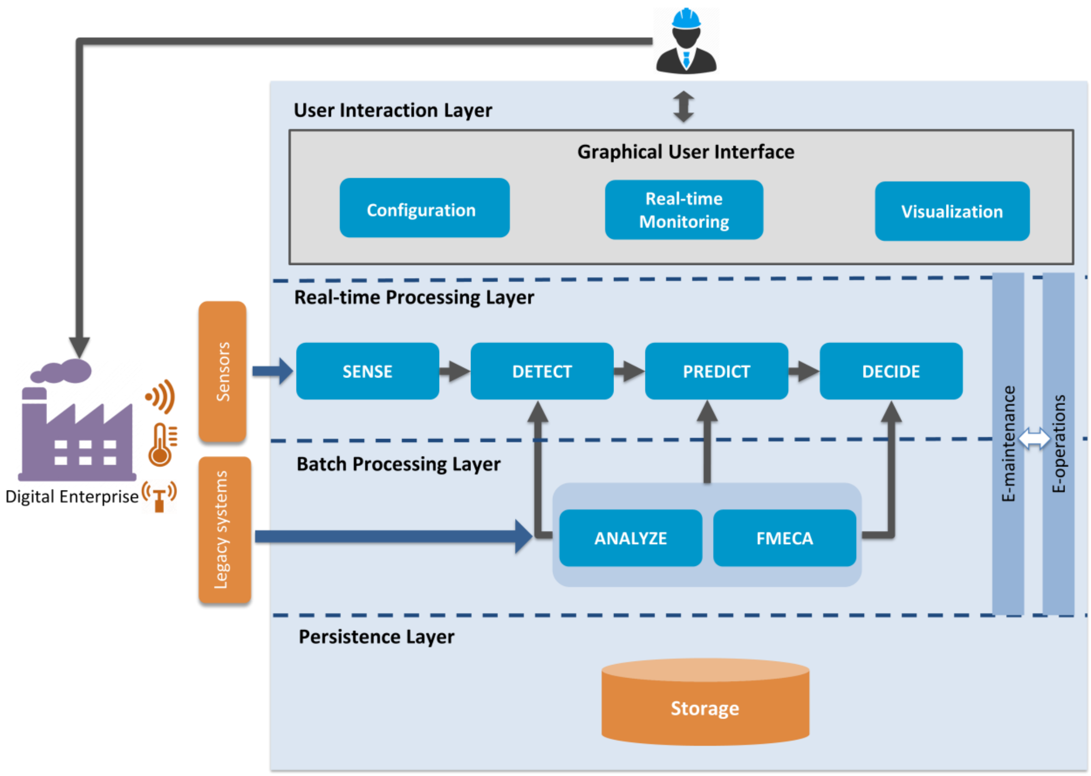

Figure 4-2: The Functional/ High Level View of the Proactive Maintenance Conceptual Architecture

The functional/ high level view of the Proactive Maintenance conceptual architecture consists of five main phases: SENSE, DETECT, PREDICT, DECIDE, ANALYZE, and FMECA (Failure Mode, Effects and Criticality Analysis). It should be noted that FMEA is a bottom-up, inductive analytical method which may be performed at either the functional or piece-part level. FMECA extends FMEA by including a criticality analysis, which is used to chart the probability of failure modes against the severity of their consequences. The result highlights failure modes with relatively high probability and severity of consequences, allowing remedial effort to be directed where it will produce the greatest value. It also consists of four layers: User Interaction Layer, Real-time Processing Layer, Batch Processing Layer, Persistence Layer. In the Real-time Processing Layer and Batch Processing Layer there are interactions between the e-maintenance and the e-operations services. The e-operations services deal with processing and analysis of data, information and knowledge dealing with other manufacturing operations that are closely related to maintenance and affect each other. For instance, logistics management (e.g. spare parts inventory, lead times, etc.), quality management (e.g. scrap rate threshold, defects, etc.) and production planning (e.g. production plan, resource plan, etc.) are key issues to be considered for maintenance planning. On the other way around, these manufacturing operations are influenced by the decisions taken related to maintenance operations. Consequently, the e-maintenance services interact with the e-operations services in order to schedule jointly the maintenance and the production activities along with quality and logistics aspects. The scope of each Layer along with the phases is explained below.

The ***User Interaction Layer*** occupies the top level of the conceptual architecture and is addressed with an integrated Graphical User Interface (GUI). It has three main goals:

- To enable the user configuring the components constituting the architecture (e.g. enabling the embodiment of the appropriate expert knowledge about various input parameters, inserting user preferences, etc.)
- To allow the user monitor real-time information deriving from the respective phases, i.e. the current and the predicted equipment behaviour, warning alerts, the generated recommendations, etc.
- To provide visualization capabilities incorporating appropriate techniques. This phase includes the gathering, storing, analysing and visualizing the results with respect to maintenance-related information in conjunction with production, logistics and quality issues. This information is exposed to the user using line diagrams, histograms, pie-charts, heat maps, relationships, geo maps, etc. supporting various strategies for problem investigation, e.g. drill-

down to identify root causes, generalisation to find similar occurrences in the past and to develop improvement measures.

At the ***Real-time Processing layer***, the respective phases are executed based on the sensor-generated real-time data. The phases that are associated to the Real-time Processing Layer are: SENSE, DETECT, PREDICT, DECIDE. Each phase is triggered by the previous one, taking also into account relevant data and information regarding logistics management, quality and production planning. The scope of each phase in this Layer is described below:

- **SENSE**: This phase deals with the collection, aggregation and manipulation of sensor data. The large amounts of enterprise data as well as the existing information and knowledge come from various heterogeneous data sources dealing with different manufacturing operations. The SENSE phase handles these data and processes them to the subsequent phases in the form of sensor data streams for further analysis and processing.
- **DETECT**: This phase deals with the real-time state detection and health assessment of a whole system or components with respect to a mechanical system in order to provide a diagnostic output. The analysis is carried out by different algorithms. Therefore, there is a library with typical algorithms for data analytics easily extensible with new algorithms. The results of each analysis can be weighted in order to estimate the current condition of the analysed component. This integrative approach for the state determination of complex technical systems recognizes the presence of an unusual (and potentially hazardous) state within the behaviours or activities of a system (e.g. measuring indicators of degradation), with respect to some model of 'normal' behaviour. To do this, it requires the last updated model of normal behaviour, the normal operation threshold, etc. The diagnostic models continuously learn from the actual equipment behaviour by updating and improving the incorporated diagnostic models with data analytics and FMECA information.
- **PREDICT**: This phase includes real-time state prediction of a whole system or components with respect to the mechanical system, e.g. prediction about the time-to-failure, the probability distribution function of the failure occurrence,

the Remaining Useful Life (RUL), the Remaining Life Distribution (RLD). The failure prediction constitutes the backbone of predictive maintenance. The analysis is carried out by different prognostic algorithms through the definition of calculation flows and their instantiation to machines, components or sites. These algorithms are executed considering constraints deriving from quality management, logistics management and production planning (e.g. acceptable scrap rate thresholds in case of a machine malfunction) as well as results of historical data analytics (e.g. for creating the model) and FMECA. The algorithms and the prognostic models can be then continuously.

- **DECIDE**: This phase includes decision making algorithms for providing recommendations ahead of time (i.e. in a proactive manner) taking into account the real-time prognostic information and information/ knowledge deriving from users (e.g. maintenance engineers) or from further data analysis. On the basis of the real-time prognostic information, the optimal mitigating maintenance actions and the optimal times for their implementation are recommended considering both perfect and imperfect maintenance actions with various degrees. This phase also takes into account logistics, production and quality issues. In this way, maintenance decision making can be shifted from expert knowledge and/ or early warnings into business performance optimization. The decision models are continuously updated based on data analytics and FMECA results (e.g. with the updated risk ranking).

At the ***Batch Processing Layer***, the respective phases are executed based on the legacy data and the updated existing information regarding e-maintenance and e-operations. The phases that are associated to the Real-time Processing Layer are: ANALYZE, FMECA. The scope of each phase of the Batch Processing Layer is described below:

- **ANALYZE**: This phase is triggered on a batch mode and aims to conduct analysis on legacy and historical data in order to find useful information and to produce rules, regarding downtimes and failures of a production line or other information related to other manufacturing operations. The exploitation of the legacy and historical data can lead to finding patterns and clustering/classifying failures based on similar characteristics. Moreover, the analysis is conducted on the basis of actions that mitigated the impact of a failure or of failures that actually occurred before the recommended actions implementation. To this end, this phase is able to feed into the FMECA, the PREDICT and the DECIDE phases so that they provide more informed results taking into account the knowledge extracted from the legacy data systems.

- **FMECA**: This phase includes an FMECA mechanism which incorporates algorithms for the identification of potentially relevant and critical failures modes and conducts analysis of the criticalities that might arise. It foresees the development of dedicated automatized processes in order to enable, on the basis of failures modes, critical elements, process, logistics and production data what are the most likely effects and what are their implications in terms of maintenance and operations management. The resulting failure modes, frequencies, risks and other associated results feed into the DETECT, PREDICT and DECIDE phases in order to update the respective algorithms and models.

The ***Persistence Layer*** of the conceptual architecture aims to support the functionalities implemented at the Real-time Processing and Batch Processing Layers as well as at the User Interaction Layer of the architecture. It includes a Database Abstraction Layer (DAL) and houses a relational database engine where all information needed by the other three layers is stored and retrieved. For the raw sensor data itself, this data storage concept is enhanced by a database for time-series to ensure efficient and reliable storage of this data, while visualization functionalities will use an indexing database to facilitate the exposure of analytics.

The functional/ high level view of the conceptual architecture represents the Proactive Maintenance flow in the sense that the output of one phase triggers the next subsequent phase.

Table 4-1 shows the potential inputs and outputs of each phase based on the functional/ high level view of the Proactive Maintenance conceptual architecture. It also takes into account relevant literature about condition-based, predictive maintenance.

Table 4-1: Inputs and Outputs of each UPTIME phase

Proactive Maintenance phases	Input	Output
SENSE	• ***Data demands/requirements*** (e.g. information about the data to be collected and required pre-processing) • ***Raw sensor data*** (e.g. about measured parameters used as indicators of degradation)	• ***Data set / time series*** (e.g. sampled sensor data) • **Sensor data streams**
DETECT	• ***Sensor data set*** (e.g. about measured parameters used as indicators of degradation) • ***Historical data and domain knowledge*** (e.g. thresholds indicating a dangerous state)	• ***Current health state*** • ***Alert of a potentially dangerous state***
PREDICT	• ***Current health state*** • ***Alert of a potentially dangerous state*** • ***Sensor data set / time series*** • ***Historical data and domain knowledge*** (e.g. about the measured parameter used as indicator of degradation till failure)	• ***Early warning*** *OR* • ***RUL/ RLD and confidence level*** *OR* • ***Probability distribution of the failure occurrence***
DECIDE	• ***RUL/ RLD/ time-to-failure and confidence level*** *OR* • ***Probability distribution of the failure occurrence*** • ***Failure prevention and mitigation measures*** • ***Risk Ranking*** • ***Results of legacy data analytics***	• ***Early notification*** *OR* • ***Recommendations about:*** ▪ ***Optimal actions*** ▪ ***Optimal time for implementation***
FMECA	• ***Rules, patterns and failures classification*** • ***Legacy data*** on maintenance (e.g. machinery failures, maintenance actions) • ***Threshold parameters***	• ***Reliability Block Diagrams*** • ***Parameter threshold*** • ***Risk ranking*** • ***Failure prevention and mitigation measures***
ANALYZE	• ***Legacy data*** on maintenance (e.g. machinery failures, maintenance actions, data from the production plan to be aligned with maintenance) • ***Historical data*** on machinery maintenance and operations	• ***Rules, patterns and failures classification*** • ***Constraints and additional operational information***

4.3.2 Mapping Proactive Maintenance Conceptual Architecture to RAMI 4.0

In this Section, the Proactive Maintenance conceptual architecture in the frame of RAMI 4.0 is presented. In Proactive Maintenance, the RAMI 4.0 is adopted to communicate the scope and design of the system, to further collaboration and integration with other relevant initiatives by framing the developed concepts and technologies in a common model. In this sense, the Proactive Maintenance conceptual architecture is compatible with RAMI 4.0 facilitating maintenance implementation in the frame of Industry 4.0. In this way, RAMI 4.0 is instantiated to maintenance operations. This is a challenging task since the Industry 4.0 paradigm is still evolving with limited past experience of successful implementations.

RAMI 4.0 is a three-dimensional model representing different interconnected features of the technical – economical properties and showing how to approach the issue of Industry 4.0 in a structured manner. It consists of three axes: (i) the hierarchy levels; (ii) the architecture layers; and, (iii) the lifecycle value stream. The following sub-sections present these axes in the context of Proactive Maintenance. The need for sub-models for individual aspects and processes in RAMI 4.0 has been recognized as a crucial next step for its further evolution (Hankel, and Rexroth, 2015). Mapping of the Proactive Maintenance conceptual architecture to RAMI 4.0 enables the integration of the maintenance process with the other operations and processes of the manufacturing enterprise based upon the Industry 4.0 paradigm. An I4.0 component is a crucial aspect of Industry 4.0. It deals with the digitization of assets in the manufacturing process. It consists of one or more assets and an administrative shell. The administrative shell is the virtual representation of an asset. The I4.0 component is located within the layers of RAMI 4.0, up to the Functional Layer. It can adopt various positions in the life cycle and value stream, and occupy various hierarchical levels. The following sub-sections describe the three axes of RAMI 4.0 in the context of Proactive Maintenance.

4.3.2.1 Hierarchy Levels

Industry 4.0 brings changes in the architecture of the classical control pyramid of production complexes as well technological processes. Industry 4.0 architecture of hierarchical level shows a functional assignment of components (Zezulka et al., 2016). This axis within an enterprise or factory follows the IEC 62264 and IEC 61512 standards. The level over and below the IEC standards area represents steps further and describes also groups of factories, collaboration within external engineering firms, component suppliers and customers. Therefore, the hierarchy levels are: product, field device, control device, station, work centre, enterprise, and connected world. Proactive Maintenance architecture is applicable at a component, machine or production process level. In this sense, it can be implemented in flexible smart systems and machines capable of interacting and communicating across the hierarchy levels through a network. The implementation of a Proactive Maintenance system in a "Connected World" (i.e. connected factories with integrated maintenance processes) would require its use by all of them in order to create synergies (e.g. between a factory and its supplier of maintenance spare parts).

4.3.2.2 Architecture Layers

Figure 4-3 shows the Proactive Maintenance conceptual architecture in the frame of the RAMI 4.0 architecture layers. The individual layers and their interrelationships are described below.

Asset Layer: Since this layer represents the reality ("physical things in the real world"), production equipment and users are part of it. Proactive Maintenance is implemented on the production equipment with the involvement of Proactive Maintenance system users. The production equipment can be further analysed.

Integration Layer: This layer makes provision of information on the assets in a form which is available for computer processing by connecting elements as well as people with IT. In the context of Proactive Maintenance, this layer involves the equipment-installed sensors, the actuators and the legacy data systems (MES, ERP, etc.). It also includes the Human Machine Interfaces of the legacy data systems (e.g. ERP GUI) through which the users insert data.

Communication Layer: Since this layer provides standardization of communication by means of uniform data format, it includes the IoT Gateway, the Broker and the Legacy Data Uplifting. In this way, the Proactive Maintenance solution will gather the data from the information sources for further processing in the subsequent Information Layer.

Information Layer: This layer provides pre-processing of events and execution of event-related rules by enabling their formal description. It also manages data persistence, ensures consistent data integrity and transformation for feeding into the

Functional Layer. In the context of Proactive Maintenance, it includes sensor and legacy data pre-processing, which correspond to functionalities of data preparation and pre-processing while feeding into the Real-time (Stream) Processing and the Batch Processing environment respectively. To this end, this layer also includes the Real-time Processing and the Batch Processing environment as well as the Storage, i.e. the appropriate databases. In this way, the required data are extracted and combined accordingly in order to be available by the functions of the next layer.

Functional Layer: This layer enables the formal description of functions and creates platform for horizontal integration of various functions. It contains run time and modelling environment for services supporting the business processes and a run time environment for applications and technical functionalities. In this layer, the following functions take place: Detection (which includes the real-time detection/ diagnostic algorithms), Prediction (which includes the real-time prediction/ prognostic algorithms), Proactive Decision Making (along with its feedback functionalities) and FMECA as well as the legacy data analysis that aim to update and improve the modelling and the parameter values of predictive maintenance core functions (i.e. diagnosis, prognosis, proactive decision making, FMECA). The aforementioned functions are executed on the basis of data integrity of the previous layer.

Business Layer: This layer ensures the integrity of functions in the value stream and enables mapping business models and the resulting of the overall process. It takes into account the policies, rules and constraints according to which the system operates through the interrelationships of maintenance to other manufacturing operations. It also creates a link among different business processes through the exposure of appropriate information to the user. In this sense, this layer involves the Configuration of the Proactive Maintenance solution, the Real-time Monitoring of Proactive Maintenance functions and the Visualization of its results. It also incorporates its interrelationships with other business processes and the integration with manufacturing operational functions interrelated with maintenance (e.g. logistics management, quality management, production planning).

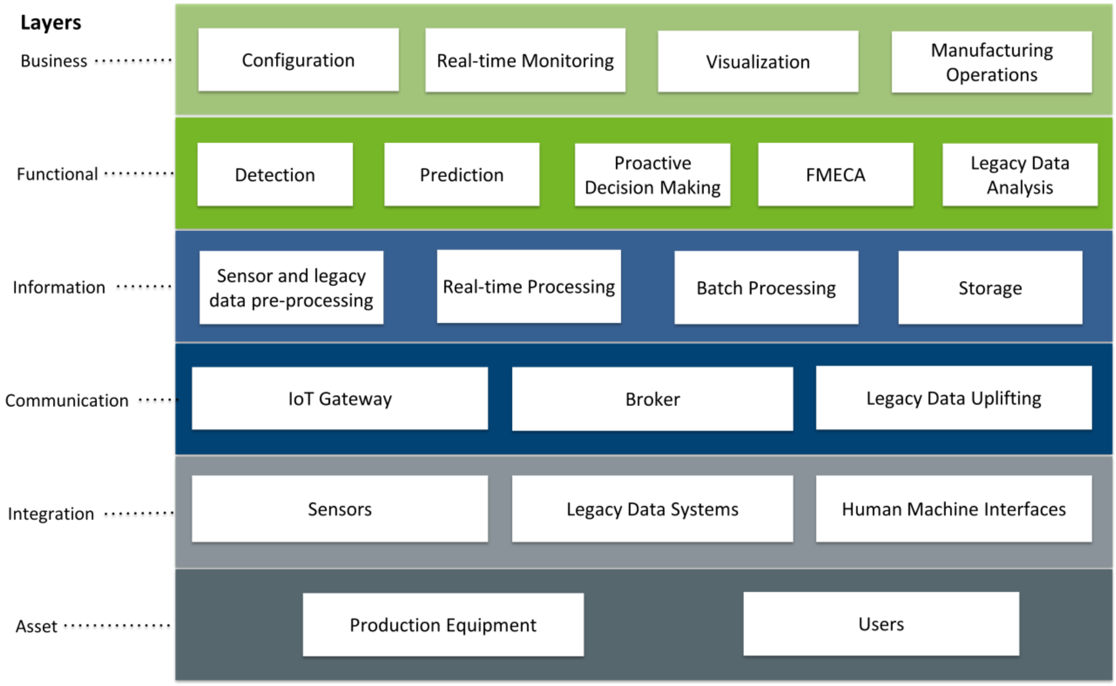

Figure 4-3: Mapping the Proactive Maintenance conceptual architecture to RAMI 4.0

4.3.2.3 Lifecycle Value Stream

The lifecycle value stream axis is divided to Type and Instance. The Type is divided to Development and Maintenance / Usage, while the Instance is divided to Production and Maintenance / Usage (Platform Industrie 4.0). A type represents the initial idea, while each manufactured product represents an instance of that type (Platform Industrie 4.0). The instances are sold and delivered to customers. The change from type to instance may be repeated many times (Platform Industrie 4.0). Feedback from customers to instances of the type may lead to corrections. Such modifications deal with the type, i.e. they are applied as amendments to the type documentation and new instances of the modified type are produced. The value stream in the totally digitized production can be viewed in conjunction with value-adding processes, since it enables linking of purchasing, production planning, logistics, quality, customers and suppliers (Platform Industrie 4.0).

The lifecycle value stream of Proactive Maintenance has both managerial and technical implications. As far as the managerial perspective is concerned, the type includes the idea as well as the development and validation of a Proactive Maintenance strategy. After successful validation, the new consulting service is released.

Each instantiation of the Proactive Maintenance strategy to a specific production process or industry represents an instance of that type. As far as the technical perspective is concerned, the type includes the idea as well as the development and testing of a prototype for Proactive Maintenance which set the basis for serial production. Each instantiation of the system to a specific equipment, production process or industry, and to a specific legacy data system or installed sensor represents an instance of that type.

5 Proactive Decision Making in Maintenance Management

In this Chapter, the proactive approach in decision making is presented. Based on this approach, proactive event-driven decision methods are developed in order to provide recommendations for maintenance and maintenance-driven operations. Automation of proactive maintenance decisions on the basis of real-time sensor-driven prognostic information is an unexplored area.

5.1 Introduction and Motivation

Despite the significance of proactive maintenance decisions (Gupta & Lawsirirat, 2006; Campos, 2009; Ahmad, and Kamaruddin, 2012; Guillen et al., 2016), their automation by providing system-generated recommendations in a real-time, event- driven environment remains a challenge (Van Horenbeek, et al., 2013; Aboelmaged, 2015). Existing works regarding maintenance applications have usually the following limitations: (i) they provide only a diagnostic or a prognostic output; (ii) they rely on processing of batches of data and not on real-time, event-driven information; (iii) when they provide recommendations, they deal with immediate action implementation, something that does not lead to an optimized performance (because the expected loss may be minimized some time into the future, before the occurrence of the undesired event); (iv) they rarely are integrated to algorithms addressing other operational issues driven by maintenance (e.g. inventory, supplier selection). In addition, several research works in proactive computing have only been described conceptually and have not been embedded in a real-time, event-driven environment.

In the following sub-sections, the approach of proactive decision making and its instantiation to maintenance operations is described. The approach and the decision methods address two blocks of the conceptual architecture for the Decide phase the “DMI Configuration” (as far as the decision methods configuration is concerned) block of the User Interaction Layerand the “Proactive Decision Methods” block of the Real-time Processing Layer. These two blocks are highlighted with red color in the conceptual architecture in Figure 5-1.

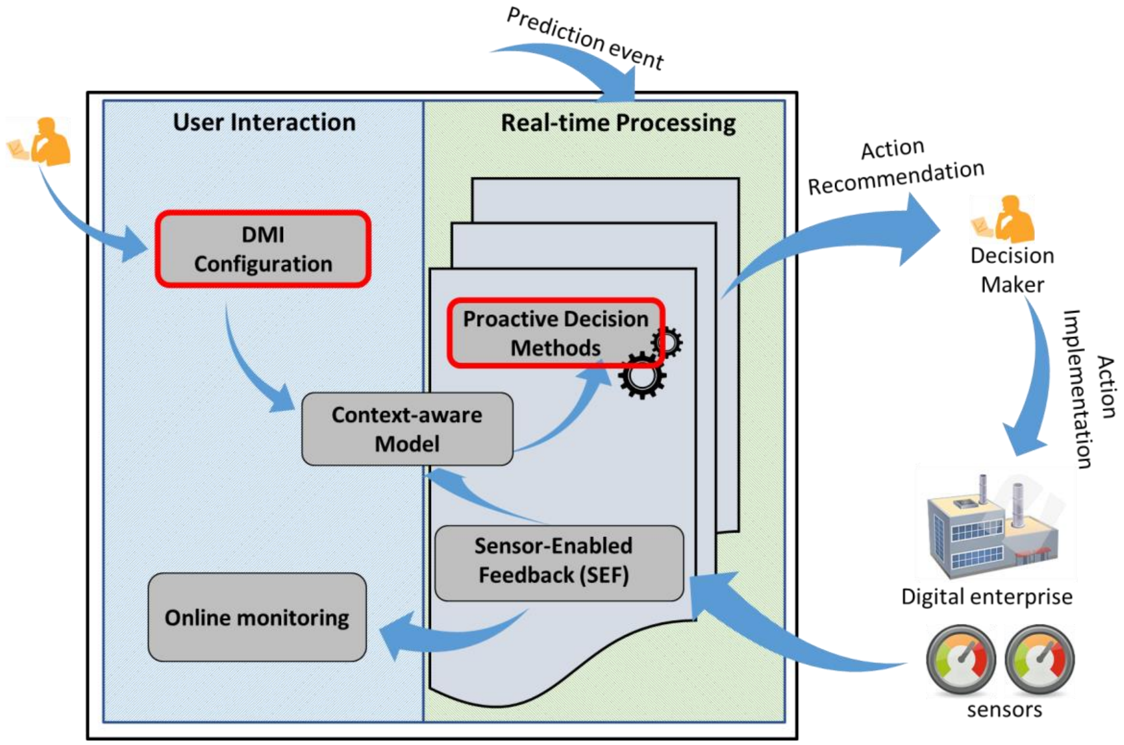

Figure 5-1: The functionalities for proactive decision making in the conceptual architecture.

5.2 The Proactive Approach in Decision Making

The proposed approach focuses on enabling decision makers to create new decision method instances addressing the problem at hand, e.g., the maintenance needs of specific manufacturing equipment. Therefore, they are able to configure them by editing the domain knowledge that is required by the method. This domain knowledge may include a list of actions, the cost of the undesired event, the costs of actions, etc. Decision method instances are specific instances of decision methods, corresponding to specific equipment or other subject of a predicted undesired event which triggers, during runtime, the decision method for mitigating it. Decision methods are then enacted online in order to generate timely and reliable proactive recommendations based on the analysis of the streaming data and the derived predictions for undesirable situations, i.e. events which lay outside the desired states space. Proactive recommendations deal with the optimal maintenance actions and the optimal time for their implementation. The recommendations are actually generated on the basis of the utility or the loss prediction (expected utility / loss) in the course of time until the decision horizon. Therefore, the aim is to apply an action at a time that maximizes the expected utility (or minimizes the expected loss).

Figure 5-2 depicts the conceptual approach for proactive decision making. The expected loss due to maintenance is represented as a function of the implementation time. Therefore, t = 0 is the time when a prediction is received and a recommendation is provided. The expected loss functions are optimized within the boundaries of t = 0 and the time of the end of decision horizon (e.g. next planned maintenance). For example, for three alternative actions, the optimal action is Action 2 and the optimal time for its implementation is the time when the expected loss is minimized. In this way, the user gets a recommendation that minimizes the expected loss and provides them a time window until the recommended implementation time in order to be prepared appropriately. With the proposed approach, decision making can be shifted from early warnings into business performance optimization.

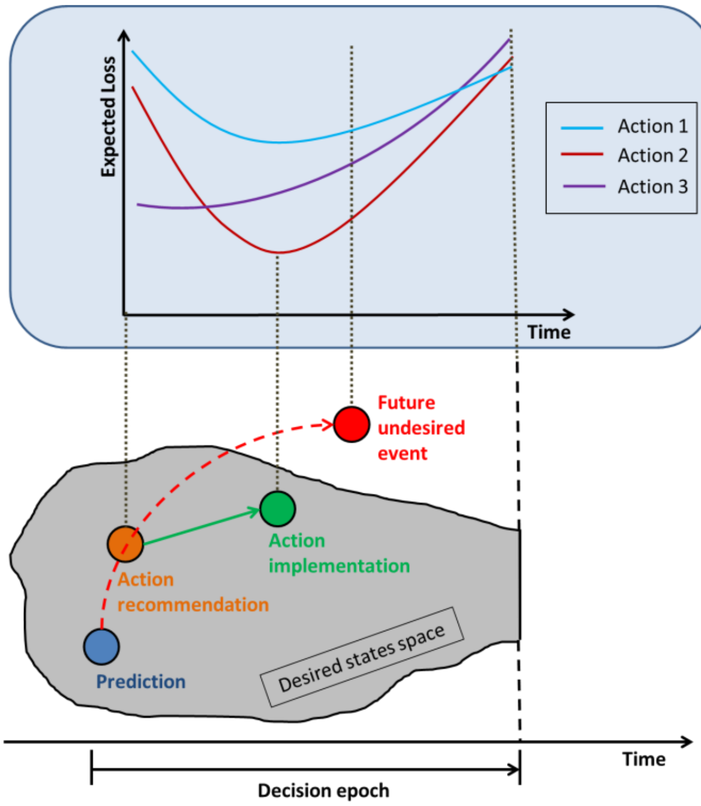

Figure 5-2: Conceptual approach for proactive decision making (Adapted from Feldman et al., 2013)

5.3 Overview of the Proposed Proactive Decision Methods

In this PhD thesis, there have been developed two proactive event-driven methods for proactive decision making for maintenance actions, two for joint proactive maintenance and logistics optimization and one for proactive selection of maintenance spare parts suppliers. These methods are presented in the following Sections. The aforementioned proactive event-driven decision methods provide a different output recommendation and each one of them requires a different input. The input is taken by two sources: events generated by the Predict phase and users through a decision configurator dashboard. This information is shown in Table 5-1, Table 5-2 and Table 5-3 and is further described in the following sections. The optimization of the resulting functions is conducted by using the Brent's method which is is a root-finding algorithm combining the bisection method, the secant method and inverse quadratic interpolation (Gegenfurtner, 1992).

Table 5-1: Input and output of proactive decision making for maintenance actions

<table>
<tr><th>Decision Method</th><th>Input from user</th><th>Input from events</th><th>Output</th></tr>
<tr><td>Proactive Expected Loss Rate Optimization</td><td>• Corrective action cost function
• Planned action cost function
• Planned time for action implementation</td><td rowspan="2">• Probability Distribution of the occurrence of the event
• Parameters of the probability distribution (e.g. λ for exponential)</td><td>• Optimal time for the predefined action implementation</td></tr>
<tr><td>Proactive Markov Decision Process</td><td>• List of maintenance actions
• Action cost functions
• Time-to-undesired event after action
• Delays
• Cost of undesired event (e.g. failure)
• Decision horizon</td><td>• Optimal action
• Optimal implementation time</td></tr>
</table>

Table 5-2: Input and output of proactive decision making for joint maintenance and logistics actions

Decision Method	Input from user	Input from events	Output
Proactive joint replacement and spare parts inventory decision model	• Cost of undesired event • Planned action cost function • Planned time for action implementation (e.g. planned maintenance) • Shortage inventory cost as a function of time • Holding inventory cost as a function of time • Lead time between the time of placing the order up and the time of receiving the order	• Probability Distribution of the occurrence of the event • Parameters of the probability distribution (e.g. λ for exp.)	• Optimal maintenance (mitigating) action • Optimal implementation time of the maintenance action • Optimal spare parts ordering (prerequisite) action • Optimal time of ordring
Proactive joint maintenance and spare parts inventory decision model	• Cost of failure • Cost function of the action effect • Cost function of the action implementation • Cost of buying the spare parts • Cost function of shortage inventory • Lead time between the time of placing the order up and the time of receiving the order • Decision horizon		• Optimal time for maintenance (mitigating) action implementation • Optimal time for spare parts ordering (prerequisite) action implementation

Table 5-3: Input and output of proactive decision making for supplier selection

Decision Method	Input from user	Input from events	Output
Proactive selection of maintenance spare parts suppliers	• Available budget • Number of suppliers • Historical data about portfolios of suppliers	• Prices prediction till next planned maintenance, • Recommended time for ordering	• Markowitz bullet' and its 'efficient frontier' • The optimal portfolio of suppliers

The proposed decision methods are based on failure probability predictions and thus, on reliability analysis. In this sense, they are triggered by probability distribution functions of failure occurrence that have been derived from degradation modelling techniques. Based on the terminology of reliability analysis, an event density function of $\boldsymbol{\varepsilon}$, denoted by $\boldsymbol{g^{\varepsilon}(t)}$, indicates the probability that $\boldsymbol{\varepsilon}$ will occur at time t. The cumulative distribution function of g is denoted by $\boldsymbol{G^{\varepsilon}(t)}$, and is called the lifetime distribution function of $\boldsymbol{\varepsilon}$. $\boldsymbol{G^{\varepsilon}(t)}$ indicates the probability that $\boldsymbol{\varepsilon}$ will occur between time zero and time t (Engel et al., 2012; Kapur, and Pecht, 2014), while $\boldsymbol{\overline{G}^{\varepsilon}(t) = 1 - G^{\varepsilon}(t)}$ denotes the cumulative probability distribution function of the undesired event not occurring. When an action $\boldsymbol{a}$ is applied to reduce the probability of an undesired event, $\boldsymbol{a}$ is associated with a new event density function $\boldsymbol{g_a^{\varepsilon}(t)}$, which is the probability that $\boldsymbol{\varepsilon}$ occurs at time t, although $\boldsymbol{a}$ has been applied before t. This happens because the implementation of action $\boldsymbol{a}$ does not prevent $\boldsymbol{\varepsilon}$ with certainty (Engel et al., 2012). Therefore, the probability distributions are calculated as shown in Equation 5-1 and Equation 5-2. In Equation 5-2, the conditioning (denominator) takes into account the fact that until the action occurrence at $\boldsymbol{t_1}$, the distribution in place was $\boldsymbol{G^{\varepsilon}}$ (Engel et al., 2012). For example, in case of exponential distribution, time-to-breakdown = 1 / λ, where λ is the parameter of exponential distribution.

Equation 5-1

$$P^{\varepsilon}(t_1, t_2) = \frac{G^{\varepsilon}(t_2) - G^{\varepsilon}(t_1)}{1 - G^{\varepsilon}(t_1)}$$

Equation 5-2

$$P_{a_i}^{\varepsilon}(t_1, t_2) = \frac{G_{a_i}^{\varepsilon}(t_2) - G_{a_i}^{\varepsilon}(t_1)}{1 - G^{\varepsilon}(t_1)}$$

$P^{\varepsilon}(t_1, t_2)$ denotes the probability distribution function of the occurrence of the undesired event in the time interval (t_1, t_2), conditioned on not occurring until time t_1, while $P_{a_i}^{\varepsilon}(t_1, t_2)$ denotes the probability distribution function of the occurrence of the undesired event in the time interval (t_1, t_2) conditioned on not occurring until time t_1 and assuming that the action a has been implemented exactly at time t_1. This happens because the implementation of action a does not prevent ε with certainty. Moreover, $\bar{P}^{\varepsilon}(t_1, t_2)$ denotes the probability distribution function that the undesired event ε does not occur within the time interval (t_1, t_2)

conditioned on not occurring until time t_1. In the following sections, we present the mathematical formulation of the three aforementioned decision methods.

5.4 Proactive Decision Making for Maintenance Actions

5.4.1 Motivation

Advances in maintenance management methods have led to the transformation of the traditional ‘‘fail and fix’’ practices into the ‘‘predict and prevent’’ ones (Muller et al., 2008b). New practices put failure prediction at the backbone of maintenance management (Ahmad, and Kamaruddin, 2012). Maintenance management can take advantage of the recent advancements in proactive event-driven computing, for fully exploiting its capabilities and supporting proactive decisions, ahead of time.

Next generation of maintenance management will incorporate event stream processing and advanced computation capabilities enabling generation of recommendations supporting proactive decisions. Despite the significance of proactive maintenance decisions (Gupta & Lawsirirat, 2006; Campos, 2009; Ahmad, and Kamaruddin, 2012), their automation by providing system-generated recommendations in a real-time, event- driven environment has not been realized yet. Therefore, developing methods and information systems that provides automated decision making ahead of time on the basis of predictions, capable of processing data generated by sensors and able to be deployed in a real industrial environment remains a challenge (Ahmad, and Kamaruddin, 2012; Van Horenbeek, and Pintelon, 2015; Aboelmaged, 2015).

5.4.2 State-of-the-Art Analysis

Despite the high amount of works regarding maintenance decision making algorithms, there are still several aspects that concern both industry and academia. For example, it is difficult for manufacturing companies to deploy and adapt the decision making algorithms existing in literature to their own specific business context, physical models and data availability (Ruschel et al., 2017). This fact becomes even more important in the context of the Industry 4.0 paradigm and modern big data technologies.

Consequently, manufacturing companies are slowly adopting novel technologies and information systems for improving their maintenance operations, while technology providers are used to provide solutions with limited capabilities (e.g. monitoring of parameters, domain-specific diagnostic and prognostic algorithms). Moreover, existing algorithms and information systems for maintenance decision making have a loose integration with predictive analytics algorithms generating prognostic information. The common practice is to utilize the current level of degradation that is derived from the analysis of the indicators measured by sensors along with expert knowledge.

Existing works rely on processing of batches of data at specific sampling times (Iung et al., 2009; Peng et al., 2010; Julka et al., 2011). This inhibits the responsiveness of the system to provide event-driven prognostic information and thereafter provide on-the-fly decision making for maintenance. To this end, there is the need for a shift towards scalable and efficient (near) real-time decision making algorithms. This aspect has both a technological (use of appropriate technologies, e.g. for streaming data) and a functional (use of appropriate decision models triggered only when there are predictions about future failures, e.g. recursive and computationally efficient) perspective.

There is a gap in literature regarding generic decision models representing the decision making process instead of the physical process. Moreover, there is a gap regarding the use of probabilistic methods in a streaming context with the aim to tackle with uncertainty. Finally, a remaining challenge is the lack of methods and algorithms capable of recommending optimized actions at optimized times for both perfect and imperfect (of various degrees) maintenance.

5.4.3 The Proposed Decision Methods

5.4.3.1 Proactive Expected Loss Rate Optimization

Quantitative risk analysis seeks to numerically assess probabilities for the potential consequences of risk, and is often called probabilistic risk analysis or probabilistic risk assessment. Risk analysis is a technique for identifying, characterizing, quantifying, and evaluating the loss from an event. Cost risk analysis or cost uncertainty analysis is an important aspect of cost estimation. Cost risk is defined as the probability of the occurrence of an event multiplied by its impact in cost (Arunraj, and Maiti, 2007). A cost risk function is calculated by adding the products of each alternative value i of the cost function with the probability of having this cost function.

The Proactive Expected Loss Rate Optimization (ELR) method estimates the expected loss rate (expected loss per unit of time) of a single pre-defined action each time a prediction event triggers Decide phase. It recommends the optimal time for the pre-defined maintenance action implementation. This method is based upon cost risk analysis (combined with reliability analysis due to the utilization of failure probability) which is defined as the probability of the occurrence of an event multiplied by its impact in cost (Arunraj and Maiti, 2007). Equation 5-3 shows the calculation of the ELR.

Equation 5-3

$$ELR(t) \ = \frac{C_{ue}(t) * G^{\varepsilon}(t)}{t_{ue}} + \frac{C_{pa}(t) * \bar{G}^{\varepsilon}(t)}{t_{pa}}$$

Where: $C_{ue}(t)$ is the cost function of the undesired event, $C_{pa}(t)$ is the cost function of a planned action implemented at a pre-defined time, t_{pa} is the pre-defined time when a planned action is implemented, t_{ue} is the most probable value of the time-to-undesired event distribution function (e.g. in case of exponential distribution, it is equal to 1 / λ).

The loss rate is defined as the loss per unit of time. So, the expected loss rate is the addition of the existing expected loss rates multiplied by their associated cumulative probability distribution functions. In this case, the probability distribution function of the occurrence of the undesired event can be of arbitrary distribution. Previous works have assumed static theoretical probability distributions (Vanneste, and Van Wassenhove, 1995) or batches of data that update the decision module at a specific sampling time (Elwany, and Gebraeel, 2008) or by configuring a pre-defined set of possible times of the undesired event (Wu et al., 2007), while assuming constant costs throughout time. However, these ways are not applicable to a streaming data processing environment where sensors gather data in a very high frequency and costs of actions may vary according to the implementation time. Hence, a modification of the method in accordance to principles of proactive event-driven computing theory (e.g. use of cost functions), as shown in Equation 5-3, is required so that it can be embedded in a streaming architecture and therefore, the decision method is enacted when a prediction event about an undesired event is received.

5.4.3.2 Proactive Markov Decision Process

In Markov Decision Process (MDP), a policy is evaluated according to its expected utility: the expectation on the value of the random variable defined as the sum of rewards obtained by using the policy (Puterman, 2014). This expectation can be defined recursively using the Bellman equation. The classic MDP model assumes a discrete time model, where state transitions can be taken in fixed time steps and its solution is a policy that indicates the optimal action in each state according to the Bellman’s equation. This policy is evaluated according to its expected utility (or expected loss). In this setting, deciding when to take the action is a crucial part of the decision (Engel et al., 2011). Thus, the decision algorithm that uses MDP can provide proactive recommendations and support decisions about when to take which action (Engel et al., 2012).

This can be done by considering the transition probability distributions as a function of time and thus, the expected utility functions can be optimized in order to find out which action has the maximum expected utility (or equally, the minimum expected loss) and at which time. The MDP states and their transitions are formulated according to the proactive computing model and the expected utility of each action is estimated by using the backward induction algorithm (Engel et al., 2012). For example, assuming that there are three possible actions and each one of them needs a delay δi from its implementation until it starts taking effect, the MDP model is formulated as shown in Figure 5-3 (Engel et al., 2012). Based on this formulation, the MDP is solved using backwards induction for finite horizon problems (Engel et al., 2012).

Each state of the MDP proactive model corresponds to a reward which is derived from the cost of undesired event and the costs of actions. Costs of actions can have either a fixed value or a value as a function of implementation time, since the cost of taking an action changes in the course of time according to the time of its implementation. The costs are inserted by the user along with the delays, the effect of each action (how much time each action prolongs the lifetime of the equipment) as well as the end of the decision epoch (the time at which there is no need for a decision any more – e.g. time of planned maintenance).

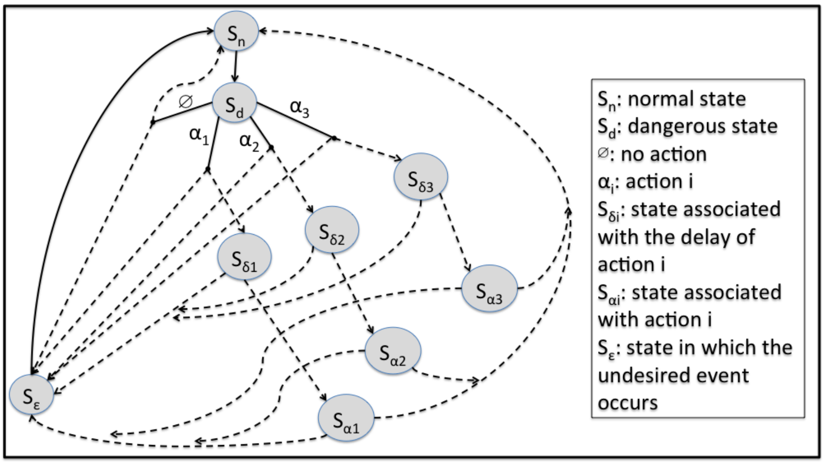

Figure 5-3: MDP formulation for proactive decision making of maintenance actions

The algorithm provides as output the optimal action and the optimal time of applying this action by conducting optimization of the expected utility functions of the alternative actions. This model is able to consider both perfect and imperfect maintenance actions with various degrees. In this method, the probability distribution function of the occurrence of the undesired event must belong to the exponential family, due to the Markov property. In the manufacturing domain, and more specifically in machine prognostics, the probability distribution functions of the occurrence of the undesired event depend on the degradation modelling until the breakdown. Degradation modelling usually follows an exponential, a gamma or a Weibull distribution (Elwany, and Gebraeel, 2008; Kapur, and Pecht, 2014; Guillén et al., 2016). However, in some cases where the rate of degradation is not significantly affected by the cumulative damage, the linear degradation model can be used (Elwany, and Gebraeel, 2008) and therefore, the Proactive MDP method is not applicable.

There is no cost (or benefit) of being at state Sn, hence $U(S_n) = 0$. In state ε, there is a penalty of C_ε (i.e. the cost of failure), hence $U(S_\varepsilon) = C_\varepsilon$. In state ai we incur penalty of $C_{a_i}(t_{a_i})$ (i.e. the cost function of the action) and, given the probability to move to state ε, the policy evaluation gives:

$$U(S_{a_i}) = C_a(t_{a_i}) + P(S_{a_i}, S_\varepsilon) * U(S_\varepsilon) = C_{a_i}(t_{a_i}) + P(S_{a_i}, S_\varepsilon) * C_\varepsilon$$

In state δ_i, there is a penalty of $C_{\delta_i}(t_{\delta_i})$ (i.e. the cost function of the action implementation) and given the probability to move to state f the policy evaluation gives:

$$U(S_{\delta_i}) = C_{\delta_i}(t_{\delta_i}) + P(S_{\delta_i}, S_\varepsilon) * EL(S_\varepsilon) + P(S_{\delta_i}, S_{a_i}) * U(S_{a_i})$$
$$= C_{\delta_i}(t_{\delta_i}) + P(S_{\delta_i}, S_\varepsilon) * C_\varepsilon + P(S_{\delta_i}, S_{a_i}) * [C_{a_i}(t_{a_i}) + P(S_{a_i}, S_\varepsilon) * C_\varepsilon]$$

Finally, the state S_d has not any penalty itself. Therefore, the utility function is computed as follows:

$$U(S_d) = P(S_d, S_{\delta_i}) * U(S_{\delta_i}) + P(S_d, S_\varepsilon) * U(S_\varepsilon)$$
$$= P(S_d, S_{\delta_i}) * \{C_{\delta_i}(t_{\delta_i}) + P(S_{\delta_i}, S_\varepsilon) * C_\varepsilon + P(S_{\delta_i}, S_{a_i}) * [C_{a_i}(t_{a_i}) + P(S_{a_i}, S_\varepsilon) * C_\varepsilon]\} + P(S_d, S_\varepsilon) * C_\varepsilon$$

Consequently, the utility function for each action is derived from Equation 5-4:

Equation 5-4

$$U^{a_i} = P(S_d, S_{\delta_i}) * \{C_{\delta_i}(t_{\delta_i}) + P(S_{\delta_i}, S_\varepsilon) * C_\varepsilon + P(S_{\delta_i}, S_{a_i}) * [C_{a_i}(t_{a_i}) + P(S_{a_i}, S_\varepsilon) * C_\varepsilon]\} + P(S_d, S_\varepsilon) * C_\varepsilon$$

Let U^0 denote the expected utility of taking no action. Backward induction for this policy gives:

$$U^0(S_d) = P(S_d, S_n) * U^0(S_n) + P^0(S_d, S_\varepsilon) * U^0(S_\varepsilon) = P^0(S_d, S_\varepsilon) * C_\varepsilon$$

The transition probabilities from S_d to S_ε or S_{δ_i} are:

$$P(S_d, S_\varepsilon) = P^\varepsilon(t_0, t_{\delta_i})$$

$$P(S_d, S_{\delta_i}) = 1 - P^\varepsilon(t_0, t_{\delta_i})$$

To proceed from δ_i to α_i, probabilities are given by:

$$P(S_{\delta_i}, S_\varepsilon) = P^\varepsilon(t_{\delta_i}, t_{\alpha_i})$$

$$P(S_{\delta_i}, S_{\alpha_i}) = 1 - P^\varepsilon(t_{\delta_i}, t_{\alpha_i})$$

that is, we move to S_{a_i} if ε does not occur between the time the delay and the action implementation. The transition from S_{δ_i} to S_ε occurs with the complementary probability.

Finally, the distribution over the event occurrence in state a_i is denoted by:

$$P(S_{a_i}, S_\varepsilon) = P_{a_i}^{\varepsilon}(t_{a_i}, T)$$

T indicates the decision horizon, i.e. the end of decision epoch. If no action is taken, the probability to go to state ε is the probability of the event occurrence over the entire interval:

$$P^0(S_d, S_\varepsilon) = P^\varepsilon(t_0, T)$$

And $P^0(S_d, S_n)$ is the complementary probability.

Therefore, Equation 5-4 is transformed to the expression of Equation 5-5:

Equation 5-5

$$\begin{aligned} U^{a_i} = {} & \left[1 - P^\varepsilon\left(t_0, t_{\delta_i}\right)\right] \\ & * \left\{C_{\delta_i}\left(t_{\delta_i}\right) + P^\varepsilon\left(t_{\delta_i}, t_{\alpha_i}\right) * C_\varepsilon + \left[1 - P^\varepsilon\left(t_{\delta_i}, t_{\alpha_i}\right)\right]\right. \\ & \left. * \left[C_{\delta_i}\left(t_{\delta_i}\right) + P_{a_i}^{\varepsilon}\left(t_{a_i}, T\right) * C_f\right]\right\} + P^f\left(t_0, t_{a_i}\right) * C_f \end{aligned}$$

The shape of the expected utility curves is determined by the three main factors that comprise it:

- The cost incurred by the occurrence of failure prior to the time of the action taking affect (corresponding to the first two factors in Equation 5-5). This factor is monotonic increasing in the time of action, since the longer we wait with taking an action, the greater the probability that a failure will happen beforehand.
- The cost incurred by the occurrence of failure despite the application of the mitigating action (third factor in Equation 5-5), which is monotonic decreasing, since the probability that a failure will happen until the end of epoch is decreasing as time progresses.
- The cost of taking the action (last factor in Equation 5-5), which is also decreasing with time, for two reasons: (i) the later the action is planned for, the smaller the

probability it will be taken, since the probability that a failure occurs before the action increases; (ii) the action cost itself is typically nonincreasing.

5.5 Joint Proactive Maintenance and Logistics Optimization

5.5.1 Motivation

Manufacturing failures cause significant problems in human safety, environmental impact and reliability of industrial processes. The fact that unexpected failures deal with uncertainty and stochastic degradation process of manufacturing equipment leads to high uncertainty in the decision making process as well (Van Horenbeek et al., 2013). Thus, there is an increasing demand of maintenance management policies as well as associated information systems in order to reduce unexpected failures, eliminate unscheduled downtimes, and minimize maintenance-related costs (Wu et al., 2007).

Since maintenance and inventory management are strongly interconnected, they should both be considered simultaneously when optimizing a company's operations (Van Horenbeek et al., 2013). Moreover, an accurate reliability evaluation is essential for taking reliable maintenance modelling and spare parts inventory planning decisions (Venkatesan, 1984; Armstrong, and Atkins, 1996; Aronis et al., 2005; Vaughan, 2005; Wu et al., 2007; Elwany, and Gebraeel, 2008; Wang, 2012; Van Horenbeek et al., 2013). The decision about the predictive maintenance of equipment requires a balance between the cost due to premature replacement and the cost of unexpected failure. Moreover, the ordering time of spare parts and their stocking quantities should be planned so that holding costs are minimized by avoiding, at the same time, stock-outs (Elwany, and Gebraeel, 2008; Bohlin, and Wärja, 2015).

Due to the recent advances in technology and information systems and the plethora of methods for prognosis, decision models for joint maintenance and inventory optimization on the basis of prognostic information (e.g. Remaining Useful Life (RUL), Remaining Life Distribution) coming from real-time data (e.g. through sensors) have just started to emerge (Van Horenbeek et al., 2013). Real-time data processing for proactive decision making poses several challenges in efficiency and scalability of the associated information systems. Currently, most of such models and methods can be run offline or on the basis of batches of data at specific sampling times. Although there are research works dealing with extracting insights about current and future situation of business processes, decision making on the basis of real-time, event-driven predictive analytics is still an underexplored area. More specifically, rarely joint maintenance and logistics decision models are real-time and event-driven.

The e-maintenance concept can significantly enhance proactive decision making in maintenance-driven operations management. However, despite the increasing capabilities of e-technologies, maximizing the e-maintenance benefits for the overall maintenance efficiency requires more than technology (Guillen et al., 2016). There is the need for models and methods capable of being embedded in real-time systems triggered by real-time prognostic information in an event processing, streaming computational environment.

To the best of our knowledge, the most representative research work for such kind of problems was proposed by (Elwany, and Gebraeel, 2008) who transformed the decision model proposed by (Armstrong, and Atkins, 1996), so that it is updated continuously in real-time according to the RUL estimation each time a sensor measurement is gathered. To do this, it takes into account the sampling time and follows the “Sense and Respond” concept. However, the availability of a multitude of data generated in the form of very high frequency events by various sources, paves the way for coupling prognostic-based decision methods with sensor-based, event-driven architectures that can support efficient processing of events and improved scalability, while having the ability of handling probability distributions functions instead of parameters (e.g. RUL).

The proposed joint predictive maintenance and spare parts inventory decision models advance the state-of-the-art since they can be deployed in a sensor-based, real-time big data industrial environment using an Event Driven Architecture (EDA) and the e-maintenance concept (Muller et al., 2008) in the context of the framework for Proactive Maintenance. Due to the available prognostic information, the optimal time for maintenance of a part of equipment can be recommended and spare parts can be ordered JIT. The integration in an

EDA enables handling large amounts of data generated by sensors in high frequency, where the continuous update of the decision model is not possible.

5.5.2 State-of-the-Art Analysis

Companies keep inventories of spare parts in order to have availability in case of maintenance. The amount of spare parts in inventory depends on the demand, i.e. the corrective and the preventive maintenance actions requiring the associated spare parts. Therefore, maintenance and inventory management are strongly interconnected and should both be considered simultaneously when optimizing a company's operations (Van Horenbeek et al., 2013). Most of the research works regarding joint maintenance and inventory optimization deal with decisions that rely on time-to-failure/ reliability distributions derived from experimental setups or manufacturing companies' specifications instead of real-time data and thus, they are not able to update the recommendations according to the actual and / or the predicted health state of the equipment. Although in the last years there have been published many research works about real-time prognostics, joint maintenance and spare parts decision models on the basis of these predictions have not been explored, as a consequence of a general lack of proactive decision methods for maintenance. Such an approach could support manufacturing companies minimize their major costs, since a decrease in spare parts inventory cost is among the most significant indirect benefits provided by a proactive strategy (Van Horenbeek et al., 2013).

In addition, almost all published papers on this domain deal with the application of CBM strategy taking into consideration the actual level of degradation, but not the prediction about the future degradation, the future failure or other prognostic information. So, there is untapped opportunity to explore such decision models to the implementation of Proactive Maintenance policy in industrial applications (Van Horenbeek et al., 2013). Due to the available prognostic information, proactive maintenance actions can be recommended and spare parts can be ordered Just-In-Time (JIT) (Van Horenbeek et al., 2013). On the other hand, the equipment downtime may be affected by logistics-related delays, while the time needed for finishing the implementation of the appropriate maintenance actions is rarely accurately known (Van Horenbeek et al., 2013). Finally, the vast majority of published papers assume that the parts of equipment are perfectly maintained after a pre-defined action implementation or do not mention any assumption regarding the degree of restoration (Van Horenbeek et al., 2013).

5.5.3 The Proposed Decision Methods

5.5.3.1 Proactive joint replacement and spare parts inventory decision model

This decision method is based on cost risk analysis (Hulett, 2016) combined with reliability analysis (Ibrahim et al., 2005; Kapur, and Pecht, 2014), while it takes into account the fact that a failure may occur till the next planned maintenance, even though a maintenance action has been implemented, due to low quality of the spare parts replaced or errors in the maintenance process of equipment. This decision model aims to provide timely and reliable recommendations about the optimal time for maintenance and the optimal time for ordering spare parts on the basis of a probability distribution function of a failure occurrence along with its parameters. Degradation modelling usually follows an exponential, a gamma or a Weibull distribution (Elwany, and Gebraeel, 2008; Kapur, and Pecht, 2014). However, in some cases where the cumulative damage does not significantly affect the degradation rate, the linear degradation model can be used (Elwany, and Gebraeel, 2008), since this method does not require a probability distribution belonging to the exponential family.

Each factor of the decision model's long-term maintenance and inventory costs equations represents a cost risk based on the input received from the real-time prediction event. In each time period, there are different associated costs that are expressed as a function of maintenance actions implementation time because their duration may be unknown or too random and there is a cost per unit of time. In addition, the prediction event is received and the recommendation is provided at time t = 0. The long-term maintenance cost as a function of time is extracted by Equation 5-6 while the long-term inventory cost as a function of time is extracted by Equation 5-7. Moreover, Table 1 presents the explanation for each variable.

Equation 5-6

$$C_m(t) = c_f(t) * P^{\varepsilon}(0,t) + \left(c_f(t) + c_p(t)\right) * P_a^{\varepsilon}(t,T) + c_p(t) * \bar{P}^{\varepsilon}(0,T)$$

Equation 5-7

$$C_o(t) = c_s(t) * P^{\varepsilon}(0,t+L) + c_s(t) * P_a^{\varepsilon}(t+L,T) + c_h(t) * \bar{P}^{\varepsilon}(0,T)$$

When an action $\boldsymbol{a}$ is applied to reduce the probability of an undesired event, $\boldsymbol{a}$ is associated with a new event density function $\boldsymbol{g_a^{\varepsilon}(t)}$, which indicates the probability that $\boldsymbol{\varepsilon}$ occurs at time t, although $\boldsymbol{a}$ has been applied before t. This happens because the implementation of action $\boldsymbol{a}$ does not prevent $\boldsymbol{\varepsilon}$ with certainty. Therefore, the probability distributions are calculated as shown in Equation 5-1 and Equation 5-2. In Equation 5-2, the conditioning (denominator) takes into account the fact that until the action occurrence at $\boldsymbol{t_1}$, the distribution in place was $\boldsymbol{G^{\varepsilon}}$.

Table 5-4: Explanation of the decision model's variables.

Variable	Explanation
$\boldsymbol{P^{\varepsilon}(t_1, t_2)}$	Probability distribution function that the failure ε occurs within the time interval (t_1, t_2) conditioned on not occurring until time t_1
$\boldsymbol{P_a^{\varepsilon}(t_1, t_2)}$	Probability distribution function that the failure ε occurs within the time interval (t_1, t_2) conditioned on not occurring until time t_1 and assuming that the action a has been implemented exactly at time t_1
$\boldsymbol{\bar{P}^{\varepsilon}(t_1, t_2)}$	Probability distribution function that the failure ε does not occur within the time interval (t_1, t_2) conditioned on not occurring until time t_1
$\boldsymbol{c_f(t)}$	Cost of failure and of the associated corrective actions as a function of implementation time
$\boldsymbol{c_p(t)}$	Cost of planned maintenance as a function of implementation time
$\boldsymbol{c_s(t)}$	Shortage inventory cost as a function of time
$\boldsymbol{c_h(t)}$	Holding inventory cost as a function of time
$\boldsymbol{L}$	Lead time between the time of placing the order up and the time of receiving the order
$\boldsymbol{T}$	Time until next planned maintenance

$\boldsymbol{c_f(t)}$ is presented in the first and second factor of Equation 5-6. $\boldsymbol{C_f}$ being referred to the cost of failure for each time unit, in the first factor of Equation **5-6**, $\boldsymbol{c_f(t) = C_f * t}$ (e.g. in case of linear function), because the associated probability distribution function refers to the time period (0, t), while in the second factor of Equation 5-6, t is replaced by (T-t), e.g. $\boldsymbol{c_f(t) = C_f * (T - t)}$, because the associated probability distribution function refers to the time period (t, T). $\boldsymbol{c_p(t)}$ is referred to the set of specific pre-defined actions and is presented to the second and third factor of Equation 5-6. It depends on the time period which it refers to. $\boldsymbol{C_p}$ being referred to the cost of planned maintenance for each time unit and $\boldsymbol{\bar{t}_p}$ to the average time needed for planned maintenance, in the second factor of Equation 5-6, $\boldsymbol{c_p(t) = C_p * (T - t)}$ (e.g. in case of linear cost function), while in the third factor of Equation 5-6, t is replaced by $\boldsymbol{\bar{t}_p}$, e.g. $\boldsymbol{c_p(t) = C_p * \bar{t}_{dp}}$, because the associated probability distribution function refers exactly to T, when the planned maintenance is conducted. $\boldsymbol{c_s(t)}$ also depends on the time period which it refers to and is presented to the first and second factor of Equation 5-7. $\boldsymbol{C_s}$ being referred to the shortage cost for each time unit, in the first factor of Equation 5-7, t is replaced by (t+L), i.e. $\boldsymbol{c_s(t) = C_s * (t + L)}$, while in the second factorof Equation 5-7, t is replaced by $\boldsymbol{\left(T - (t + L)\right)}$, i.e. $\boldsymbol{c_s(t) = C_s * \left(T - (t + L)\right)}$. Finally, $\boldsymbol{c_h(t)}$ depends on the time period which it refers to as well and is presented to the third factor of Equation 5-7. $\boldsymbol{C_h}$ being referred to the holding cost for each time unit, in the third factor of Equation 5-7, t is replaced by $\boldsymbol{\left(T - (t + L)\right)}$, i.e. $\boldsymbol{c_h(t) = C_h * \left(T - (t + L)\right)}$.

Equation 5-6 is minimized in order to provide the optimal time of conducting maintenance t_m. In this way, the time-based maintenance can become condition-based by applying the same pre-determined activities when the long-term replacement cost is minimum. This equation consists of three factors which represent the cost risks:

- The cost due to the probability of the occurrence of failure before the time of maintenance actions implementation. This factor shows that the longer we wait for implementing an action, the greater the probability that a failure will happen beforehand.
- The cost due to the probability of the occurrence of failure despite the application of the mitigating action. This factor shows that the probability that a failure will happen until the end of epoch is decreasing in the course of time.

The cost of implementing the action at the end of decision epoch, i.e. the next planned maintenance. This factor is taken into account because there is the possibility of the failure not occurring in the decision epoch although it has been predicted. Equation 5-7 is minimized in order to provide the optimal time of ordering the spare parts t_o. In this way, the spare parts can be ordered JIT, so that the long-term inventory cost is minimum. This equation takes into account the obsolescence of spare parts, which affect the inventory costs, and also consists of three factors which represent the cost risks:

- The cost due to the probability of the occurrence of failure before the time of spare parts ordering plus the lead time required.
- The cost due to the probability of the occurrence of failure despite the action implementation and, therefore, the lack of more spare parts.
- The cost of implementing the action at the end of decision epoch, that is the next planned maintenance. This factor is taken into account because there is the possibility of the failure not occurring in the decision epoch although it has been predicted and thus, the spare parts that have been ordered remain in the warehouse till the next planned maintenance.

5.5.3.2 Proactive joint maintenance and spare parts inventory decision model

This decision model is triggered by prognostic information in an event processing computational environment on the basis of sensor-generated real-time data. Unlike other approaches, it incorporates multiple alternative maintenance actions since the recommended proactive maintenance actions address perfect and various degrees of imperfect maintenance, while each one is mapped to the associated order of spare parts. Rarely joint maintenance and logistics decision models are real-time and event-driven, while they usually provide recommendations about a pre-defined maintenance action (assuming perfect maintenance) with its associated pre-defined order of spare parts.Moreover, it incorporates an MDP model handling transition probabilities distribution functions of time, while, in the place of state rewards, there are costs as functions of action implementation time. Consequently, its output is an action-time policy instead of an action-state policy.

The decision model's output is a set of recommendations about the optimal mitigating (perfect or imperfect) maintenance action (out of a list of alternative actions) along with its implementation time and the optimal order of spare parts that are related to this action along with the optimal ordering time. Domain knowledge entered by users corresponds to the proposed model's input parameters and includes the cost of the equipment failure (e.g. breakdown), the alternative actions along with their cost parameters, and the new lifetime after the action implementation (i.e. how much time each action prolongs the lifetime of the equipment) as well as the decision horizon (e.g. next planned maintenance). The latter is defined by the end of decision epoch, i.e. the time after which the effect of the predicted undesired event fades and the probability of its occurrence returns to normal (Engel et al., 2012). The action-related cost parameters deal with two factors: the cost of action implementation and the cost of action effect (after the action implementation). These two factors apply in both maintenance and inventory aspects and are expressed as a function of implementation time, because actions often affect operation until some specific future time (e.g. taking machinery down to maintenance and losing the rest of the working week). In this sense, the cost is a decreasing function in the activation time. The decision model takes advantage of the basic model for proactive event-driven computing (Engel et al., 2012) and extends it in order to address the joint optimization of maintenance and spare parts ordering in a proactive way when there are multiple alternative maintenance actions and associated spare parts orders. To this end, a MDP model is used and is formulated accordingly.

The output of the MDP is not a policy consisting of an action-state pair, but a policy of an action-time pair, and therefore, the Bellman equation is structured accordingly. The decision model is able to provide recommendations about when to take which action provided that the cost of taking the action and / or the cost of the action effect changes over time. To do this, it incorporates the transition probability distributions as a function of implementation time. The state rewards of the MDP correspond to the costs as functions of implementation time. Consequently, the result is the action with the minimum expected loss (instead of the maximum utility) and the optimal time of applying it. The expected loss function of each action is estimated by using the backward induction algorithm for finite horizon problems (Watkins, and Dayan, 1992) and the Bellman equation is minimized with respect to time. The proactive formulation of the MDP model is solved for both maintenance and logistics so that the resulting expected loss functions are jointly optimized.

Figure 5-4 shows an example of the proactive MDP formulation for joint maintenance and logistics optimization for three alternative actions. On the basis of this formulation, for arbitrary number of actions, the equations of the joint decision model are derived, i.e. the maintenance equation (for each maintenance mitigating action) and the spare parts ordering equation (for each order associated with the respective maintenance mitigating action). Both of them are derived in relation to the predicted failure, but there are different transition probability functions and state rewards in the same formulation, depicted in Figure 5-4. The state rewards correspond to the maintenance costs (i.e. cost of failure, cost of action implementation, cost of action effect) for each alternative action and the inventory costs (i.e. shortage cost, holding cost) associated with each maintenance action along with their lead times. Table 5-5 shows the explanation of the proposed decision model's variables.

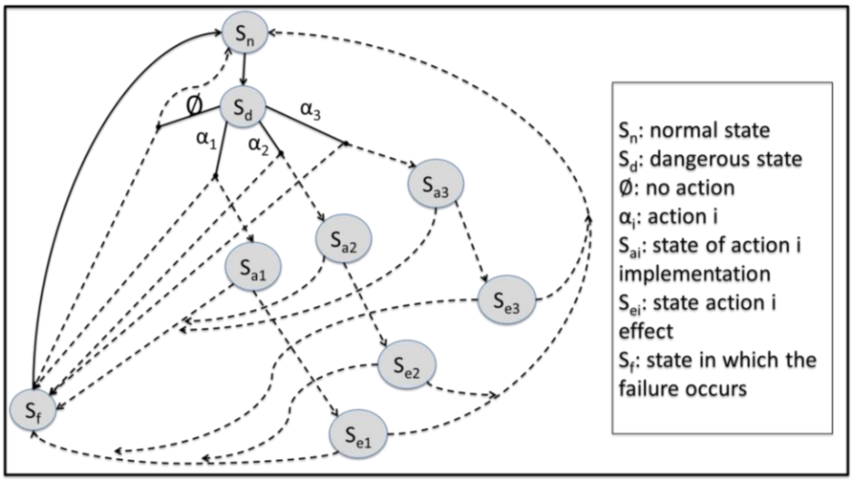

Figure 5-4: An example of the proactive MDP formulation for joint maintenance and logistics optimization.

Table 5-5: Explanation of the proposed model's variables.

Variable	Explanation
$\boldsymbol{P^f(t_1, t_2)}$	Probability distribution function that the failure f occurs within the time interval (t_1, t_2) conditioned on not occurring until time t_1
$\boldsymbol{P_a^f(t_1, t_2)}$	Probability distribution function that the failure f occurs within the time interval (t_1, t_2) conditioned on not occurring until time t_1 and assuming that the action a has been implemented exactly at time t_1
$\boldsymbol{EL^{a_i}(t)}$	Expected loss function for maintenance action a_i
$\boldsymbol{C_f}$	Cost of failure
$\boldsymbol{C_{e_i}(t)}$	Cost function of the action effect
$\boldsymbol{C_{a_i}(t)}$	Cost function of the action implementation
$\boldsymbol{EL^{o_i}}$	Expected loss function for spare parts order o_i
$\boldsymbol{C_{sp}}$	Cost of buying the spare parts
$\boldsymbol{C_s(t)}$	Cost function of shortage inventory
$\boldsymbol{L}$	Lead time between the time of placing the order up and the time of receiving the order
$\boldsymbol{T}$	Decision horizon

5.5.3.2.1 Maintenance Expected Loss Function

For the maintenance equation, based on the aforementioned MDP formulation, there is no cost (or benefit) of being at state Sn, hence $EL(S_n) = 0$. In state f, there is a penalty of C_f

(i.e. the cost of failure), hence $EL(S_f) = C_f$. In state ei we incur penalty of $C_{e_i}(t_{e_i})$ (i.e. the cost function of the action effect) and, given the probability to move to state f, the policy evaluation gives:

$$EL(S_{e_i}) = C_e(t_{e_i}) + P(S_{e_i}, S_f) * EL(S_f) = C_{e_i}(t_{e_i}) + P(S_{e_i}, S_f) * C_f$$

In state a_i, there is a penalty of $C_{a_i}(t_{a_i})$ (i.e. the cost function of the action implementation) and given the probability to move to state f the policy evaluation gives:

$$\begin{aligned} EL(S_{a_i}) &= C_{a_i}(t_{a_i}) + P(S_{a_i}, S_f) * EL(S_f) + P(S_{a_i}, S_{e_i}) * EL(S_{e_i}) \\ &= C_{a_i}(t_{a_i}) + P(S_{a_i}, S_f) * C_f + P(S_{a_i}, S_{e_i}) * [C_{e_i}(t_{e_i}) + P(S_{e_i}, S_f) * C_f] \end{aligned}$$

Finally, the state S_d has not any penalty itself. Therefore, the expected loss is computed as follows:

$$\begin{aligned} EL(S_d) &= P(S_d, S_{a_i}) * EL(S_{a_i}) + P(S_d, S_f) * EL(S_f) \\ &= P(S_d, S_{a_i}) * \{C_{a_i}(t_{a_i}) + P(S_{a_i}, S_f) * C_f + P(S_{a_i}, S_{e_i}) * [C_{e_i}(t_{e_i}) + P(S_{e_i}, S_f) * \\ &\quad C_f]\} + P(S_d, S_f) * C_f \end{aligned}$$

Consequently, the expected loss function for each mitigating maintenance action is derived from Equation 5-8:

Equation 5-8

$$EL^{a_i} = P(S_d, S_{a_i}) * \{C_{a_i}(t_{a_i}) + P(S_{a_i}, S_f) * C_f + P(S_{a_i}, S_{e_i}) * [C_{e_i}(t_{e_i}) + P(S_{e_i}, S_f) * C_f]\} + P(S_d, S_f) * C_f$$

Let EL^0 denote the expected loss of taking no action. Backward induction for this policy gives:

$$EL^0(S_d) = P(S_d, S_n) * EL^0(S_n) + P^0(S_d, S_f) * EL^0(S_f) = P^0(S_d, S_f) * C_f$$

The transition probabilities from S_d to S_f or S_{a_i} are:

$$P(S_d, S_f) = P^f(t_0, t_{a_i})$$

$$P(S_d, S_{a_i}) = 1 - P^f(t_0, t_{a_i})$$

To proceed from ai to ei, probabilities are given by:

$$P(S_{a_i}, S_f) = P^f(t_{a_i}, t_{e_i})$$

$$P(S_{a_i}, S_{e_i}) = 1 - P^f(t_{a_i}, t_{e_i})$$

that is, we move to S_{e_i} if f does not occur between the time the action is applied until the time it takes effect. The transition from S_{a_i} to S_f occurs with the complementary probability.

Finally, the distribution over the event occurrence in state ei is denoted by:

$$P(S_{e_i}, S_f) = P_{e_i}^f(t_{e_i}, T)$$

T indicates the decision horizon, i.e. the end of decision epoch. If no action is taken, the probability to go to state f is the probability of the event occurrence over the entire interval:

$$P^0(S_d, S_f) = P^f(t_0, T)$$

And $P^0(S_d, S_n)$ is the complementary probability.

Therefore, Equation 5-8 is transformed to the expression of Equation 5-9:

Equation 5-9

$$EL^{a_i} = [1 - P^f(t_0, t_{a_i})] * \{C_{a_i}(t_{a_i}) + P^f(t_{a_i}, t_{e_i}) * C_f + [1 - P^f(t_{a_i}, t_{e_i})] * [C_{e_i}(t_{e_i}) + P_{a_i}^f(t_{e_i}, T) * C_f]\} + P^f(t_0, t_{a_i}) * C_f$$

Equation 5-9 expresses the expected loss of each mitigating maintenance action. The minimization of the expected loss functions of all the alternative actions with respect to implementation time provides a recommendation about the optimal action (the action with the global minimum) and the optimal time for its implementation (the time when the expected loss has its global minimum). In Equation 5-9, there is the cost function of the action implementation $C_{a_i}(t_{a_i})$ (i.e. how much the process of action implementation costs – e.g. cost of spare parts, technician pay rate, etc.) and the cost function of the action effect $C_{e_i}(t_{e_i})$ (i.e. how much the result of the action costs – e.g. cost of operating at reduced equipment load). Provided that an estimation of the duration of action implementation is known, $t_{a_i} = t$ and $t_{e_i} = t + \Delta t$, where t indicates the time of action implementation. The polynomial of the action cost function of implementation as well as the initial estimation of the duration of action implementation can be continuously updated, as we are explaining below. In addition, t_0 is considered equal to 0. Consequently, Equation 5-9 is transformed to Equation 5-10:

Equation 5-10

$$EL^{a_i}(t) = [1 - P^f(t_0, t)] * \{C_{a_i}(t) + P^f(t, t + \Delta t) * C_f + [1 - P^f(t, t + \Delta t)] * [C_{e_i}(t + \Delta t) + P^f_{a_i}(t + \Delta t, T) * C_f]\} + P^f(t_0, t) * C_f$$

Considering a fixed cost function of action implementation and the time periods to which the cost function of action effect corresponds, Equation 5-10 is transformed to Equation 5-11:

Equation 5-11

$$\begin{aligned} EL^{a_i}(t) = {} & [1 - P^f(t_0, t)] \\ & * \{C_{a_i} + P^f(t, t + \Delta t) * C_f + [1 - P^f(t, t + \Delta t)] \\ & * [C_{e_i}(T - t - \Delta t) + P^f_{a_i}(t + \Delta t, T) * C_f]\} + P^f(t_0, t) * C_f \end{aligned}$$

5.5.3.2.2 Logistics Expected Loss Function

Similarly to the previous calculations, the logistics-related equation (dealing with spare parts ordering) for each alternative maintenance action is derived from backwards induction algorithm on the basis of the same MDP formulation. In this case, there is a shortage inventory cost function $C_s(t)$ which is inserted in the following equations and a holding cost function which is taken into account indirectly due to the complementary probabilities. In addition, there is a cost of buying the spare parts C_{sp}. The state negative rewards represent the inventory-related costs and the action states represent the order of spare parts that is mapped to each action, as it has been defined at the configuration of the equipment instance. The ordering of spare parts business function is driven by maintenance, therefore, the MDP formulation remains the same, but each state has a different reward which corresponds to the spare parts ordering costs. So, the backwards induction algorithm gives:

$$EL(S_n) = 0$$

$$EL(S_f) = C_s(t_f) = C_s(T-T) = 0$$

$$EL(S_{e_i}) = 0 + P(S_{e_i}, S_f) * EL(S'_f) = P(S_{e_i}, S_f) * C_s(t_{e_i})$$

$$\begin{aligned} EL(S_{a_i}) &= C_{sp} + P(S_{a_i}, S_f) * EL(S'_f) + P(S_{a_i}, S_{e_i}) * EL(S_{e_i}) \\ &= C_{sp} + P(S_{a_i}, S_f) * C_s(t_{a_i}) + P(S_{a_i}, S_{e_i}) * P(S_{e_i}, S_f) * C_s(t_{e_i}) \end{aligned}$$

$$\begin{aligned} EL(S_d) &= P(S_d, S_{a_i}) * EL(S_{a_i}) + P(S_d, S_f) * EL(S'_f) \\ &= P(S_d, S_{a_i}) * [C_{sp} + P(S_{a_i}, S_f) * C_s(t_{a_i}) + P(S_{a_i}, S_{e_i}) * P(S_{e_i}, S_f) * C_s(t_{e_i})] + P(S_d, S_f) \\ &\quad * C_s(t_d) \end{aligned}$$

Therefore, the expected loss function for each action is given by Equation 5-12:

Equation 5-12

$$EL^{o_i} = P(S_d, S_{a_i}) * [C_{sp} + P(S_{a_i}, S_f) * C_s(t_{a_i}) + P(S_{a_i}, S_{e_i}) * P(S_{e_i}, S_f) * C_s(t_{e_i})] + P(S_d, S_f) * C_s(t_d)$$

Let EL^0 denote the expected loss of taking no action. Backward induction for this policy gives:

$$EL^0(S_d) = P(S_d, S_n) * EL^0(S_n) + P^0(S_d, S_f) * EL^0(S_f) = P^0(S_d, S_f) * C_s(t_d)$$

Finally, the expected loss function of ordering the associated spare parts for each action is given by Equation 5-13:

Equation 5-13

$$EL^{o_i}(t) = [1 - P^f(t_0, t_{a_i})] * \{C_{sp} + P^f(t_{a_i}, t_{e_i}) * C_s(t_{a_i}) + [1 - P^f(t_{a_i}, t_{e_i})] * P^f_{a_i}(t_{e_i}, T) * C_s(t_{e_i})\} + P^f(t_0, t_{a_i}) * C_s(t_d)$$

Taking into account the lead times of the spare parts orders, Equation 5-13 can be transformed to Equation 5-14:

Equation 5-14

$$EL^{o_i}(t) = [1 - P^f(t_0, t + L)] * \{C_{sp} + P^f(t + L, t + L + \Delta t) * C_s(t + L) + [1 - P^f(t + L, t + L + \Delta t)] * P^f_{a_i}(t + L + \Delta t, T) * C_s(t + L + \Delta t)\} + P^f(t_0, t_{a_i}) * C_s(T)$$

Considering the time periods to which the shortage cost function corresponds to, Equation 5-14 is transformed to Equation 5-15:

Equation 5-15

$$EL^{o_i}(t) = [1 - P^f(t_0, t + L)] * \{C_{sp} + P^f(t + L, t + L + \Delta t) * C_s(T - t - L) + [1 - P^f(t + L, t + L + \Delta t)] * P^f_{a_i}(t + L + \Delta t, T) * C_s(T - t - L - \Delta t)\} + P^f(t_0, t + L) * C_s(T)$$

5.5.3.2.3 Joint optimization of maintenance and logistics

Equation 5-10 and Equation 5-14 constitute the generic proactive decision model for joint maintenance and logistics optimization that is triggered by a prediction event containing the PDF of the equipment under consideration failure. Since the PDF depends on the degradation modelling until the breakdown, it will usually follow distribution belonging to the exponential family (e.g. exponential, Weibull, gamma) (Kapur, and Pecht, 2014) and therefore, it will fulfil the Markov property. Otherwise, it should be filtered and processed by the previous decision method for joint maintenance and logistics optimization. Before optimizing the equations of the proposed decision model, the PDFs should be calculated according to reliability theory. Therefore the Equation 5-1 and the Equation 5-2 are adapted accordingly.

Equation 5-16

$$P^f(t_1, t_2) = \frac{G^f(t_2) - G^f(t_1)}{1 - G^f(t_1)}$$

Equation 5-17

$$P_{a_i}^{f}(t_1, t_2) = \frac{G_{a_i}^{f}(t_2) - G_{a_i}^{f}(t_1)}{1 - G^{f}(t_1)}$$

5.6 Proactive Selection of Maintenance Spare Parts' Suppliers

5.6.1 Motivation

Since manufacturing companies need to work with different suppliers of maintenance spare parts, the purchasing department can play a key role in cost reduction and risk optimization as well as in empowering the suppliers for improved quality, response time and reliability of supplies deliveries (Sepehri, 2013). In this sense, the strategic process of supplier management is replacing the function of purchasing (Sepehri, 2013) involving a smaller numbers of highly qualified buyers, decentralized control of non-value adding items and greater planning activity horizons. Consequently, supplier selection becomes one of the most important operations of supply chain management, since it should split the order quantities among suppliers for creating a constant environment of competitiveness (Sepehri, 2013).

5.6.2 State-of-the-Art Analysis

In manufacturing enterprises, procurement deals not only with the raw materials required for the production process, but also with spare parts needed for maintenance. Therefore, the supplier relationship strategy should be aligned with the equipment maintenance strategy (Slack et al., 2010). Since the supplier selection process occupies a large amount of resources, companies expect to conclude in high value contracts. However, prices of spare parts and raw materials are subjected to fluctuations with uncertain trends, making procurement, and especially supplier relationship management, a key element of business performance (Sepehri, 2013). Suppliers' prices affect long-term business profitability, business reputation and output product's price, thus suppliers' prices prediction algorithms and autonomous interacting software agents have gathered an increased interest during the last years (Godarzi et al., 2014). At the same time, procurement management should ensure reliability and quality of supplies in conjunction with the transaction costs and risks in a dynamic uncertain environment (Sepehri, 2013). Procurement management driven by proactive maintenance can benefit from lean manufacturing in order to eliminate operation's wastes during the production process (Cortes et al., 2016). Cooperating with one outsourced supplier may cause significant problems (Cortes et al., 2016), so having more choices of suppliers that produce and deliver the same components can lead to less future risks and costs (Sepehri, 2013).

5.6.3 The Proposed Decision Method

This decision method is triggered by: (i) a recommendation about the optimal actions as well as the optimal time for a maintenance action implementation along with the optimal time of ordering the required spare parts; and (ii) the prediction of the spare parts' prices until the decision horizon. It also takes into account the available purchasing budget, the number of the potential suppliers and historical data about past portfolios. Then, it provides recommendations about the optimal portfolio of suppliers given the purchasing budget at the recommended future ordering time so that the expected losses are minimized. Moreover, it incorporates the last update for prediction of suppliers' prices throughout a decision horizon (e.g. until next planned maintenance), as derived from a predictive analytics service. The output of the algorithm is the 'Markowitz bullet' and its 'efficient frontier' as well as the optimal portfolio of suppliers, i.e. the percentage of the available purchasing budget that will be spent in each supplier out of a list of potential suppliers. The use of a portfolio optimization approach supports the allocation of scarce resources in the manufacturing enterprise to different supplier relationships and thus, the minimization of supply-related risks. Since information processing is asynchronous, the supplier recommendation service receives and stores the most recent update of the suppliers' prices predictions (e.g. from an ERP system based on EDI data) in order to use it when the joint maintenance and logistics recommendation service triggers it.

Modern portfolio theory, or mean-variance analysis, is a mathematical framework for assembling a portfolio of assets such that the expected return is maximized for a given level of risk, defined as variance. Its key insight is that an asset's risk and return should not be assessed by itself, but by how it contributes to a portfolio's overall risk and return (Markowitz, 1950). In the supplier selection problem, assets correspond to a pre-defined number of potential suppliers for maintenance spare parts. MPT shows how to choose a portfolio with the maximum possible expected return for the given amount of risk. Two essential decisions are necessary to be made to choose the best portfolio from a number of possible portfolios, each with its risk and return opportunities: (i) Determine a set of efficient portfolios; and (ii) Select the best portfolio out of the efficient set.

Therefore, the optimal ordering time is received by Markowitz Portfolio Theory (MPT) optimization algorithm (Markowitz, 1952) and is processed in order to enable the purchasing department to decide in advance what proportion of the procurement budget should be spent to each supplier based on the prices that they offer in the course of time for the same maintenance spare parts. In this case, the assets correspond to the suppliers and the portfolio indicates the percentage of the whole amount of money that should be given to each supplier for company's procurement.

The optimal portfolio of suppliers is defined according to the risk and expected return equations of MPT.

Expected Return: $E(R_p) = \sum_i w_i E(R_i)$, where R_p is the return on the portfolio of suppliers, R_i is the return on supplier i and w_i is the the proportion of supplier "i" in the portfolio).

Portfolio return variance: $\sigma_p^2 = \sum_i w_i^2 \sigma_i^2 + \sum_i \sum_{j \neq i} w_i w_j \sigma_i \sigma_j \rho_{ij}$, where ρ_{ij} is the correlation coefficient between the returns on suppliers i and j.

Portfolio return volatility: $\sigma_p = \sqrt{\sigma_p^2}$

The manufacturing company that needs spare parts to be supplied can reduce the risk of the portfolio of suppliers simply by holding a diversified portfolio of suppliers. The "risk-expected return space" plot of MPT represents every possible combination of risky suppliers and the collection of all such possible portfolios defines a region in this space. The left boundary of this region is a hyperbola, and the upper edge of this region is the efficient frontier in the absence of a risk-free supplier. Combinations along this upper edge represent portfolios for which there is lowest risk for a given level of expected return. Equivalently, a portfolio laying on the efficient frontier represents the combination offering the best possible expected return for given risk level. The tangent to the hyperbola at the tangency point indicates the best possible CAL.

For a given "risk tolerance" $q \in [0, \infty)$, the efficient frontier is found by minimizing the following expression:

$w^T C w - q * R^T w$, where:

- w is a vector of portfolio weights and $\sum_i w_i = 1$.
- C s the covariance matrix for the returns on the suppliers in the portfolio
- $q \geq 0$ is a "risk tolerance" factor, where 0 results in the portfolio with minimal risk and ∞ results in the portfolio infinitely far out on the frontier with both expected return and risk unbounded
- R is a vector of expected returns
- $w^T C w$ is the variance of portfolio return
- $R^T w$ is the expected return on the portfolio

An alternative approach to specifying the efficient frontier is to do so parametrically on the expected portfolio return $R^T w$. This version of the problem requires that we minimize $w^T C w$ subject to $R^T w = \mu$ for parameter μ. This problem is solved using convex optimization (Diamond, and Boyd, 2016) because it is a complex problem with bounds, constraints and a Lagrange multiplier. **Figure 5-5** depicts an example of the Markowitz bullet and the Efficient Frontier.

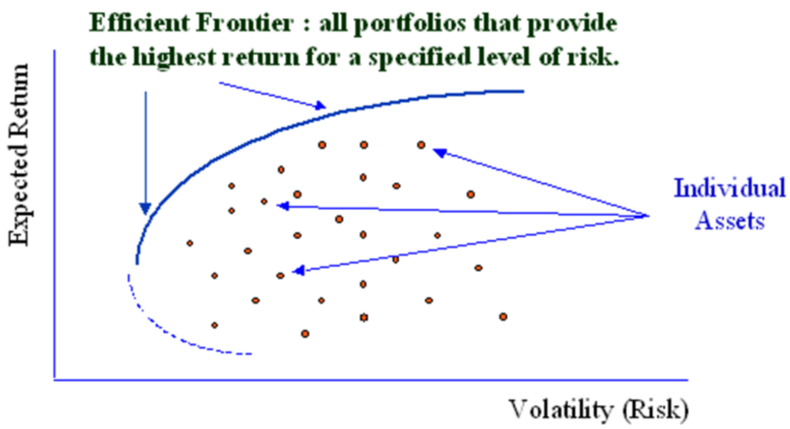

Figure 5-5: The Markowitz bullet and the Efficient Frontier.

6 Continuous Improvement of Proactive Decision Making

In this Chapter, the proposed approach for continuous improvement of proactive event-driven decision making is presented. In order to tackle with the high sensitivity of proactive decision making to its input parameters, the Sensor-Enabled Feedback (SEF) approach processes and analyses sensor-generated data with the aim to improve the accuracy of proactive decision methods' user-defined parameters and consequently, the reliability of recommendations.

6.1 Introduction and Motivation

As an emerging technology, Internet of Things (IoT) is expected to offer promising solutions to transform the operation and role of manufacturing systems with the use of appropriate sensory, communication, networking, and information processing technologies (Da Xu et al., 2014). Since design and operation of a manufacturing system requires decision making at all levels and domains of business activities, prompt and effective decisions depend not only on reasoning techniques, but also on the quality and quantity of data. Every major shifting of manufacturing paradigm has been supported by the advancement of information technology. The evolution of IoT and the development of industrial event-processing technologies pave the way for proactivity in decision making, i.e. the ability to decide and act ahead of time based on data-driven predictions.

Proactive event-driven decision making is highly sensitive to its input parameters (Engel et al., 2012), especially to those related to action cost, as shown in the evaluation results. Even slightly different action cost values compared to their actual values may lead to the recommendation of a wrong (not optimal) action and/or timing for its implementation. Since cost related information may be either estimated by humans or measured through sensors, these deviations may occur due to user input's inaccuracies or the quality of collected data (e.g. due to sensor noise), respectively.

Proactive decision making is sensitive to its input parameters, in the sense that their inaccurate estimations can lead to wrong recommendations. Deviations of action costs as a function of time have a strong impact over the generated recommendation, since even slightly different values of action cost functions compared to their actual values, may lead to the recommendation of a wrong action and / or a wrong optimal time for its implementation, as shown in Figure 6-1. To overcome the aforementioned problems associated with the inaccuracy of manually inserted cost-related information and the resulting inaccurate recommendations, our approach enables the continuous learning of each action cost function by considering the actual costs incurred because of the action during the time period it is implemented.

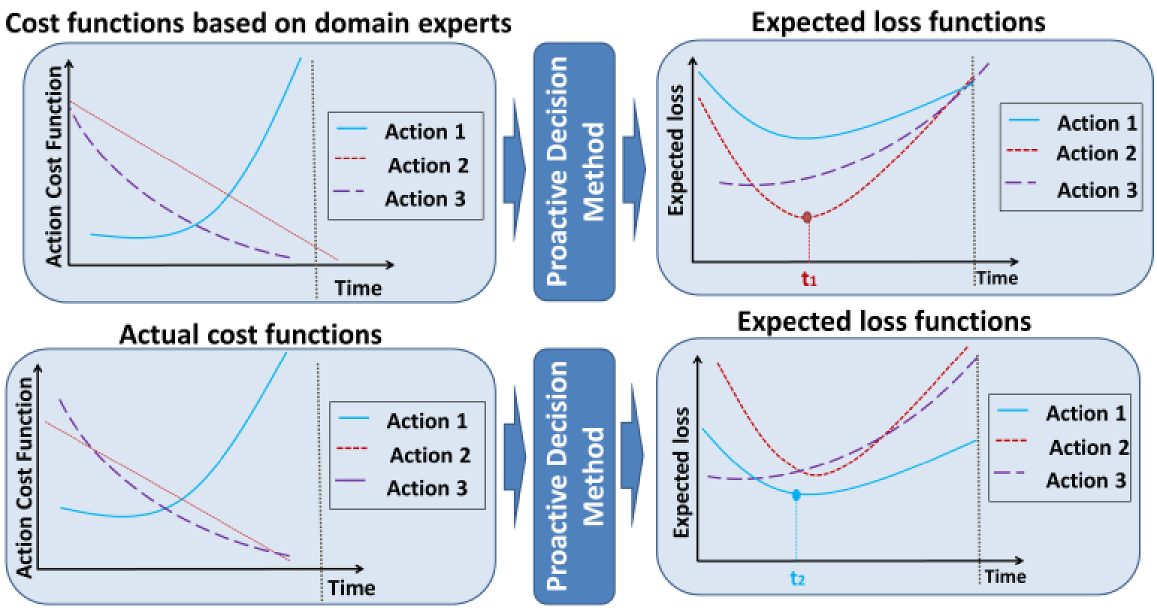

Figure 6-1: The effect of inaccurate cost functions on the proactive decision making output.

In this Section, the development of an approach for automated and accurate cost estimations in a real-time streaming computational environment is presented. The aim is to enhance proactive event-driven decision making for maintenance. The proposed Sensor-Enabled Feedback (SEF) approach collects data during action implementation and uses them as feedback with the aim to update the action cost function. The updated cost function can then be used in the next recommendation cycle involving this action. Therefore, the aim of the SEF approach is twofold: (i) to inform the user online about the estimated cost of action during action implementation, and (ii) to update the cost function of the specific action and use it in the next recommendation in which this action is involved. The proposed approach is independent of the proactive decision methods used, in the sense that it increases the accuracy of their cost-related input parameters without changing the methods themselves.

In the following sub-sections, the approach of SEF and its instantiation to maintenance operations is described. The approach and the algorithm address three blocks of the conceptual architecture for the Decide phase: the “DMI Configuration” (as far as the SEF configuration is concerned) block of the User Interaction Layer, the “Proactive Decision Methods” block of the Real-time Processing Layer and the “Online Monitoring” block of the User Interaction Layer. These three blocks are highlighted with red color in the conceptual architecture in Figure 6-2.

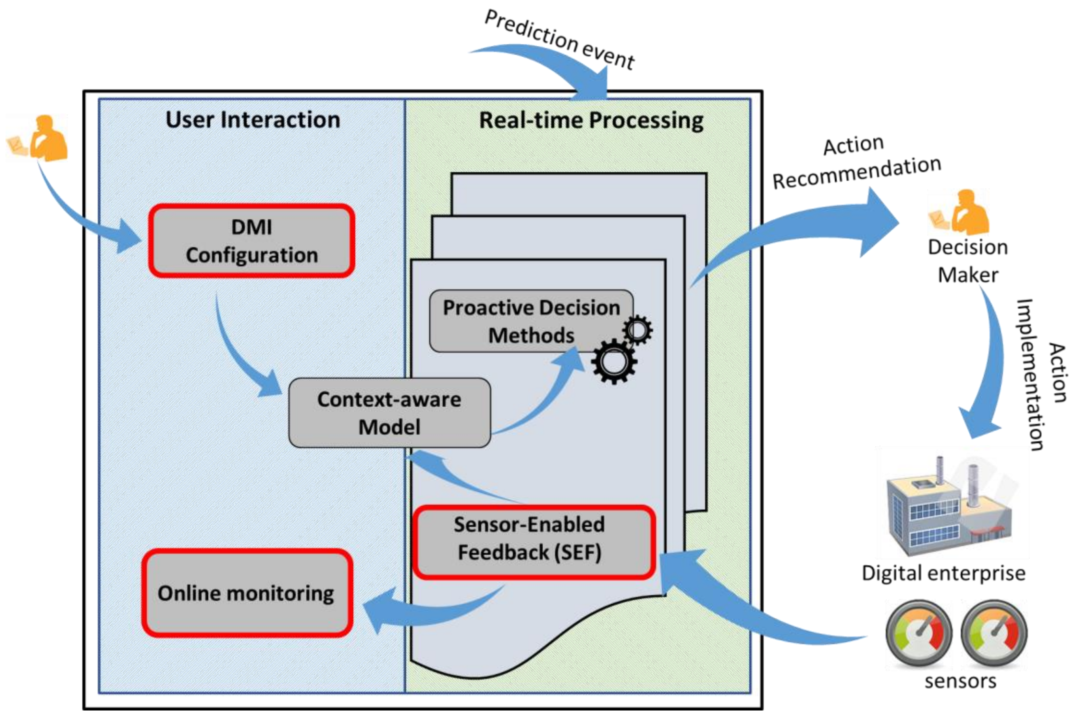

Figure 6-2: The functionalities for continuous improvement of proative decision making in the conceptual architecture.

6.2 *State-of-the-Art Analysis*

Although user input inaccuracy and industrial sensor noise affect significantly the effectiveness of condition monitoring and maintenance optimization, only few research works have investigated approaches for tackling them (de Jonge et al., 2017). Despite the wide use of sensor data acquisition and manipulation, existing research works mainly focus on condition monitoring applications for visualization, for exposing real-time information to the user and for detecting the current health state of a manufacturing system. These inaccuracies can be eliminated with the use of adaptation mechanisms during the actions implementation (Krumeich et al., 2016). To the best of my knowledge, there is not an approach for exploiting sensor and legacy data for cost estimations with the aim to improve the generated maintenance recommendations. For accurate cost estimations, all the contributors (i.e. cost factors) to the cost function of each action (e.g. the waiting orders, the equipment availability, the transport costs) should be taken into account. This fact is achieved through the user interaction and specifically, by enabling the expert to insert their domain knowledge in an information system at design time. To this end, the need for generic tools capable of integrating this information in order to formulate cost functions and thus, facilitate decisions has recently been identified in literature (Carlander et al., 2016). However, existing works consider the domain knowledge inserted by domain experts as fixed.

6.3 The Approach for Sensor-Enabled Feedback

The current work develops a method for continuous improvement of proactive recommendations through event-driven SEF, in order to extend and build on the Decide phase. SEF gathers and processes feedback related to cost with the aim to continuously improve proactive maintenance decision making. This improvement is realized in terms of accuracy in the estimation of the cost-related input parameters and therefore, in terms of reliability of the generated recommendations. The SEF approach enhances and extends proactive event-driven decision making for maintenance by utilizing the combination of online changepoint detection, noise filtering and curve fitting algorithms.

The DMI configuration requires domain knowledge from the user, which includes quantified cost functions of the various alternative actions over time. Action cost may be a function of its implementation time, while actions usually affect the operations until a specific future time (end of decision epoch). Examples of actions include take the equipment down for maintenance and lose the production for the rest of the working day, or reduce the production rate until the end of the shift. In these cases, cost is mainly a decreasing function with respect to action implementation time because the later an action is implemented, the lower the total cost associated with the action is, due to its shorter duration until the end of the decision horizon. On the other hand, the earlier an action is implemented, the lower the failure risk is. In some cases, the cost function can express different meanings. For example, if an action cannot prevent the undesired event, but it can reduce its impact, the cost of the action can include the reduced cost of undesired event. Such cost-related information may be limited or inaccurate, while the cost functions themselves may also change in the course of time, making their initial estimations from domain experts not only cumbersome to obtain but also obsolete.

The cost functions are configurable according to the implementation domain, the available sensors and the problem to be addressed. They are structured based on the sensor measurements and the cost data either provided by the user or existed in the manufacturing company's systems (e.g. the production plan in the ERP). In other words, at the user interaction layer, the user is able to formulate the cost functions based on their expert knowledge and the available historical data in order to take into account all the contributors (i.e. cost factors) to the cost function of each action (e.g. the waiting orders, the equipment availability, the transport costs). The cost functions are formulated with respect to action implementation time. The SEF approach is implemented in two sub-components, which are detailed in the next sections: "Total Cost Calculation" and "Cost Function Estimation".

6.3.1 Total Cost Calculation

The total action cost function in the manufacturing domain is typically an aggregation of different cost factors such as labour cost, cost due to downtime, cost due to scrapped parts, cost due to warranty claims, cost of spare parts, etc. Depending on the nature of the alternative actions, different existing manufacturing cost models can be used to decompose action costs to several cost factors; see e.g. (Amorim-Melo et al. 2014). SEF leverages frequent feedback on the actual values of the various cost factors through different sensors (e.g. pressure sensor, accelerometer, etc. but also ERP) that provide either directly or indirectly cost-related real-time information during system operation. Noise in cost-related measurements expressed as a cost deviation exists due to the noise apparent in hardware sensors and data/ information quality deficiencies (accuracy, timeliness, adequacy and credibility) (Li, and Lin, 2006) of information stored in production systems. Real-time information processing is able to overcome issues of delay and distortion (Hazen et al., 2014) provided that the level of data consistency is high in the attributes that are objective to the data (Kwon et al., 2014). SEF uses post-action implementation cost factor data for refining the total action cost function.

Figure 6-3 provides a zoom-in view of the "Total Cost Calculation" SEF sub-component. This sub-component gathers sensor data corresponding to cost factors, identifies significant deviations of their values in the course of time with the aim to detect when the corresponding action starts or ends, removes noise from the sensor measurements (thus improving data/ information quality) and calculates the total cost of the action by aggregating the measurements of all the cost factors.

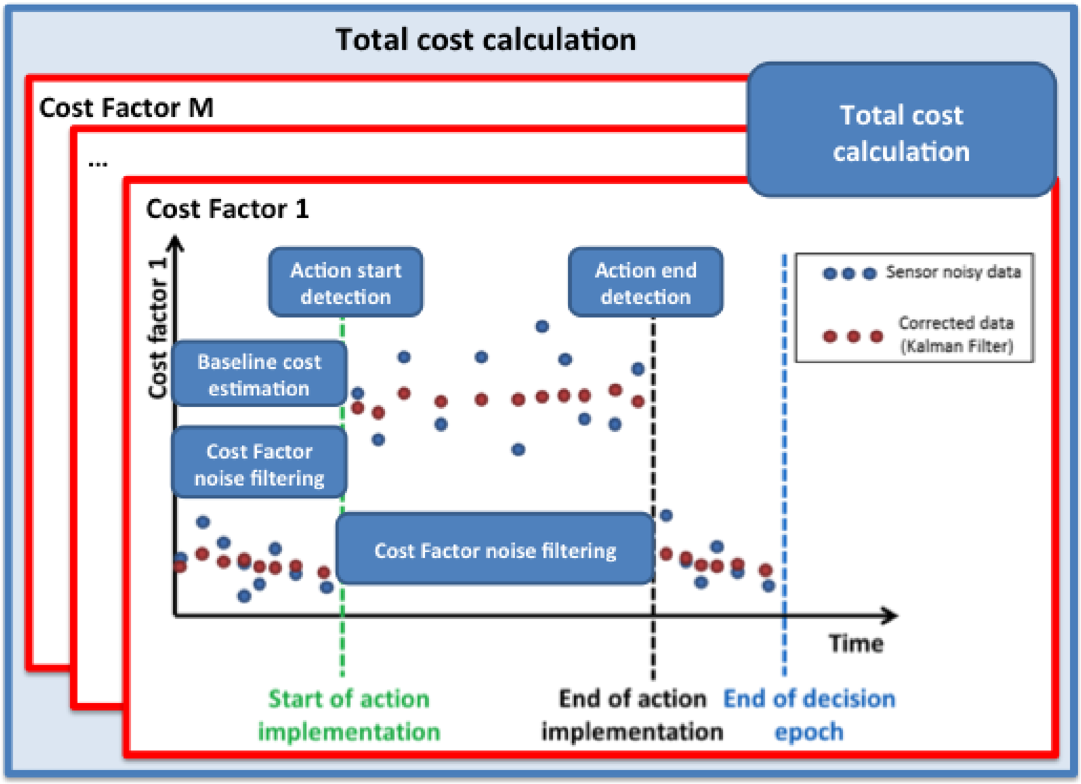

Figure 6-3: The process for each cost factor the aggregation of which results in the "Total Cost Calculation" sub-component of SEF

The costs that are related to sensor measurements and company's systems may not be attributed to the single recommended action; therefore, we distinguish between baseline cost data measured before action implementation and action-related cost data, with the latter being calculated by subtracting the baseline from the total cost during action implementation. As shown in Figure 6-3, the "Total Cost Calculation" SEF sub-component incorporates the estimation of that cost baseline for each cost factor, i.e. of an aggregated value of cost data measured during the period before the action implementation. First, the sub-component identifies whether the current state is a state before action implementation (i.e. sensor data corresponds to baseline costs), or a state during action implementation (i.e. sensor data corresponds to total costs). The transition from the one state to the other is identified online through the action start/end detection processes, both of which are described in more details below. Once the point of transition has been identified, the cost baseline is calculated by applying Curve Fitting algorithms over the corrected (after noise filtering) cost data time series preceding it. The whole process is exposed to the users through online monitoring and visualization.

The action start/end detection processes, mentioned above, are responsible for identifying transitions from a "no action" to an "action" state and vice versa, respectively. At some point, a higher cost compared to the baseline is identified in one or more cost factors. This is an indicator of an action starting, since this cost increase occurs due to an action implementation. Its cost function consists of these specific cost factors that have previously defined and configured during user interaction. The transition between a "no action" and an "action" state is identified through an online cost changepoint detection algorithm. Such kinds of algorithms have been proved to achieve high levels of accuracy and effectiveness by conducting online, real-time anomalies detection in a recursive way (Maleki et al., 2016); however, they have mainly be used in fault detection applications. The focus is on causal predictive filtering; generating an accurate distribution of the next unseen datum in the sequence, given only data already observed. In the proposed methodology, a real-time, event-driven cost changepoint detection algorithm which considers only the most recent change by incorporating Bayesian Inference (Adams, and MacKay, 2007) is applied. In this way, the algorithm identifies immediately the change of the system state (no action – baseline cost, action-action cost) based on the probabilistic distribution over the possible runs. The algorithm is formulated as follows.

We assume that a sequence of cost factors observations $CF_1, CF_2, \dots, CF_T$ may be divided into non-overlapping product partitions (Barry, and Hartigan, 1992). The delineations between partitions are called the changepoints. We further assume that for each partition

ρ, the data within it are i.i.d. from some probability distribution $P\left(CF_t|\eta_\rho\right)$. The parameters η_ρ, $\rho = 1,2, \dots$ are taken to be i.i.d. as well. We denote the contiguous set of observations between time a and b inclusive as $x_{a:b}$. The discrete a priori probability distribution over the interval between changepoints is denoted as $P_{gap}(g)$. We are concerned with estimating the posterior distribution over the current "run length," or time since the last changepoint, given the data so far observed. We denote the length of the current run at time t by r_t. We also use the notation $CF_t^{(r)}$ to indicate the set of cost factors observations associated with the run r_t. A r may be zero, the set $CF^{(r)}$ may be empty. The function H is the hazard function.The overview of the Bayesian Online Cost Changepoint Detection algorithm is shown below:

1. **Initialize**

$$P(r_0) = S(r) \; or \; P(r_0 = 0) = 1$$
$$v_1^{(0)} = v_{prior}$$
$$X_1^{(0)} = X_{prior}$$

2. **Observe New Datum x_t**
3. **Evaluate Predictive Probability**

$$\pi_t^{(r)} = P\left(CF_t|v_t^{(r)}, X_t^{(r)}\right)$$

4. **Calculate Growth Probabilities**

$$P(r_t = r_{t-1} + 1, x_{1:t}) = P(r_{t-1}, CF_{1:t-1})\pi_t^{(r)}\left(1 - H(r_{t-1})\right)$$

5. **Calculate Changepoint Probabilities**

$$P(r_t = 0, CF_{1:t}) = \sum_{r_{t-1}} P(r_{t-1}, CF_{1:t-1})\pi_t^{(r)} H(r_{t-1})$$

6. **Calculate Evidence**

$$P(CF_{1:t}) = \sum_{r_t} P(r_t, CF_{1:t})$$

7. **Determine Run Length Distribution**

$$P(r_t|CF_{1:t}) = P(r_t, x_{1:t})/P(CF_{1:t})$$

8. **Update Sufficient Statistics**

$$v_{t+1}^{(0)} = v_{prior}$$

$$X_{t+1}^{(0)} = X_{prior}$$

$$v_{t+1}^{(r+1)} = v_t^{(r)} + 1$$

$$X_{t+1}^{(r+1)} = X_t^{(r)} + u(CF_t)$$

9. Perform Prediction

$$P(CF_{t+1}|CF_{1:t}) = \sum_{r_t} P(CF_{t+1}|CF, r_t)P(r_t|CF_{1:t})$$

10. Return to Step 2

Since sensors and other sources of data generate noisy and low quality data related either to the baseline cost or to the cost of the implemented action, a noise filtering algorithm is required so that an accurate estimation of the costs (i.e. after removing noise) is made, allowing to base further processing for the calculation of baseline and action costs to more reliable data. Noise filtering of cost data time series is based on Kalman filter (and its non-linear extensions where the state transition and observation models are not linear functions of the state, but they are of differentiable type, i.e. Unscented Kalman Filter) (Kalman, 1960; Wan, and Van Der Merwe, 2000; Julier, and Uhlmann, 2004), one of the most widely used methods for filtering, tracking, estimation and prediction (Ali, and Ushaq, 2009; Liu et al., 2016) since it minimizes the variance of the estimation Mean Squared Error (MSE) (Jwo, and Cho, 2007). The main advantage of Kalman filter over other noise filtering methods is in the computational efficiency of the algorithm due to its efficient use of matrix operations allowing for longer real-time artifact removal (Rajan, and Rajalakshmy, 2014). This aspect is crucial for the proposed SEF approach due to the need for scalable real-time big data processing. Moreover, since Kalman filter provides a sequential Minimum MSE estimation for a time-varying parameter vector that follows a state-space dynamical model, it combines several advantages from other noise filters and namely simplicity, optimality, tractability and robustness (Ali, and Ushaq, 2009; Liu et al., 2016). In addition, the Kalman filter has been proved to perform better than the median filter, the Butterworth low-pass filter and the discrete wavelet package shrinkage in terms of Signal-to-Noise Ratio (SNR) and correlation coefficient (R) between filtered and reference signals (Wang et al., 2011). Moreover, the Kalman filter has been proved to have a better performance than Fast Fourier Transform detection (Will, and Cardoso, 2012). Despite its wide use in condition monitoring applications aiming to detect the current health state of the equipment, it has not been used for cost estimations with the aim to improve the generated maintenance recommendations.

For each cost factor, sensors generate noisy data with a specific frequency either with uniform or with non-uniform sampling. These noisy data can be filtered to remove noise and provide an accurate estimation of the variable of interest (cost factor). Different sensor samplings (e.g. uniform, non- uniform sampling) and cost function polynomials per cost factor are supported. The type of cost functions may be affected either by the cost model itself (e.g. labour cost may be a linear function which corresponds to a pay rate of X euros per hour and consequently, the noise corresponds to data low quality which leads to missing data due to errors in data entry) or by a business process that causes a cost increase (e.g. the number of defects per unit of time affects the cost function due to scrapped parts).

Therefore, SEF filters noisy cost-related measurements in an event processing computational environment and has two steps in each iteration: (i) Prediction, and (ii) Correction. In each step, a set of equations based on Kalman Filter theory is solved, aiming to remove noise from the cost-related measurements, as shown in Table 6-1.

Table 6-1: Cost noise filtering set of equations

Prediction	Correction
Cost Factor Value Prediction: $\widehat{CF}_k^- = A\widehat{CF}_{k-1}$ *Covariance Prediction:* $P_k^- = AP_{k-1}A^T$	*Kalman Gain:* $K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$ *Cost Factor Value Update:* $\widehat{CF}_k = \widehat{CF}_k^- + K_k(z_k - H\widehat{CF}_k^-)$ *Covariance Update:* $P_k = (I - K_k H)P_k^-$

In the prediction step, the Kalman filter produces estimates of the current state variables, along with their uncertainties. Once the outcome of the next measurement (necessarily corrupted with some amount of error, including random noise) is observed, these estimates are updated using a weighted average, with more weight being given to estimates with higher certainty. The algorithm is recursive. It can run in real time, using only the present input measurements and the previously calculated state and its uncertainty matrix; no additional past information is required.In this way, a series of measurements observed over time containing statistical noise and other inaccuracies is used and estimates of unknown variables that tend to be more accurate than those based on a single measurement alone are produced. To do this, a joint probability distribution over the variables for each timeframe is estimated (Kalman, 1960).

Our model assumes that the true value of the cost factor at time k is evolved from the state at (k − 1) according to the "cost factor value prediction" equation. $\widehat{CF}_k^-$ corresponds to the prior estimate of cost factor value at the kth time step, A represents the cost value transition matrix which is applied to the previous state, P_k^- represents the prior error covariance matrix, P_k the error covariance matrix, while $\widehat{CF}_k$ corresponds to the current estimate of cost value at the kth time step. K_k is the Kalman Gain, H represents the cost measurement matrix, R represents the cost deviation caused by sensor noise, while z_k corresponds to the cost measurement vector. It should be noted that at the point of transition (changepoint) from the "no action" to the "action" state, the cost factor noise filtering algorithm referring to the baseline cost measurements stops and restarts being applied for the new values (after action implementation).

During each decision epoch that an action is recommended and implemented, the total cost of this action is calculated based on the measured values of the underlying cost factors and their associated timestamps. This calculation is conducted by adding all the cost measurements during the implementation of the action for all the associated cost factors. This cumulative total action cost is used not only for online monitoring by the user, but also for further data processing in the context of the cost function estimation sub-component which is described in the following sub-Section.

6.3.2 Cost Function Estimation

Figure 6-4 provides a zoom-in view of the "Cost Function Estimation" SEF sub-component. The left part of Figure 6-4 shows the cumulative total cost for an action in several decision epochs where that action has been implemented. In each iteration, the action has been implemented at a different time point with respect to the end of decision epoch. The cost C(ti) represents the total cost of the action when it was implemented at the specific time, i.e. for this specific remaining time until the end of decision epoch. The pairs of the cumulative total action cost and the remaining time until the end of decision epoch actually represent points of the action cost function as can be seen in right part of Figure 6-4.

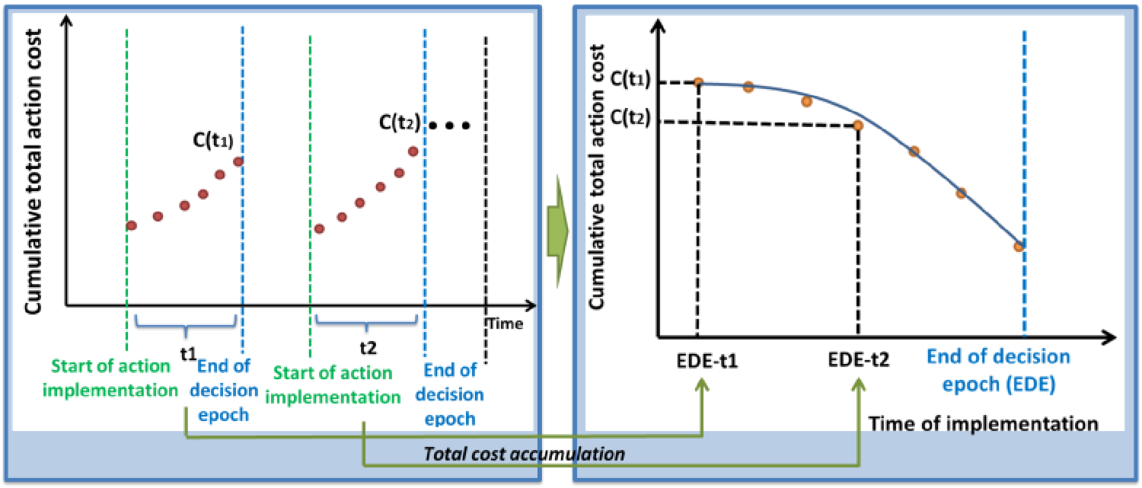

Figure 6-4: "Cost Function Estimation" sub-component of SEF.

To estimate the analytical expression of the action cost function from these points, we apply **Curve Fitting** (Shawash, and Selviah, 2013) with non-negativity constraints, since the action cost should be positive in order to express the expenses due to an action implementation. Curve Fitting is applied on the basis of points constrained to supposed polynomials using the Levenberg–Marquardt algorithm for the sum of squares minimization (Moré, 1978; Lourakis, 2005; Shawash, and Selviah, 2013) applied to the cost function, estimated according to corrected sensor data. Therefore, given a set of m empirical datum pairs of time and cumulative total action cost (t_i, C_i), the goal is to find the parameters β of the model curve $f(t, \beta)$ so that the sum of the squares of the deviations $S(\beta)$ is minimized:

$$\hat{\beta} = arg\min_{\beta} S(\beta) \equiv arg\min_{\beta} \sum_{i=1}^{m} [C - f(t_i, \beta)]^2$$

Therefore, let the Jacobian of $f(t_i)$ be denoted $J_i(t)$, then the Levenberg-Marquardt method searches in the direction given by the solution p to the equations:

$$(J_k^T J_k + \lambda_k I) p_k = -J_k^T f_k$$

Where λ_k are nonnegative scalars and I is the identity matrix. The method has the property that, for some scalar Δ related to λ_k, the vector p_k is the solution of the constrained subproblem of minimizing $\|J_k p + f_k\|_2^2/2$ subject to $\|p\|_2 \leq \Delta$.

Since the distribution that these points follow is not known in advance, a curve comparison algorithm is applied. More specifically, polynomials of various degrees are compared with respect to the Mean Squared Error (MSE) which includes a regularization term in order to avoid overfitting (Eldar et al., 2005) and to result in convergence. The MSE measures the average of the squares of the errors or deviations (Lehmann, and Casella, 2006). If $\hat{X}$ is a vector of n predictions and X is the vector of observed values of the variable being predicted, then the within-sample MSE of the predictor is the mean of the squared of errors and is computed as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \hat{X}_i\right)^2$$

Consequently, the polynomial with the lowest MSE is selected as input to Curve Fitting. Based on the data points and the polynomial order, the refined action cost as a function of the implementation time is derived in order to be used at the same DMI the next time it will be triggered by a prediction event. However, the user is provided with the capability to select whether they prefer the use of the refined cost function or the use of their initial guess. This may be required in cases where the particular action has not been recommended several times and therefore, there is not enough data for a reliable result of curve fitting. Consequently, the user may prefer to be based on their domain knowledge.

7 Context-awareness in Proactive Decision Making

In this Chapter, the context-awareness mechanism for proactive decision making is presented. More specifically, the proposed approach is able to deal with uncertainty in decision making, while it is able to be embedded in a real-time, event-driven computational enironment. Moreover, the proposed approach feeds into the proactive decision methods and is updated through SEF.

7.1 Introduction and Motivation

The emergence of the Internet of Things paves the way for enhancing the monitoring capabilities of enterprises by means of extensive use of physical and virtual sensors generating a multitude of data. The main driving concept in sensing enterprises is the use of multi-dimensional data captured through sensors generating events and providing added value information that enhances context awareness (Engel et al., 2012; Camarinha-Matos et al., 2013). The large amount of sensor-generated data leads to a strong demand for data-driven, real-time systems capable of efficiently processing them, in order to get meaningful insights about potential problems. Proactive decision making requires context-awareness (Engel et al., 2011); however, the high frequency of the real-time events and the high uncertainty pose challenges to the efficient handling of context-awareness. This Section presents an approach that aims to enhance proactive event-driven decision making, by taking into account contextual information.

The proposed probabilistic model for context-aware proactive recommendations takes into account the cost risks according to the existing context and the prediction event received. It utilizes context awareness when there is uncertainty about the values of contextual elements in order to consider several contributing factors in the decision making process and to provide optimal recommendations. To do this, it utilizes Bayesian Network (BN), in order to represent the (uncertain) causal relationships between contextual information and cost functions, along with k-menas clustering for creating the values of the BN nodes. The proposed approach is embedded in a real-time, event-driven computational environment. Finally, data about the actual action implementation cost in different contexts, which are obtained through physical and virtual sensors during the actual execution of the recommended actions, are fed back to the context-aware model through SEF with the aim to close the loop and enable continuous learning. In this sense, the approach deals with probabilistic context. The deterministic context deals with Logic Based Models for representation with facts, expressions and rules or with Ontology Based Models for formal specifications of knowledge in order to take into account the user receiving the recommendation and parameters that affect the decision method output but are not inserted as input parameters (e.g. the customer requirements, the current business goals, the existing resources, etc.). These parameters are taken into account in the form of constraints and Event-Condition-Action (ECA) rules in the expected utility or loss maintenance function of the Decide phase.

In the following sub-sections, the approach of context-awareness in proactive decision making and its instantiation to maintenance operations is described. The approach and the algorithm address two blocks of the conceptual architecture: the "DMI Configuration" (as far as the context-aware model configuration is concerned) block of the User Interaction Layer and the "Context-aware Model" block of the Real-time Processing Layer. These three blocks are highlighted with red color in the conceptual architecture in Figure 7-1.

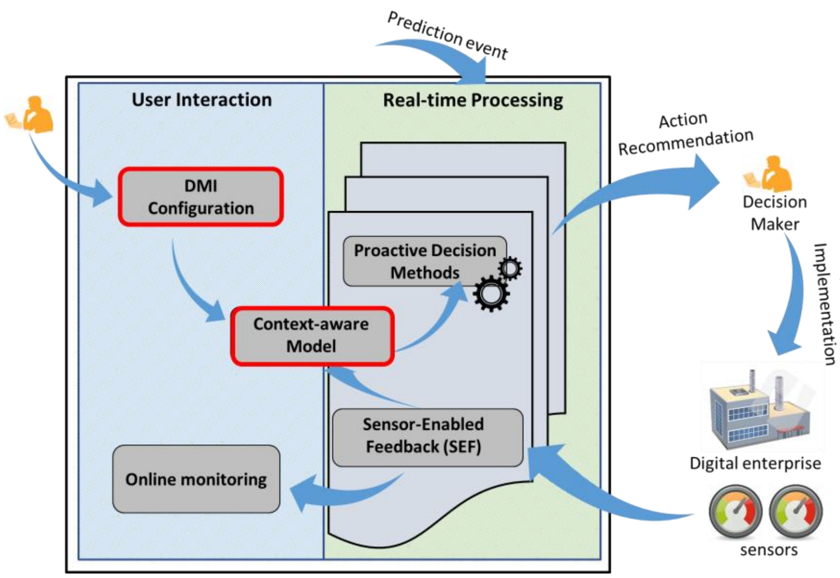

Figure 7-1: The context-awareness mechanism in the conceptual architecture.

7.2 State-of-the-Art Analysis

Context has been defined as "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves" (Perera et al., 2014). Context-aware systems are adaptable to the existing and future possible environments without the interactions of users (Lee et al., 2013) and process the context models based on the context lifecycle steps: acquisition, modelling, reasoning and dissemination (Perera et al., 2014; Schmidt et al., 2016).

The main driving concept in sensing enterprises is the use of multi-dimensional data captured through sensors generating events and providing added value information that enhances context awareness (Engel et al., 2012; Camarinha-Matos et al., 2013). The large amount of sensor-generated data leads to a strong demand for data-driven, real-time systems capable of efficiently processing them, in order to get meaningful insights about potential problems. Proactive decision making requires context-awareness (Engel et al., 2011); however, the high frequency of the real-time events and the high uncertainty pose challenges to the efficient handling of context-awareness.

Context-aware systems are adaptable to the existing and future possible environments without the interactions of users (Lee et al., 2013) and process the context models based on the context lifecycle steps: acquisition, modelling, reasoning and dissemination (Perera et al., 2014; Schmidt et al., 2016). However, research on con-text-aware systems has focused on reactive applications rather than proactive ones that could enrich proactive event-driven decision making. The ability to obtain, to process, to manage, and to provide relevant context information describing the envi-ronment and situation has become one of the most important requirements for information systems (Zaplata et al., 2013; Da Rosa et al., 2016). In addition to that, the prediction of future context is another important step for enabling devices and applications to also proactively support the user or to enable the desired automatic execution of his tasks even in dynamic environments (Mayrhofer, 2005; Zaplata et al., 2013).

However, research on context-aware systems has focused on reactive applications rather than proactive ones that could enrich proactive event-driven decision making. The ability to obtain, to process, to manage, and to provide relevant context information describing the environment and situation has become one of the most important requirements for information systems (Zaplata et al., 2013; Da Rosa et al., 2016). In addition to that, the prediction of future context is another important step for enabling devices and applications to also proactively support the user or to enable the desired automatic execution of his tasks even in dynamic environments (Mayrhofer, 2005; Zaplata et al., 2013).

Moreover, context-awareness has been considered in detection (Detect phase) and prediction (Predict phase) algorithms (Feng et al., 2009; Wan et al., 2014; Thaduri et al., 2014; Galar et al., 2015; Schmidt et al., 2016), but not in decision making algorithms and especially in proactive event-driven decision methods, where there is uncertainty about the future state of the system examined. Therefore, when a decision is required for a future situation, context is uncertain at this time. For this kind of problems, machine learning techniques are appropriate. Machine learning can be seen as a context modeling approach in terms of its objectives (Schmidt et al., 2016). It has been proved to be the best approach for intelligent context-aware systems (Thaduri et al., 2014), while it can be effectively coupled with relevant context reasoning techniques for supervised learning along with fuzzy and probabilistic logic. In this way, the sensor-generated heterogeneous and noisy data processing as well as the future state are taken into account.

Recently context awareness approaches is gaining focus of researchers from the field of CBM and predictive maintenance, however still at a conceptual level (Schmidt et al., 2016). This well-known concept in some other fields has not been investigated in Industry 4.0-enabled maintenance operations although it could be beneficial or even indispensable (Schmidt et al., 2016). Concepts related to context-awareness have not been utilized by researchers from predictive maintenance fields. This is also evident from various review and survey papers in the area where the term “context” in the frame of predictive maintenance is never directly mentioned (Schmidt, and Wang, 2015).

7.3 The probabilistic context-aware model for proactive decision making

The proposed approach aims to enhance proactive event-driven decision making by utilizing contextual information in the decision making process. Since proactive decision methods provide a recommendation about the optimal time of applying an action, the values of the contextual elements at the time when the system recommends the implementation of the action is subjected in high uncertainty, because it is not known in advance the recommended optimal time. Contextual information is propagated through SEF and coupled with domain knowledge in order to continuously improve the cost-related parameters of the decision methods and therefore, the generated action recommendations. At design time, during User Interaction, the decision maker inserts domain knowledge with the aim to define and configure the various parameters of the context-aware model. In this way, the model is initialized. This knowledge can be also inserted by integrating the system with the manufacturing company's own systems (e.g. ERP). The user inserts the context affecting the recommendation, i.e. the context affecting the associated cost parameters. Moreover, the user defines the probability of a contextual element's value occurrence, i.e. the prior probabilities. This can be obtained either by historical data or by domain knowledge and should be done only once, at the configuration, for the initialization of the context-aware model.

7.3.1 Context-aware Model Initialization

Context-awareness is treated with the use of a machine learning technique in order to effectively deal with uncertainty in a future context. Future context is not known in advance for two reasons: First, the conditions under which the system examined will function cannot be predicted with certainty. Second, the proactive decision model is triggered after the context-aware model and therefore, the recommended times of actions implementation are not known before the context prediction.

The context-aware model is initialized after the equipment instance configuration. It incorporates a Bayesian Network (BN), which is a powerful tool for knowledge representation and reasoning under conditions of uncertainty identifying the probabilistic relationships among a set of variables (Cheng et al., 2002). A BN has many advantages such as structural learning possibility, combination of different sources of knowledge, explicit treatment of uncertainty and support for decision analysis, and fast responses. The intensity of the dependencies is quantified by conditional probability distributions associated with each node (Korb and Nicholson 2010). More formally, BNs are directed acyclic graphs whose nodes represent random variables from the domain of interest, in the Bayesian sense (Heckerman, 1998).

Therefore, the network is defined by a pair $B = \langle G, \Theta \rangle$ where G is the directed acyclic graph whose nodes $X_1, X_2, \dots, X_n$ represent random variables, and whose edges represent the direct dependencies between these variables (Ben-Gal, 2007). The graph G encodes independence assumptions, by which each variable X_i is independent of its nondescendents given its parents in G. The second component Θ denotes the set of parameters of the network. This set contains the parameter $\theta_{x_{i|\pi_i}}=P_B(x_i|\pi_i)$ for each realization x_i of X_i conditioned on π_i, the set of parents of X_i in G. Accordingly, B defines a unique joint probability distribution over a set of random variables V, namely (Ben-Gal, 2007):

$$P_B(X_1, X_2, \dots, X_n) = \prod_{i=1}^{n} P_B(X_i|\pi_i) = \prod_{i=1}^{n} \theta_{X_i|\pi_i}$$

Based on the domain knowledge or the analysis of available historical data, the structure, the contents of the cause and the effect nodes as well as the prior probabilities are initialized in order to be used at the first recommendation of the instance, on the basis of a prediction event trigger. The BN is created based on the derived cause-effect (causal) relationships between the contextual elements and the alternative costs along with their prior probabilities for the decision horizon defined for the specific equipment instance. In this case, the BN provides the probability that a specific cost is valid conditioned a specific context expressed as cause nodes of contextual elements, according to the Bayes theorem, as shown in Figure 7-2.

Then, the Bayesian cost risk functions are estimated in order to be inserted in the proactive decision method instead of the cost functions themselves, when the DMI is enacted online. Cost risk indicates the probability of the occurrence of an event multiplied by its impact in cost (Hulett, 2016). A cost risk function is calculated by adding the products of each alternative value i of the cost function with the probability of having this cost function given m specific Contextual Elements (CE). Therefore, the cost risk functions (context-aware costs) are calculated based on the BN, according to Equation 7-1. The result feeds into the Reasoning sub-component of the Context-aware Model in order to be triggered by the next prediction event of the specific DMI.

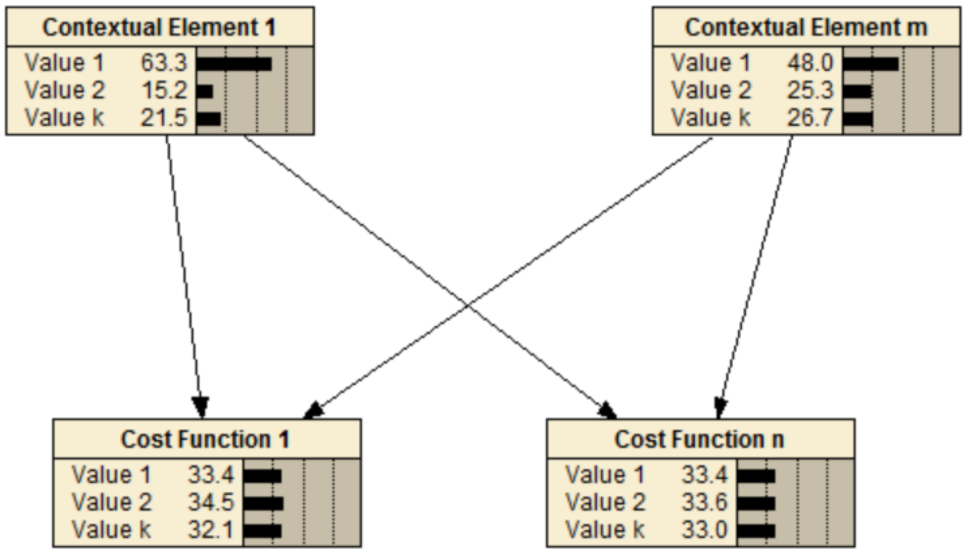

Figure 7-2: The Bayesian Network for the calculation of n expected cost functions with k alternative values conditioned m contextual elements with k alternative values.

Equation 7-1

$$C_n(t) = \sum_{i=1}^{i=k} C_{n,i}(t) * P\big(C_n(t) = C_{n,i}(t) | CE_1 \cap \dots \cap CE_m\big)$$

7.3.2 Context-aware Model Reasoning

Based on the input of the initialized context-aware model, the DMIs provide context-aware proactive recommendations. During the implementation of the recommended actions, the SEF mechanism is further utilized in order to update the structure of the BN, the content of its nodes as well as the associated conditional probabilities. The costs associated to future failures and mitigating actions are rarely simple to derive reliably by the user, during configuration, due to human subjectivity or unawareness of every aspect of the actual business situation. Moreover, reasoning of the context-aware model requires continuous learning and update. Therefore, incorporating a method for continuously improve the accuracy of proactive decision making input parameters is a critical aspect of the proposed model. After action implementation, the output of SEF feeds into the context-aware model reasoning in order to update the BN structure and to express the improved causal relationships between contextual elements (cause/ parent nodes) and alternative cost functions (effect/ child nodes) through Bayesian inference.

However, each updated cost value of the effect nodes may not be more reliable comparing to the previous one due to high inaccuracies in user's configuration or in noisy sensor measurements which prolong the model's learning duration. In addition, the rich information provided by a real-time feedback mechanism cannot easily feed into a BN, the nodes of which handle discrete or discretized values. To overcome these challenges, each cost variable in the effect nodes of the BN takes a cluster of values from the z last measurements along with their associated probabilities. The values are clustered to their relevant position and derive the most probable value (centroid) along with the associated centroid probability as a result of the X-means clustering algorithm (Pelleg, and Moore, 2000), an extension of k-means clustering algorithm. Therefore, each cluster consists of the costs with respect to their probability. In this way, context-awareness affects the cost-related parameters with respect to which proactive decision making is highly sensitive (reference), and enables the provision of more reliable recommendations by further filtering uncertainty in user's input, sensor measurements and event processing.

K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. The problem is computationally difficult (NP-hard); however, there are efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum, e.g. Lloyd's algorithm (Kriegel et al., 2017). The X-means clustering algorithm is an extension of k-means clustering, which is a method for finding clusters and cluster centers in a set of unlabeled data (Kanungo et al., 2002). K-means clustering algorithm suffers from three major shortcomings: it scales poorly computationally, the number of clusters K has to be supplied by the user, and the search is prone to local minima (Pelleg, and Moore, 2000). X-means algorithm refines cluster assignments by repeatedly attempting subdivision, and keeping the best resulting splits, until some criterion is reached (Pelleg, and Moore, 2000).

Figure 7-3 shows a BN in the effect nodes of which a X-means clustering algorithm is applied in order to create clusters of cost values. An example of the clusters in each effect node is shown in Figure 7-4, where the probabilities of the cost centroids should sum to 1. The context-awareness mechanism (BN incorporating X-means clustering in its effect nodes) has been developed as a generic method, in a modular way, in order to be able to be coupled with any proactive decision method and real-time feedback mechanism. On the basis of the centroids, the context-aware costs are calculated and feed into the proactive event-driven decision method. However, all the data including in each cluster along with the costs and the recommendations to which they lead are stored in the database and can be exposed to the user upon request.

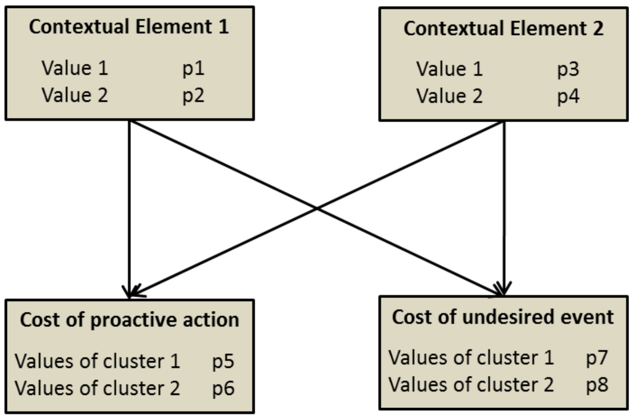

Figure 7-3: An example of the context-aware model incorporating a X-means clustering algorithm in the effect nodes of the BN.

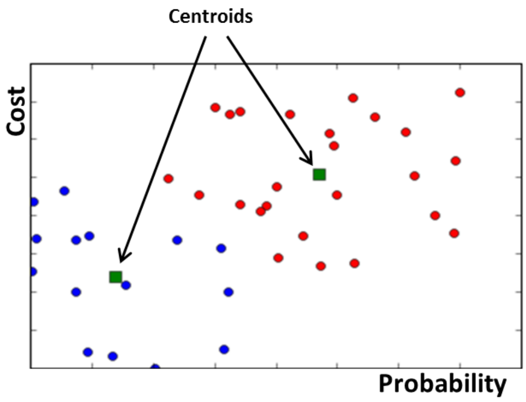

Figure 7-4: An example of the X-means clustering algorithm with two clusters in an effect node of the BN.

7.3.2.1 The Context-aware Reasoning Algorithm

Since sensor-generated big data require efficient and scalable real-time processing and the number of clusters may change as soon as the BN is updated, we take advantage of the use of X-means clustering, a method for dynamic determination of the number of clusters. In this way, the proposed algorithm refines cluster assignments. It consists of two main operations: the parameters improvement, which runs k-means algorithm until convergence; and, the clustering structure improvement, which finds out if and where new centroids should appear based on the splitting decision according to the Bayesian information criterion. In addition, a normalization equation for normalizing the probabilities of the clusters is embedded, while, after each loop, there is a control of convergence. This algorithm is shown in detail below. The steps of the algorithm are executed iteratively.

Improve parameters

It includes k-means clustering algorithm until convergence. According to the k-means clustering algorithm, given a set of observations (x_1, x_2, …, x_n) where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k (≤ n) sets S = {S_1, S_2, …, S_k} so as to minimize the Within-Cluster Sum of Squares (WCSS) (sum of distance functions of each point in the cluster to the k center). The algorithm is often presented as assigning objects to the nearest cluster by distance. The standard algorithm aims at minimizing the WCSS objective, and thus assigns by least sum of squares, which is exactly equivalent to assigning by the smallest Euclidean distance.

Consequently, its objective is to find:

$$\underset{S}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 = \underset{S}{\operatorname{argmin}} \sum_{i=1}^{k} |S_i| Var S_i$$

where μ_i is the mean of points in S_i.

Therefore, according to the k-means algorithm, given an initial set of k means $m_1^{(1)}$, $m_2^{(1)}$, …, $m_k^{(1)}$, the algorithm proceeds by alternating between two steps:

Assignment step: Assign each observation to the cluster whose mean yields the least WCSS. Since the sum of squares is the squared Euclidean distance, this is intuitively the nearest mean:

$$S_i^{(t)} = \left\{x_p : \left\| x_p - m_i^{(t)} \right\|^2 \leq \left\| x_p - m_j^{(t)} \right\|^2 \forall j, 1 \leq j \leq k \right\}$$

Where each x_p is assigned to exactly one $S^{(t)}$, even if it could be assigned to two or more of them.

Update step: Calculate the new means to be the centroids of the observations in the new clusters:

$$m_i^{(t+1)} = \frac{1}{\left|S_i^{(t)}\right|} \sum_{x_j \in S_i^{(t)}} x_j$$

Improve structure

It identifies if and where new centroids should appear by searching the space of cluster locations and the number of clusters with the aim to optimize the Bayesian Information Criterion (BIC). The BIC scoring is used both globally (when the algorithm chooses the best model) and locally (in all the centroid split tests). Given the data D and a list of alternative solutions M_j with different values of k, the posterior probabilities $P(M_j|D)$ are used to score the solutions. In order to approximate the posteriors, the following equation is used:

$$BIC(M_j) = \hat{l}_j(D) - \frac{p_j}{2} * \log R$$

Where $\hat{l}_j(D)$ is the log-likelihood of the data according to the j-th solution and taken at the maximum-likelihood point, and p_j is the number of parameters in M_j which is derived from the sum of k-1 class probabilities, M*k centroid coordinates and one variance estimate.

The maximum likelihood estimate for the variance is:

$$\hat{\sigma}^2 = \frac{1}{R-k} * \sum_i (x_i - \mu_i)^2$$

The point probabilities are:

$$\hat{P}(x_i) = \frac{R_{(i)}}{R} * \frac{1}{\sqrt{2\pi}\hat{\sigma}^M} * e^{-\frac{1}{2\hat{\sigma}^2} * \left\| x_i - \mu_{(i)} \right\|^2}$$

The log-likelihood of the data is:

$$\hat{l}(D_n) = -\frac{R_n}{2} * \log(2\pi) - \frac{R_n * M}{2} * \log(\hat{\sigma}^2) - \frac{R_n - K}{2} + R_n * \log R_n - R_n * \log R$$

Normalize cluster probabilities

Due to the existing uncertainty in context-awareness, an additional equation is required to the system of equations that the algorithm solves. Specifically, a normalization equation is needed in each effect node with the aim to normalize the Bayesian probabilities assigned to the k cluster centroids so that they are summed to 1. Therefore, the equation is as follows:

$$\sum_{\forall k} P(C_k(t) | CE_1 \cap \dots \cap CE_m) = 1$$

Consequently, the probability of a $C_k(t)$ given the $CE_1 \cap \dots \cap CE_m$ is calculated as shown:

$$\frac{P(C_k(t) | CE_1 \cap \dots \cap CE_m)}{\sum_{\forall k} P(C_k(t) | CE_1 \cap \dots \cap CE_m)}$$

Control convergence

If k > k_{max}, stop and report the best-scoring model, else go to “Improve parameters”.

8 Information System

In this Chapter, the developed proactive event-driven information system is explained. The developed information system is called ProActive seNsing enterprise Decision configurator DAshboard (PANDDA) and addresses the Decide phase of the Proactive Maintenance framework. It incorporates the functionalities presented in Chapter 5, Chapter 6 and Chapter 7. It was integrated with systems addressing the various phases of Proactive Maintenance based on the framework presented in Chapter 4 in order to result in a unified information system for Proactive Maintenance.

8.1 System architecture and implementation

8.1.1 The Overall Proactive Maintenance Information System

The overall information system for Proactive Maintenance is addressed through the integration and unification of different tools and services addressing the various phases of the framework for Proactive Maintenance. The Proactive Maintenance system is able to integrate data provided by different sources, to evaluate the quantity and the quality of the dataset, to support efficient real-time data processing, to provide access to sensor data in a streaming context and this should be done in combination with huge past data and to take into account background knowledge.

The real-time processing layer of the architecture has been implemented as a Storm topology (https://storm.apache.org). Storm is a distributed data processing system which is based on elements organized in a topology and called spouts and bolts. Spouts, which are the entry points into the real-time processing layer, poll relevant data sources such as sensors and distribute the data further in the topology. Bolts, which are the processing elements, implement the Proactive Maintenance information-processing services. Bolts are interconnected with an internal pub/sub mechanism and communicate through messages called tuples. All the integrated components of the real-time processing layer are Storm compatible in order to facilitate distributed processing of sensor-generated big data with high speed and velocity and to allow the whole system to scale.

8.1.2 Overview of PANDDA

PANDDA, which addresses the Decide phase of the Proactive Maintenance framework, is a Python web-application developed using the web2py[10] framework. Web2py is an open-source web framework (released under the LGPL version 3 license) for agile development of secure database-driven web applications, written also in Python. It follows the Model View Controller (MVC) software engineering pattern. This pattern aims to the separation of the data representation (the model) from the data presentation (the view) and also from the application logic and workflow (the controller). The three-layered PANDDA system technical architecture is shown in Figure 8-1 and its main subcomponents are explained in detail in the following Sections. The presentation layer occupies the top level of the architecture and displays information related to services available on the web-based PANDDA configurator. Business analysts, who are the main users of the PANDDA Configurator GUI, access the system through a web-browser, login with their personal accounts, and are exposed to services allowing them to create one or more instances of decision making methods, as well as configure, monitor and assess their performance. The SEF functionality of PANDDA provides real-time feedback about the execution of (the recommended) actions by incorporating and processing data provided by sensors with respect to action execution costs. The default and most common behavior of web applications which rely only on the http/https protocol is to update the user interface (the web-page) by pulling data from the server when the user requests information by clicking HTML elements of it like buttons or links. However, the real-time feedback functionality of the new version of PANDDA requires real-time monitoring of events. In order to achieve real-time update of the user interface (without the intervention of the user) we utilize the server-push and event-based publish/subscribe capabilities of the WAMP[11] protocol. On the client-side (the web-browser) we use the Autobahn|JS[12] JavaScript implementation of the WAMP protocol and correspondingly, on the server side, we use

[10] http://web2py.com/books/default/reference/29/web2py
[11] http://wamp-proto.org
[12] http://autobahn.ws/js

the Crossbar.io[13] WAMP router. In order to incorporate better insight about the cost of the actions we send all cost-related data to a Graphite/Carbon server, a highly scalable real-time graphing system which is able to store thousands of time-series per second and compute metrics on them. The complexity of this infrastructure is not exposed to the end-user who can transparently access via a single web-page data coming from all the different sources (PANDDA, WAMP router, and Graphite Web-app) because an NGINX[14] reverse proxy is configured to intervene and translate all the URLs in order to make them appear as if they are coming from the same web-server.

The logic layer controls application functionality by performing detailed processing. The services exposing the functionality of decision methods are decoupled from the PANDDA Bolt, which is part of the ProaSense Storm topology. The PANDDA system based on predictions about future undesired events implements different proactive decision methods. Moreover, the system monitors the cost of recommended actions implementation through sensors by using the SEF mechanism in order to improve the recommendations it produces. In addition, the system provides real-time information to the user about the incoming cost data and the processed data the system computes based on them. Cost data (either baseline or action-related ones) from sensors arrive to the PANDDA data processing services (which are implemented as RESTful web-services) from Apache Storm or other sources. PANDDA processes them (by applying different types of filters) and then stores the results to PANDDA RDBMS. Then it publishes the results to the WAMP router and sends them to the time-series analysis services of Graphite/Carbon.

Finally, the *data layer* houses a relational database engine like MySQL, SQLite, PostgreSQL or Oracle RDBMS where the information needed by the main algorithms of PANDDA is stored and retrieved. The Graphite server has its own internal time-series datastore which is used to store data and create graphs about them and their metrics on user-request. This service can run on a separate machine or a Virtual Machine without slowing down the data-processing (e.g. Kalman filters and regression/curve fitting) performed in the main data processing services of PANDDA.

[13] http://crossbar.io
[14] https://www.nginx.com/

PANDDA is an event-driven system that addresses the Decide phase of the 'Detect- Predict- Decide- Act' methodology according to the architecture presented in Section 4.1. It has been implemented as a Python web-application developed using the web2py framework (http://web2py.com/books/default/reference/29/web2py). It consists of four main sub-components: the PANDDA GUI, the PANDDA Control spout, the PANDDA Bolt and the PANDDA Runtime Services. PANDDA GUI is referred to the User Interaction Layer of the architecture, while PANDDA Control spout, PANDDA Bolt and PANDDA Runtime Services are referred to the Real-time Processing Layer.

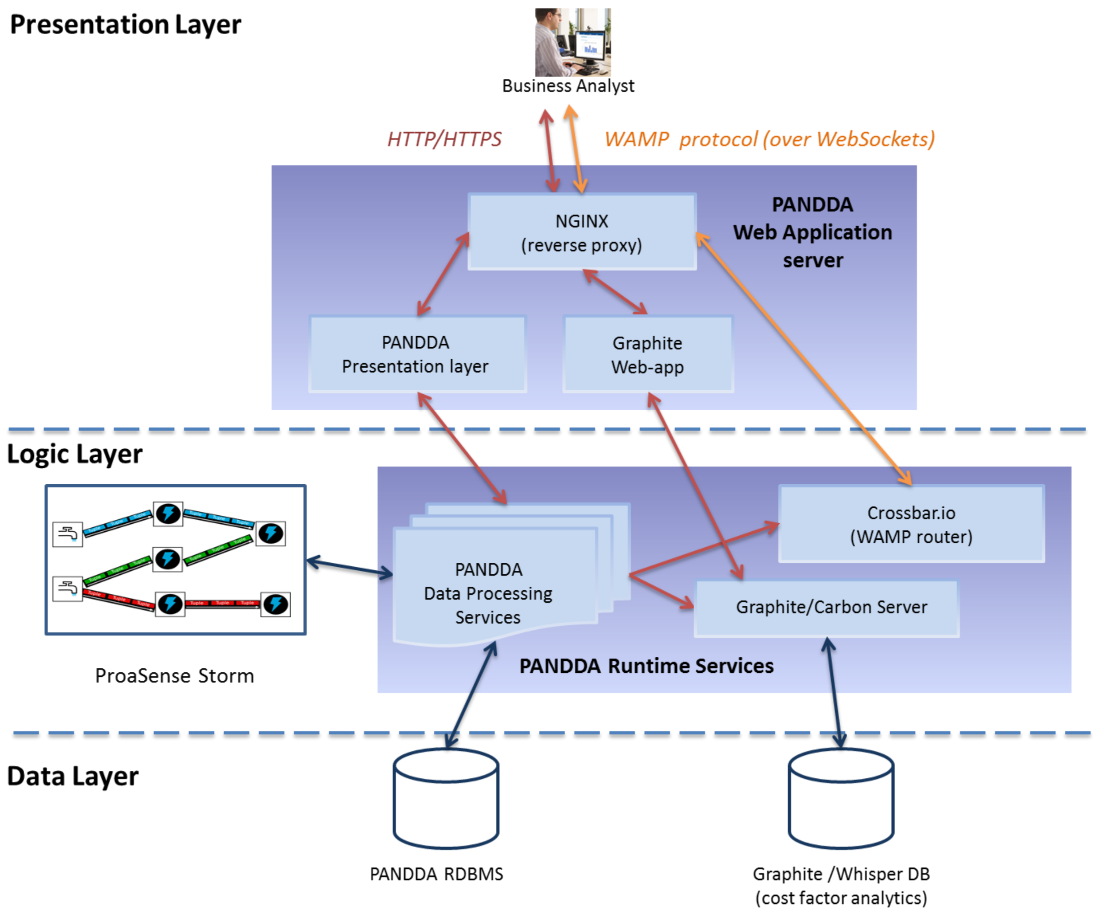

Figure 8-1: The PANDDA system technical architecture.

8.1.3 Logic Layer

The logic layer of Figure 8-1 implements the PANDDA Runtime Services which include the proactive decision making, the SEF and the context-awareness functionalities of PANDDA. These services and their runtime execution are presented in detail in the following sub-sections. A major architectural decision for PANDDA Runtime Services (Real-time Processing layer) was to decouple the PANDDA Bolt, which is part of the ProaSense Storm topology, from the services implementing the functionalities of context-aware decision management, depicted as PANDDA Runtime Services in Figure 8-1.

8.1.3.1 Proactive Decision Making

The main interactions among the sub-components of the PANDDA real-time processing layer are depicted in Figure 8-2. The PANDDA Bolt assumes the role of a proxy, as it forwards both the parameters of the decision methods defined by the business analysts (step 0 of Figure 8-2) and the parameters of the predicted undesired event (step 1 of Figure 8-2) to the PANDDA Runtime Services (step 2 of Figure 8-2). The latter extract and parse parameters from the received events, execute the functionality of the decision method instances (step 3 of Figure 8-2) and send the results back to the PANDDA Bolt (step 4 of Figure 8-2). The PANDDA Bolt generates in turn a recommendation message based on the results received, which is further propagated within the Storm topology and the rest of the ProaSense architecture (step 5 of Figure 8-2).

There are several distinct advantages of decoupling the PANDDA bolt from the service(s) implementing decision-making functionality, i.e. the PANDDA Runtime Services. First, the service(s) exposing the functionality of decision methods are decoupled from the implementation details of an Apache Bolt. In this way, the implementation flexibility is increased, as any technological platform, language and/or API can be used for the implementation of the various decision methods, allowing even the implementation of each one of them in different technological platforms, languages and/or APIs.

Second, there is no need to redeploy a PANDDA bolt and restart Apache Storm each time the configuration of some decision methods needs to be changed. On the contrary, the approach followed allows the PANDDA bolt to be configured at runtime through the PANDDA Control spout, as explained below. Third, implementation of decision-making methods as services allows their reusability in contexts different than a Storm topology, increasing their exploitation potential.

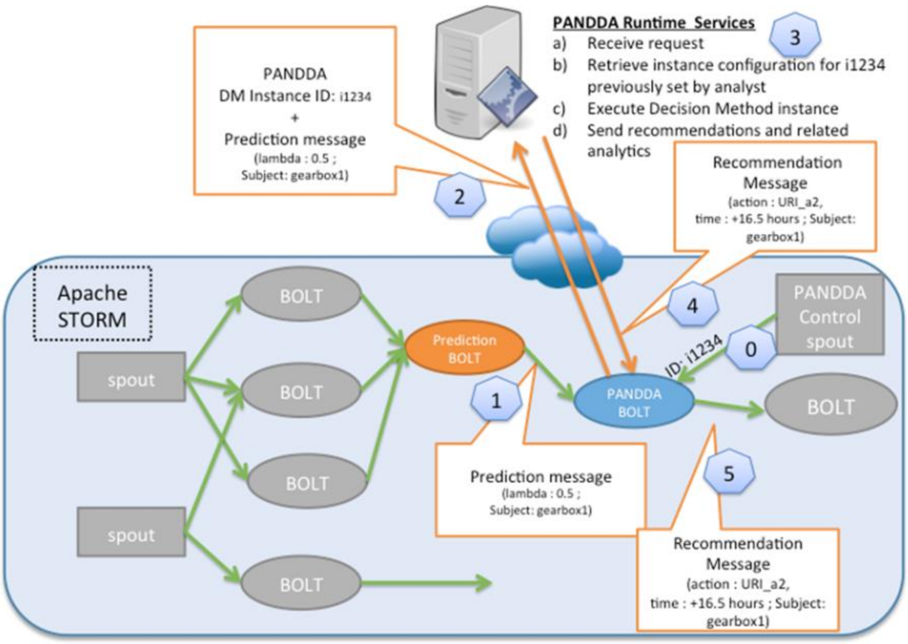

Figure 8-2: Interactions among the sub-components of the PANDDA Real-time Processing Layer

The PANDDA Control Spout is part of the ProaSense Storm topology of the Real time Processing Layer and is a component that assumes the role of a bridge between the PANDDA Decision Configurator Dashboard and the PANDDA Bolt, in the sense that it picks up the decision method configurations defined by business analysts through the former, and pushes them to the PANDDA bolt (step 0 of Figure 8-2). These two components can interact either during the initial setup and deployment of the Real time Processing Layer, during which the business analysts define decision method configurations, or during run-time, allowing reconfiguration of decision method instances through the PANDDA bolt with no need for bolt redeployment. Communication between the PANDDA Decision Configurator Dashboard and the PANDDA Control Spout is performed through Thrift messages, which are broadcasted to the broker by the former. Those messages are picked up by the PANDDA Control Spout, which deserializes and pushes them to the PANDDA bolt.

The sequence diagram of Figure 8-3 illustrates the main interactions among the subcomponents of the logic layer, as well as the interaction of the PANDDA decision making services and bolt with the other bolts of the ProaSense Storm topology. All depicted objects except of the PANDDA Decision Making Service reside inside an Apache Storm topology. The Online Decision Making (PANDDA) Bolt is a Storm Bolt that serves as a proxy between a Storm topology and the PANDDA decision making services.

The decision making methods of the PANDDA system take as input probabilistic predictions about future undesired events. The Online Analytics ProaSense component produces predictions about future events based on real-time events from sensors that may observe several different parameters of the environment (e.g. the temperature and the oil pressure of a motor or the utilization and the rate of errors in a network link). A prediction is encapsulated by Online Analytics in a PredictedEvent Apache Thrift object and emitted to the PANDDA Bolt as Storm a tuple. When a new tuple arrives to a PANDDA Bolt it must be deserialized (operation *deserilaze*). Then the PANDDA Bolt extracts information from the PredictedEvent (*eventName*, *subject*, *lambda*) and calls the external PANDDA Decision Making Service by calling the operation *getActionRecommendation*. The operation *getActionRecommendation* requires an additional field (named *instanceID*) which indicates the PANDDA decision making method instance that must be called. This information comes from the configuration of the specific PANDDA Bolt instance. Multiple instances of PANDDA Bolt can be used in the same or in different topologies.

The result of a call to *getActionRecommendation* is returned to the PANDDA Bolt. It contains information about a recommendation for an action that must be implemented at a specific time by a specific actor. All the information is encapsulated in a *Recommendation-Event* Apache Thrift message by the method *createRecommendationEvent*. This event is then sent to the Storage Layer of the PANDDA platform and is also emitted as a tuple in the Apache Storm.

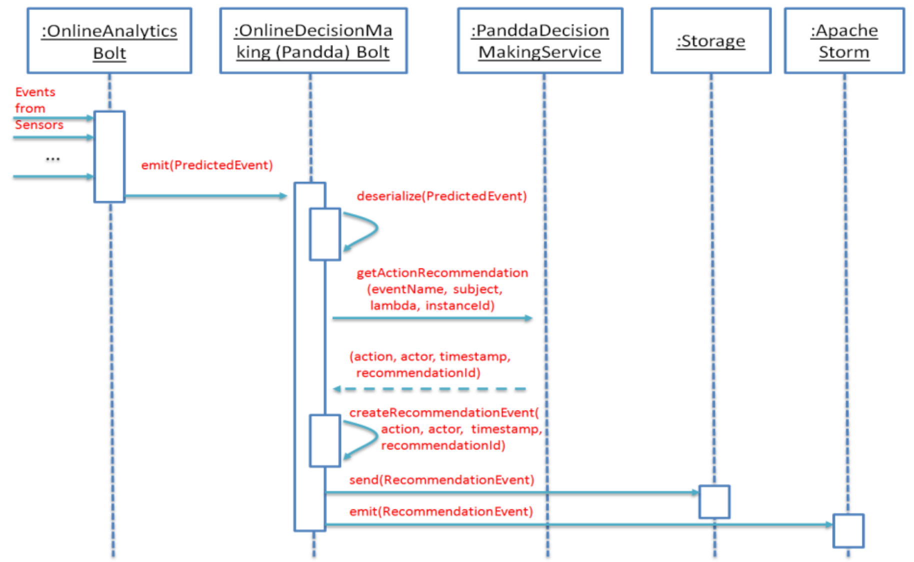

Figure 8-3: "Proactive Decision Making" functionality sequence diagram

8.1.3.2 Context-aware Model

The context-aware model incorporates the context modelling and context reasoning functionalities. Context modelling initialization is conducted at the presentation layer being triggered by the DMI configuration, while context reasoning is conducted at the logic layer being triggered by the SEF output. The contextual rules and constraints that are used at the expected maintenance utility or expected loss functions is modelled with an Ontology model based on the WWW SSN Ontology (Compton et al., 2012) according to an enterprise model that allows the consideration of the contextual elements affecting proactive decision making by establishing appropriate relationships between the contextual elements and the parameter values (Petersen et al., 2016). The context-aware model of PANDDA retrieves the appropriate information in the form of constraints and rules. On the other hand, the uncerain context is modelled with the use of BN and X-means clustering in order to catch uncertain causal (cause- effect) relationships between the contextual elements and the alternative input parameter values (e.g. a different cost due to the probability of a different context). In this case, context reasoning is conducted due to the Bayesian inference through SEF.

8.1.3.3 Sensor-Enabled Feedback (SEF)

The sequence diagram of Figure 8-4 illustrates the main interactions among the subcomponents of the logic layer, as well as the interaction of the PANDDA data processing services with the associated (cost- related) sensors, the internal components of the PANDDA system that process action- related cost data and the user's browser. It is complementary to the sequence diagram of Figure 8-3. SEF takes as input cost factor data derived from sensors related to the implementation of the (recommended) DMI action. The processing service of PANDDA retrieves action-related parameters from the RDBMS, applies different algorithms for action start/stop detection and cost factor noise filtering and then sends results to:

(a) The Graphite/Carbon service in order to produce graphs and metrics about the input and output data
(b) The WAMP router (implemented by Crossbar.io). The WAMP router publishes the results as events to all connected user browsers who have subscribed to listen for those events by visiting the page "View Real-time Action Cost".

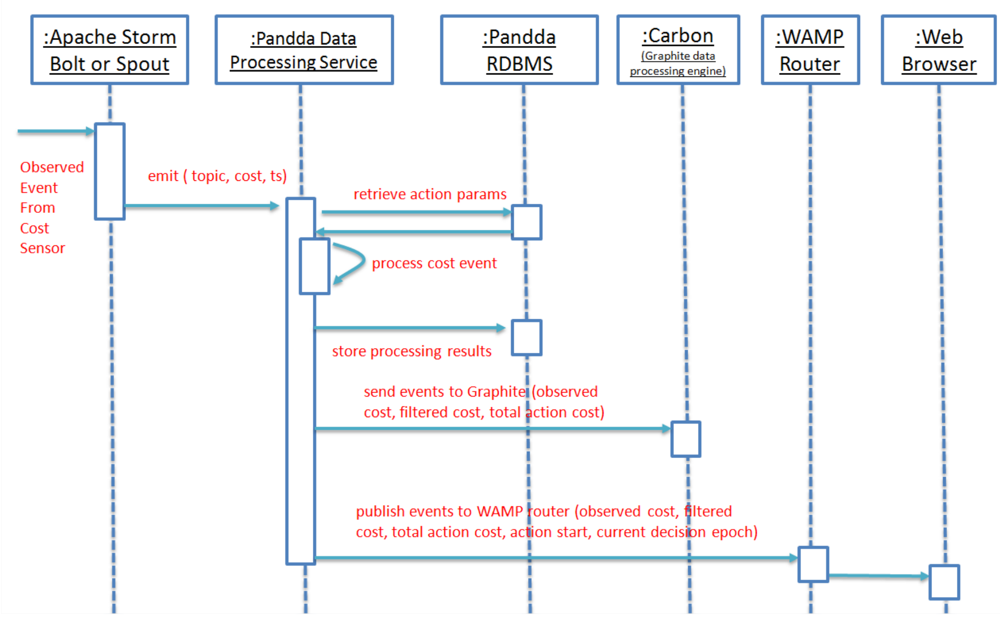

Figure 8-4: "SEF functionality" sequence diagram

The SEF functionality of the PANDDA data processing services is described in more detail with the UML activity diagram of Figure 8-5. This diagram illustrates the building blocks of the logic-layer which implements the feedback-enabled action recommendation. There are three types of events that trigger the relevant parts of the algorithm:

i. A cost event derived from a sensor related to a DMI action (events are related with action cost factors by a topic field).

ii. An event that denotes the end of a decision epoch. It is a periodic event that, in the current implementation, is derived and triggered by the system automatically based on the DMI start time and the DMI period (days, hours, or seconds) parameters.

iii. A prediction of an undesired event.

All these types of events are processed by PANDDA in parallel.

As can be seen in Figure 8-5, when a "Cost Sensor Event" arrives to PANDDA, the system calculates for each cost factor associated with the topic of this event the decision epoch period that it belongs (task "Calculate Decision Epoch Period Start") based on the timestamp of the event and the relevant DMI parameters (decision epoch start time, and decision epoch period).

- If it has been detected that the implementation of an action has started before the timestamp of the event from the associated cost factor-related sensor, PANDDA performs noise filtering (with Kalman filters) and then subtracts the baseline cost derived from this sensor. The baseline cost is the cost expected to be sent (periodically) when no action is performed (e.g. during normal operation, when no maintenance action is performed). The estimated cost of the action (due a specific cost factor) is calculated as the filtered (corrected) cost minus the baseline cost. This cost is accumulated to the total action cost and the result is stored in PANDDA DB as the current estimated total cost of the action.

-

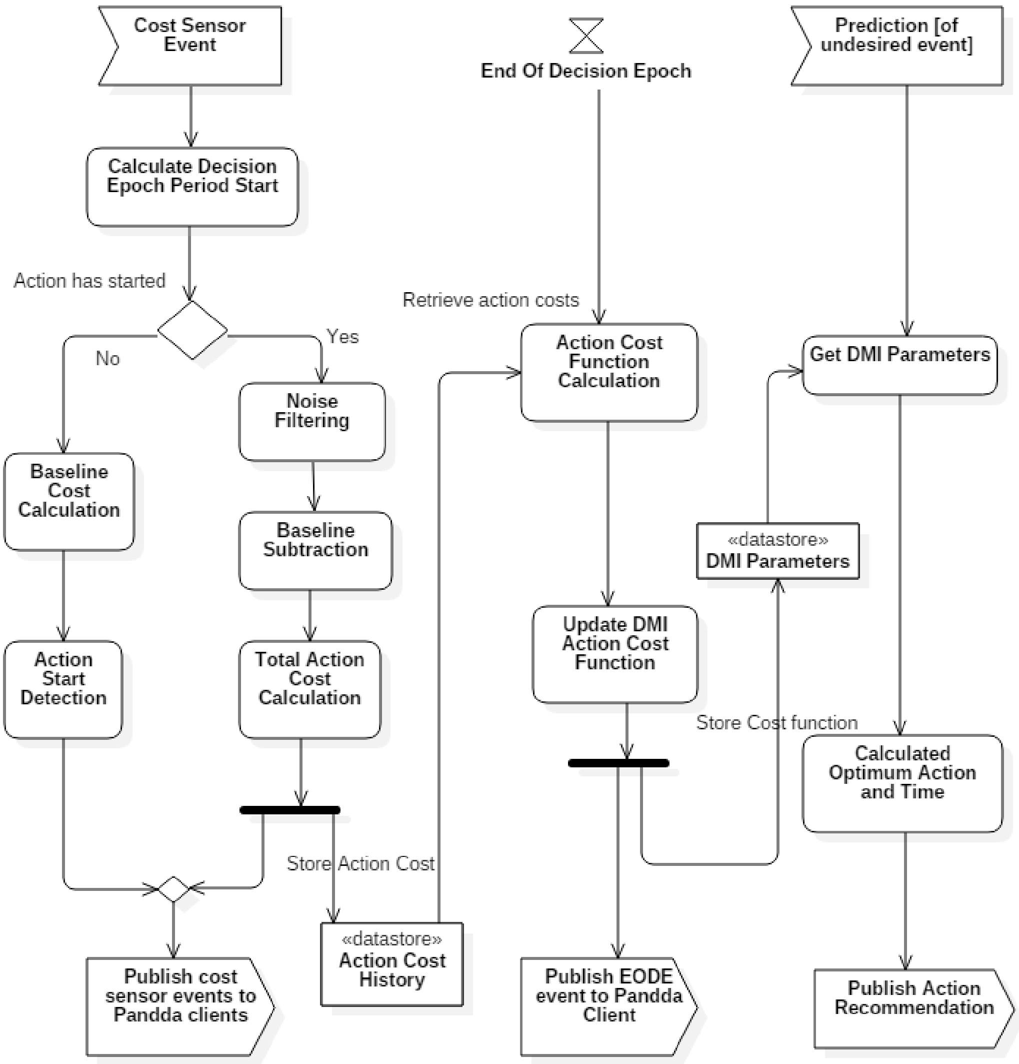

Figure 8-5: PANDDA data processing activity diagram

- If no action start has been detected (in the decision epoch period that the cost event belongs) PANDDA uses the cost event to update the baseline cost of each related cost factor (with a moving average filter) and then the system performs action start detection (by calculating anomalies from the baseline value).

In either case, when the processing of a cost event finishes (for each cost factor), PANDDA sends the results to Graphite/Carbon (in order to produce graphs) and publishes the input and output (from the calculations) events to the WAMP router. The WAMP router publishes those events to all subscribed client web-browsers.

When the time of an “end of decision epoch” event arrives, the system has already calculated the total cost of the related action(s) in the specific decision epoch (because the total action cost is being calculated incrementally). If there are enough observations about the total action cost in different decision epochs and different action start times before the end of the decision epoch, the user (or the system), can trigger the recalculation of the cost function of the specific DMI action by performing curve fitting (e.g. linear regression). The output of this task is a new cost function stored in the PANDDA DB. In any case, the system publishes the end of decision epoch event to the WAMP router.

As can be seen on the right side of Figure 8-5, when a new prediction of an undesired event arrives to the PANDDA system, it uses the stored DMI parameters to provide a recommendation by solving a MDP or one of the other two decision methods that are supported. Every time, it uses the most recent version of the cost function of each action. If an updated cost function has been calculated (by utilizing feedback during the previous steps) it uses it in the calculations. In this way, SEF contributes to the (automatic or semi-automatic) improvement of the accuracy and the efficiency (regarding to action cost minimization) of the provided proactive recommendations.

8.1.4 Data Layer

8.1.4.1 Proactive Decision Making

Figure 8-6 depicts a UML class diagram of PANDDA’s “proactive decision making” functionality data model. It serves as a dictionary of the terms that are used in the application and the relationships between them. The data model has been implemented using the Data Access Layer of the Web2py framework in combination with a relational database (the administrator can choose between MySQL, PostgreSQL, SQLite or other database engines). The main entities of the data model are briefly described in the next paragraphs.

Entity: *auth_user*. This entity holds the authentication and authorization data of PANDDA users. The schema of the entity is inherited from the Web2Py framework and for this reason it contains all the necessary fields to support many types of authentication methods.
All the entities inherit from the framework the field id. The fields *is_active*, *created_on*, *created_by*, *modified_on*, *modified_by* are also inherited from the web2py framework where they are needed and are maintained and updated transparently to the developer. The field *created_by* is used as the default filter in conjunction with the field id of the entity *auth_user* for all the entities containing it, in order to ensure that no user will be able to view data of other users. This mechanism is also implemented transparently to the software developer by the Web2py framework.
Entity: *dmm_instance*. The entity *dmm_instance* holds the data about the various instances of the decision making methods. Every decision making method instance has a name and is linked to one decision method (entity *dm_method*). The field *end_of_decision_epoch* holds the latest point in time that a decision can be made (usually the time of the next planned maintenance).
Entity: *dm_method*. The entity *dm_method* holds the information about the list of possible decision making methods and serves as a lookup table.
Entity: *mdp_instance_action*. The entity *mdp_instance_action* holds information about the actions of a *dmm_instance* implementing a Markov Decision Process method. Every MDP instance can have multiple possible actions. Every MDP action has a name and a delay (in days). The field *ttf_increase* denotes how many days the time to failure is expected to increase if the action is implemented. The cost of an action can be either fixed or daily (field *cost_factor_type*). In the first case, the field *cost_factor* contains the cost of the action, while in the second case the cost is calculated by the system as a function of *cost_factor* and the time that the action should be performed.
Entity: *mdp_instance_params*. The entity *mdp_instance_params* holds information about a *dmm_instance* that are relevant only for instances implementing the Markov Decision Process method. The field *ue_cost* contains the cost of the undesired event for the specific MDP decision making method.
Entity: *cbm_instance_action*. The entity *cbm_instance_action* holds information about the actions of a *dmm_instance* implementing a Cost Matrix Optimization method. Every CMO

instance can have multiple possible actions. Every action has a name and a delay (in days). The field *cm_cost* refers to the corrective maintenance cost of a specific action.

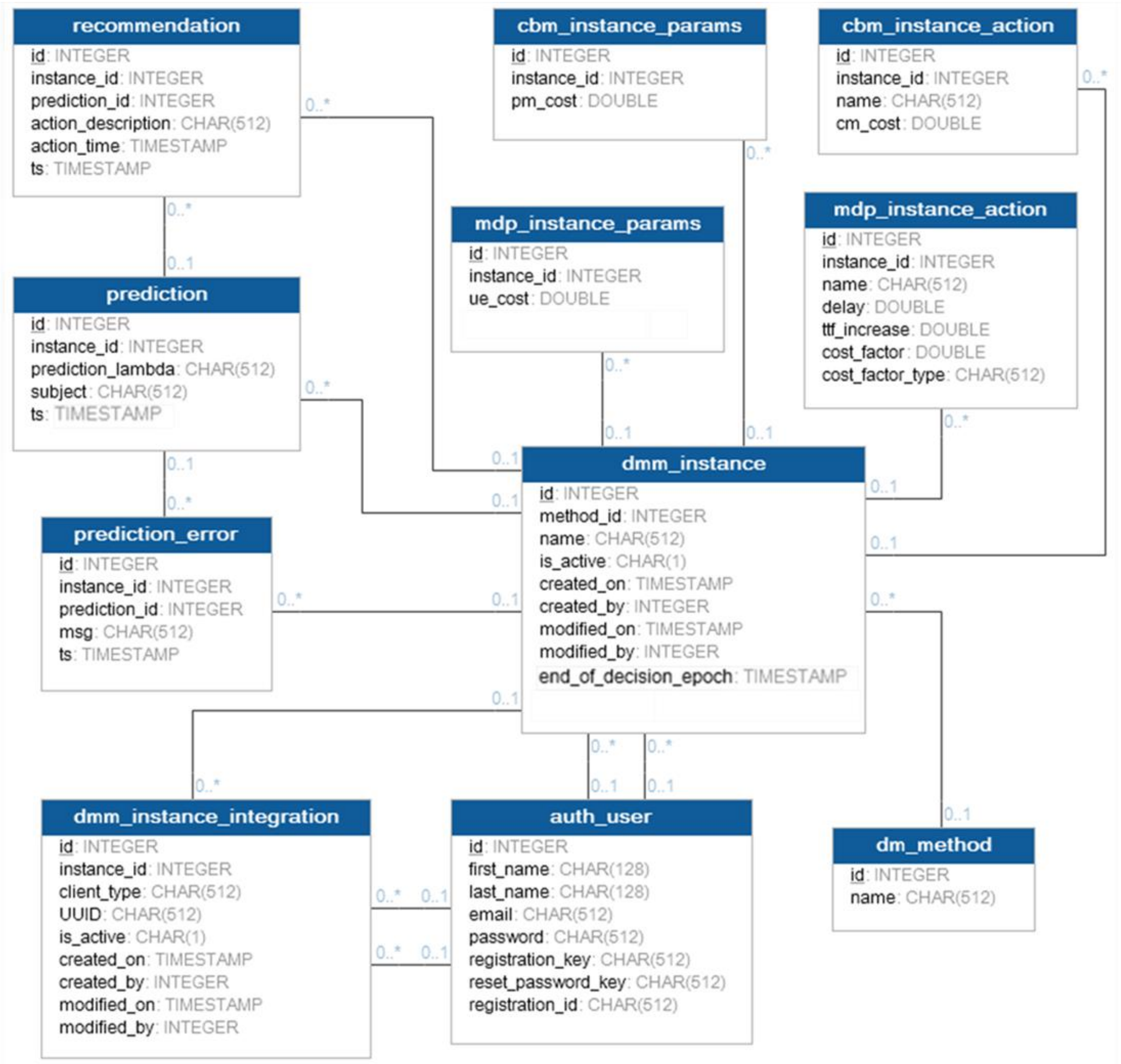

Figure 8-6: UML Diagram of the "Proactive Decision Making" functionality Data Model.

Entity: *cbm_instance_params*. The entity *cbm_instance_params* holds information about a *dmm_instance* that are relevant only for instances implementing the Cost Matrix Optimization method. The field *pm_cost* contains the cost of the planned maintenance.

Entity: *dmm_instance_integration*. The entity *dmm_instance_integration* holds the UUID of each decision making instance for each client type. If the decision method receives predictions from an Apache Storm topology the value of the field client type is "Apache Storm".

Entity: *prediction*. The entity *prediction* holds historical data about all the predicted undesired events arriving to PANDDA through the Apache Storm topology. It is linked with a specific *dmm_instance*.

Entity: *prediction_error*. The entity *prediction_error* holds historical data about all the prediction errors occurring during the processing of predicted events. It is linked with a specific *dmm_instance* and a specific *prediction*.

Entity: *recommendation*. The entity *recommendation* holds historical data about all the recommendations generated by the subcomponents if the PANDDA logic layer. It is linked with a specific *dmm_instance* and a specific *prediction*. It contains information about the recommended action (field *action_description*) and the point in time that the specific action should be performed (field *action_time*).

8.1.4.2 Sensor-Enabled Feedback (SEF)

Figure 8-7 depicts the UML class diagram of the SEF functionality data model. Based on the collected action-related cost data, SEF estimates the action cost function relative to the specific point of time that the action is implemented after the prediction (or before the end of the decision epoch). The main entities of the updated part of the data model are described in the next paragraphs.

Entity: *auth_user*. This entity holds the authentication and authorization data of PANDDA users. The schema of the entity is inherited from the Web2Py framework and for this reason it contains all the necessary fields to support many types of authentication methods. This entity has not changed in PANDDA v2 but now it is related with an additional entity (*cost_factor_topic*).

Entity: *dm_method*. The entity *dm_method* holds the information about the list of possible decision methods and serves as a lookup table. This entity has been populated with the name and the id of the new SEF-enabled method.

Entity: *dmm_instance*. The entity *dmm_instance* holds the data about the various DMIs. Every DMI has a name and is linked to one decision method (entity *dm_method*). The field *end_of_decision_epoch* holds the latest point in time that a decision can be made (e.g. the time of the next planned maintenance). When field method_id points to the new SEF-enabled method the system retrieves action and method parameters from the entities *fdbk_mdp_instance_params and fdbk_mdp_instance_action.*

Entity: fdbk_*mdp_instance_params*. The entity *fdbk_mdp_instance_params* holds information about a *dmm_instance* relevant only for instances implementing the new SEF-enabled method. The field *ue_cost* contains the cost of the undesired event for the specific method.

Entity: fdbck_*mdp_instance_action*. The entity *fdbk_mdp_instance_action* holds information about the actions of a *dmm_instance* implementing the new SEF-enabled method. Every MDP instance can have multiple possible actions. Every MDP action has a name and a delay (in days). The field *ttf_increase* denotes how many days the time to failure is expected to increase if the action is implemented. The cost of an action can be either fixed or a function of time and is coded with the coefficients a,b,c of the cost function $C\ (t) = a^2t + bt + c$. Variables a, b, c are stored in the fields *var_a, var_b, var_c*.

Entity: *cost_factor_topic*. The entity *cost_factor_topic* holds the information about the list of topics of cost events derived from sensors. For each topic the user (or the administrator) has to provide two parameters, the process noise (kalman_q) and the sensor noise (kalman_r) which are parameters of the Kalman filter that will be applied for noise filtering.

Entity: fdbck_*mdp_instance_action_cost_factor*. The entity *fdbk_mdp_instance_action_cost_factor* holds information about the cost factors of an action. Each cost factor is related to a cost_factor_topic (field: topic) and it can have a textual description (field: descr). The field cf_type indicates the cost factor type (possible values on of the “Constant”, “First Degree Polynomial” or “Second Degree Polynomial”). The cost function of an action is coded with the coefficients of the function $C\ (t) = a^2t + bt + c$, where the coefficients a, b, c are stored in the fields var_a, var_b, var_c. Depended on the value of the field *cf_type,* the user is requested to provide (real number) values for the fields var_a, var_b, var_c. The field *R* is a parameter of the Kalman filter (sensor noise). The field *bs_mean_events* stores the (maximum) number of the cost events used to calculate the moving average.

Entity: fdbck_*mdp_instance_action_feedback*. The entity *fdbk_mdp_instance_action_feedback* holds information about the timestamp (*field: ts*) of an action start (*field: event_type=”START”*) or action end event (*field: event_type=”END”*).

Entity: fdbck_*mdp_instance_action_cost_func*. The entity *fdbk_mdp_instance_action_cost_func* holds information about the total observed cost (field: *total_cost*) or the total estimated cost (field: *total_est_cost*) of an action instance in a specific decision epoch. The field *action_start_ts* holds the timestamp of an action start. Each action can occur in multiple decision epochs. The fields *epoch_no, epoch_start, epoch_end_ts* contain the corresponding information about the decision epoch when the specific action instance occurred.

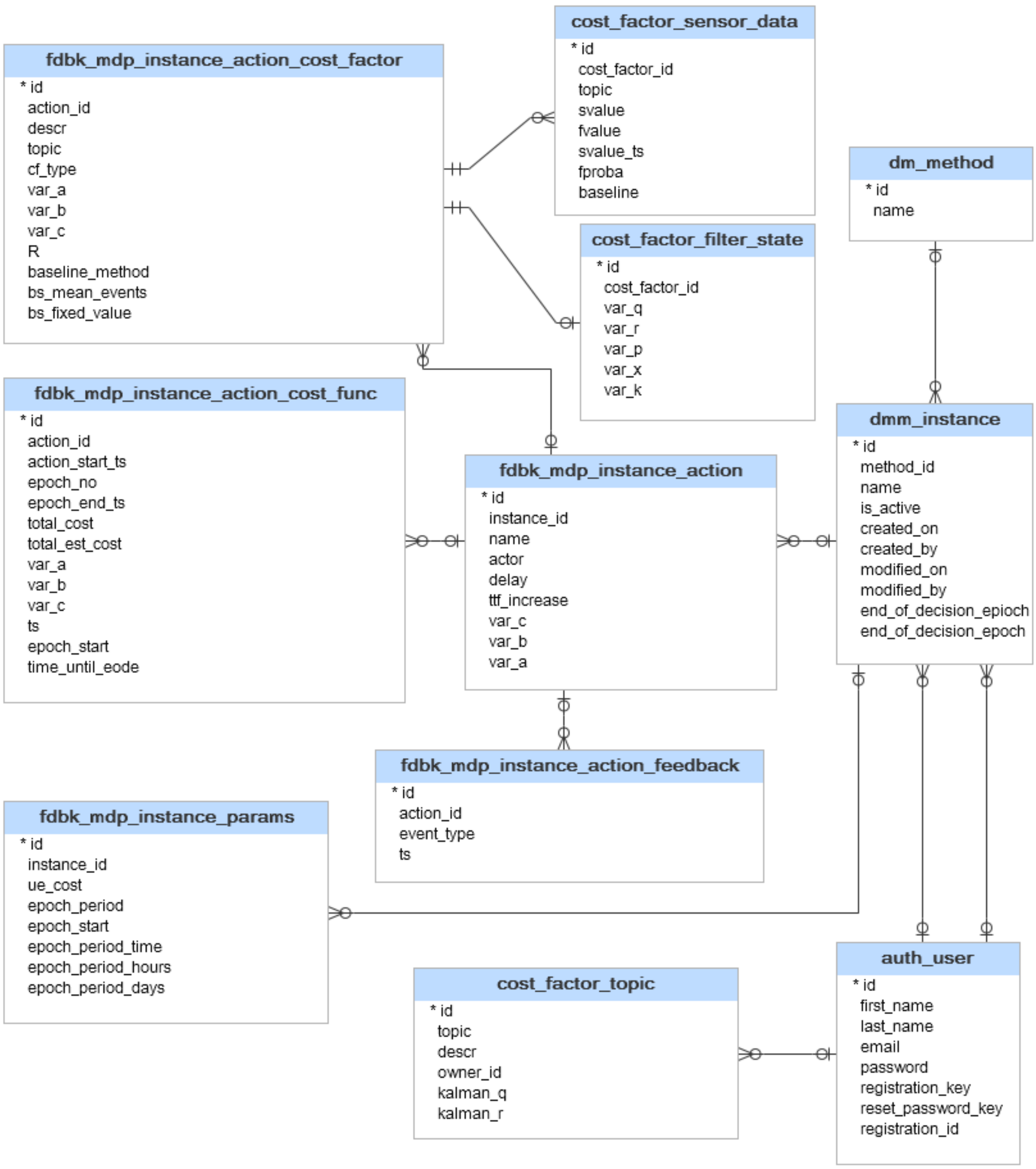

Figure 8-7: UML diagram of the SEF functionality Data Model.

Entity: *cost_factor_sensor_data*. The entity *cost_factor_sensor_data* holds information about cost factor processing. The field *cost_factor_id* is a foreign key that links to an action cost factor. The field *topic* contains the topic of a cost event. The field *svalue* contains the observed (noisy) value as it was reported by a sensor. The field *svalue_ts* contains the timestamp of the cost event derived from a sensor. The field *fvalue* contains the estimated (corrected) or filtered cost value based on the sensor data. The field *fproba* indicates how confident the system is about the estimation of the svalue. Finally the field *baseline* contains the baseline cost value the system has calculated for the specific cost factor at the specific timestamp (*svalue_ts*).

Entity: *cost_factor_fiilter_state*. The entity *cost_factor_filter_state* holds information about the current state of the (Kalman) filter of each cost factor.

8.1.5 Presentation Layer

Figure 8-8 and Figure 8-9 depict PANDDA's information architecture. They describe how the different graphical elements and web pages of the PANDDA web-application relate to one another and provide an overview of how the information presented in the PANDDA web-application is organized, structured, and labelled. The boxes with red letters depict the web-pages added in the second version of PANDDA.

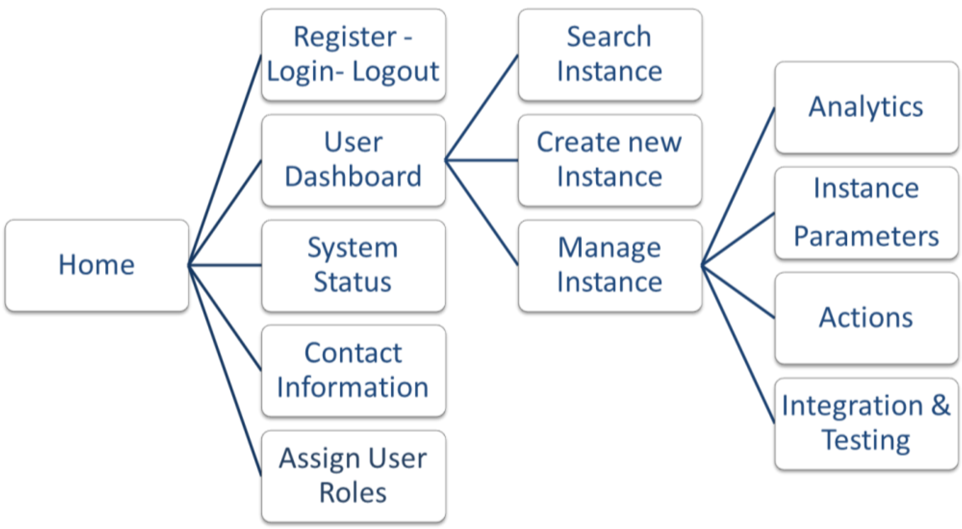

Figure 8-8: PANDDA Information Architecture (a).

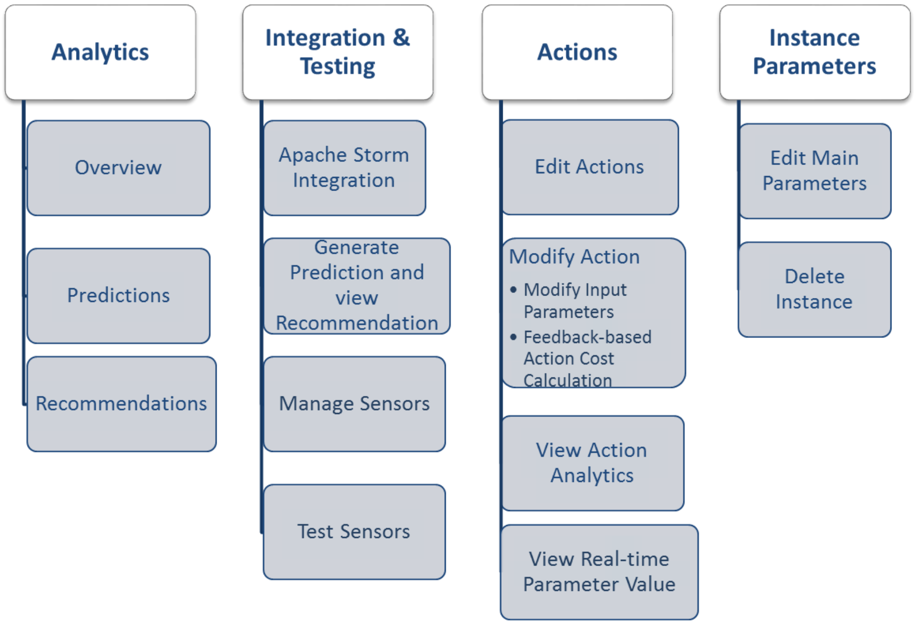

Figure 8-9: PANDDA Information Architecture (b).

8.2 PANDDA User Interface and Walkthrough

PANDDA (ProAseNse Decision configurator DAshboard) aims to enable business analysts embed the domain knowledge needed for generating recommendations of action-time pairs by using different decision methods. Specifically, PANDDA focuses on enabling business analysts to create and configure decision method instances addressing the problem at hand, as it is predicted in terms of a future undesired event (e.g. breakdown of a specific manufacturing equipment). Decision method instances are specific instances of the decision methods supported by PANDDA. Each decision method instance corresponds to specific equipment or other subject of a predicted undesired event, which triggers during runtime the decision method that aims to mitigate it. Decision method instances are treated as first class citizens in the PANDDA configurator in the sense that the user interaction with the tool has been designed on the basis of them.

Through the PANDDA GUI, business analysts can configure decision method instances by adding, removing or changing a mitigating action or a list of mitigating actions as well as other domain knowledge required by the method such as the cost of the undesired event, costs of mitigating actions, end of the decision epoch (e.g. time of next planned maintenance), etc. Decision methods are then enacted by the online decision making component, which, coupled to the ProaSense real-time architecture, generates timely and reliable mitigating action recommendations based on the predictions for undesirable situations derived on the basis of streaming data. PANDDA also focuses on improving the recommendations through a Sensor-Enabled Feedback (SEF) loop, which takes into account real-time data and improves the parameters of the underlying decision methods. Therefore, the role of SEF is twofold: (i) The user is informed online about the estimated cost of action during its implementation, and (ii) The updated cost function of the specific action is used in the next recommendation in which this action is involved.

So, PANDDA is a tool that is used at design time by business analysts and allows them to define and configure, through the PANDDA Graphical User Interface (GUI), various parameters of the decision method instances. Decision method instances are treated as first class citizens in the PANDDA configurator in the sense that the user interaction with the tool has been designed on the basis of them. For example, the main screens of PANDDA configurator allow the business analysts to create, view, search, manage and configure decision instances. The role of the various pages and other graphical elements of the PANDDA information architecture are explained in the next sub-sections which present the PANDDA user interface by considering typical user interaction sequences. The URL of the PANDDA system is: https://snf-542682.vm.okeanos.grnet.gr/pandda_v2_2/default/index.

8.2.1 Creating Decision Method Instances

The initial screen that a user sees when accessing the PANDDA application is the one shown in Figure 8-10. In order to have access to the PANDDA system, they have to click on the ‘Enter your dashboard’ button of the initial screen.

They can create a new decision method instance, which means that they can select a decision method for a specific part of equipment (e.g. gearbox) or other subject of a predicted undesired event as well as all the accompanied information required (list of actions, costs of actions, cost of the undesired event, end of the decision epoch, etc.). The user input is not identical for all the decision methods supported by PANDDA; therefore different knowledge needs to be embedded in the system according to the decision method selected. The users also insert the contextual information that is required by the context-aware model functionality.The PANDDA system can show all the instances that have been created so that the users are able to choose one of them to apply or to see more details (e.g. about the activity of the last 30 days). The relevant screen 'My instances' is shown in Figure 8-11.

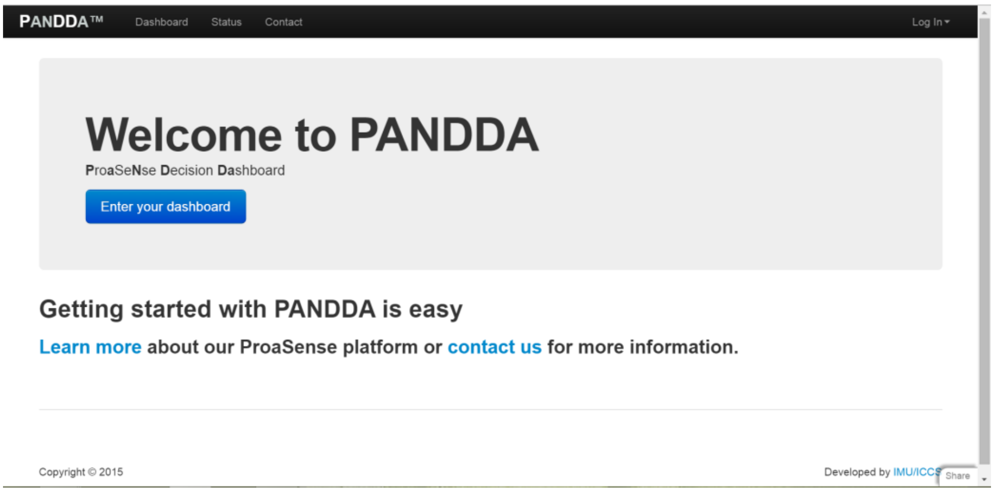

Figure 8-10: The initial screen of PANDDA

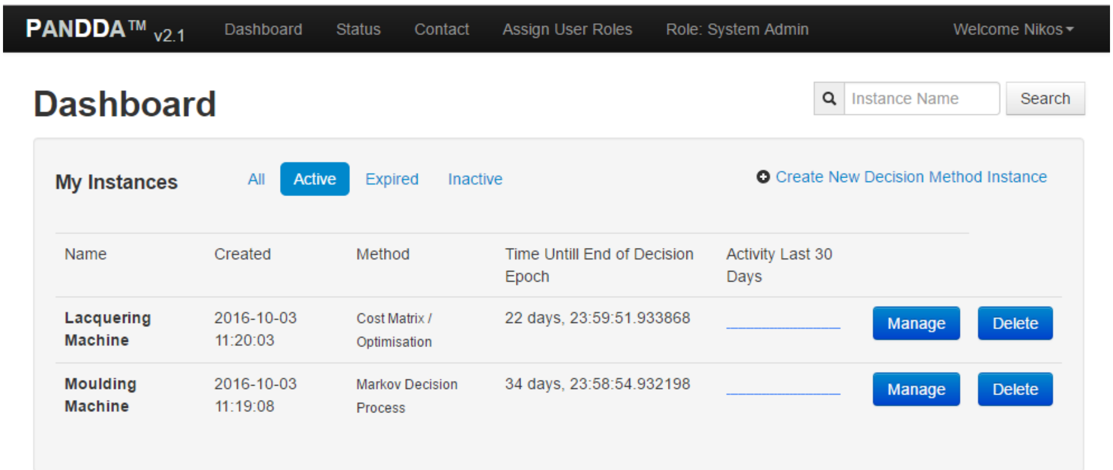

Figure 8-11: The 'My instances' screen of the PANDDA system

By pressing the button 'Create New Decision Making Method Instance', they are guided to the screen of Figure 8-12 where they can select one proactive decision method and whether they need the context-aware model and the SEF functionalities. Moreover, in this screen, they can select the name of the current instance indicating the part of equipment or other subject of the predicted undesired event that it refers to. Finally, they can select the time of the next planned maintenance, or the end of the decision epoch in the general case (decision horizon), as shown in Figure 8-13. After having finished this procedure, the button 'Submit' should be pressed in order to return to the 'My instances' screen, which has been updated with the new instance that has been introduced. In the rest of this walkthrough, user interaction with the PANDDA GUI is explained for the Markov Decision Process with Action Feedback method, but a similar procedure can be followed for the other two methods.

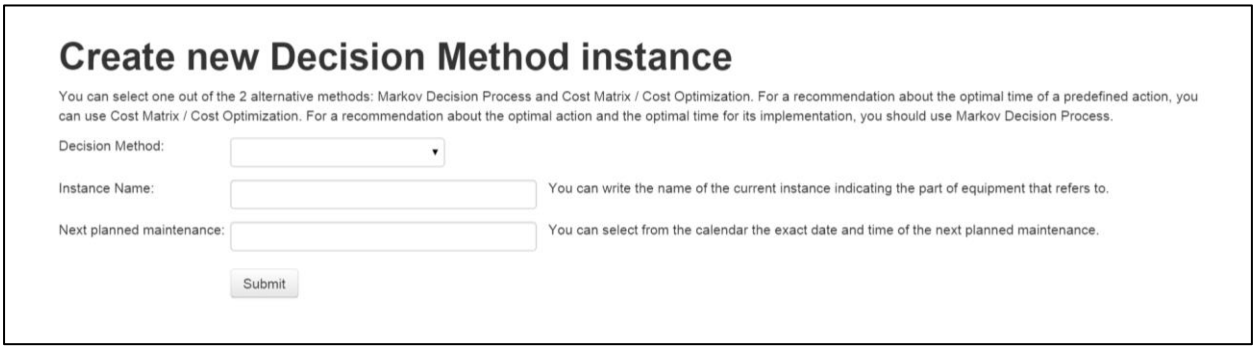

Figure 8-12: 'Create New Decision Method Instance' screen

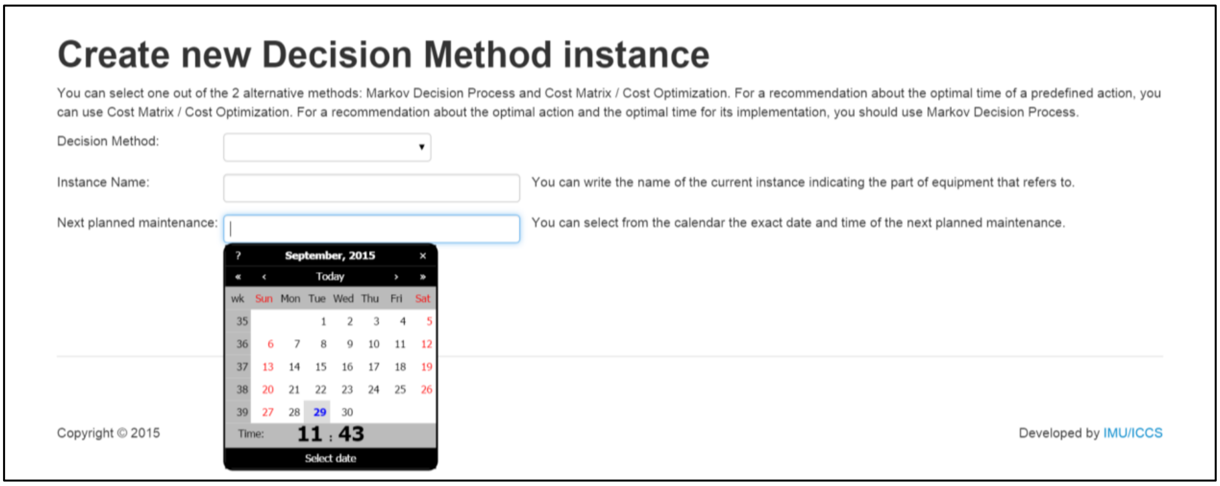

Figure 8-13: Selection of time of the next planned maintenance

The same procedure is followed for each instance needed to be created and finally, a list of instances is shown. For example, in Figure 8-14, three decision method instances have been created for the Gearbox, and the Moulding and Lacquering machines. Each one of them corresponds to a specific decision method for a specific part of equipment. Each row of the list consists of the name of instance (e.g. Gearbox), the date of its creation, the time remaining until next planned maintenance and a summary of analytics regarding the activity of the last 30 days. Next to it, there is a 'Manage' button, which leads to the screen of Figure 8-15, and a 'Delete' button. The screen of Figure 8-15 consists of four components, which will be further analysed: Analytics, Actions (Decision Alternatives), Integration / Test and

Decision Method Settings. The text in orange fond below Actions (Decision Alternatives) and Decision Method Settings components indicate that no actions and values of decision method parameters have been defined yet.

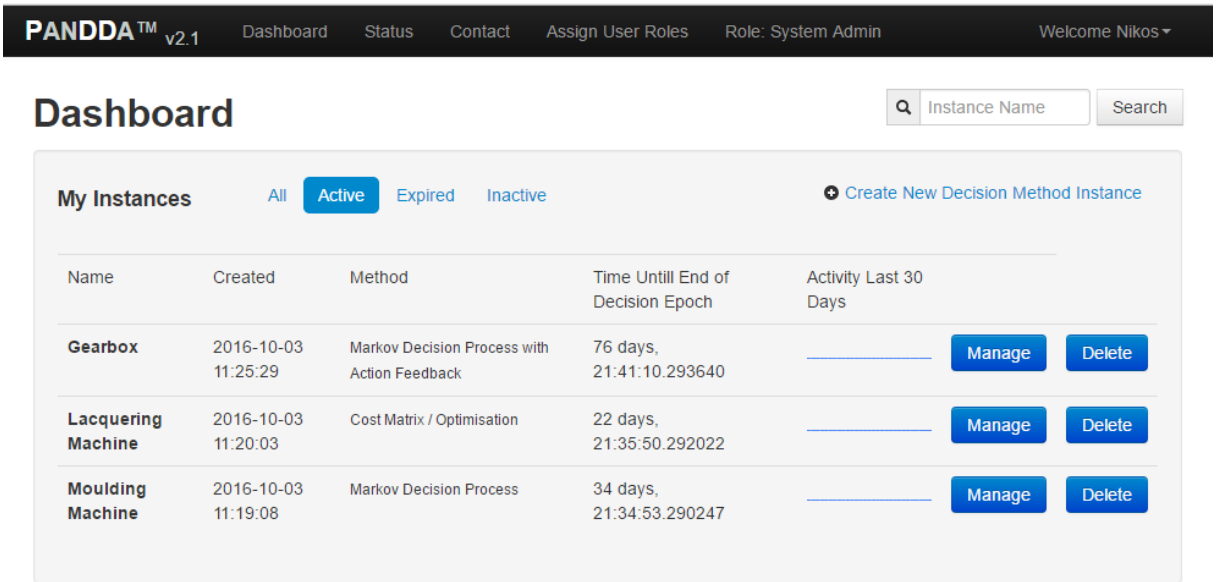

Figure 8-14: List of three instances

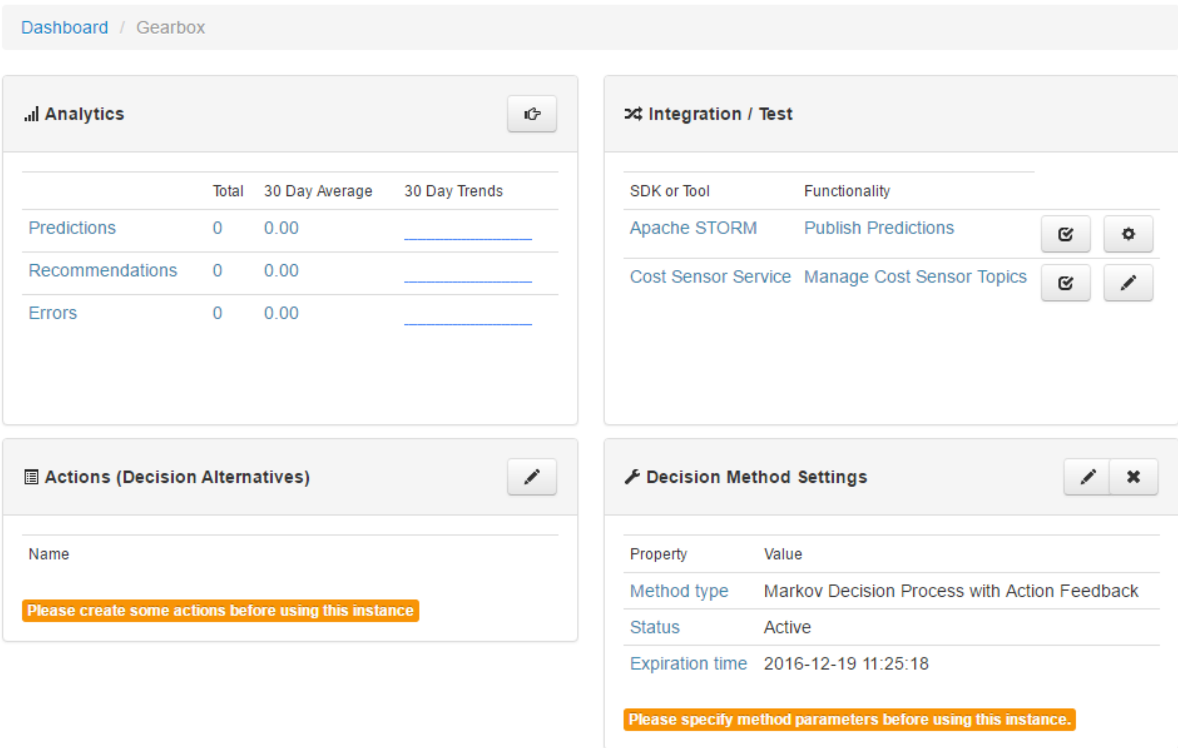

Figure 8-15: Instance management screen

8.2.2 *Managing Decision Method Instances*

The Actions (Decision Alternatives) component of the instance management screen is used so that you embed domain knowledge regarding the alternative actions that can be recommended. In the "Edit Instance Action", the user can add, edit and delete alternative actions. By pressing the 'Edit' button and then, by selecting the 'Add action' option, they are led to the Actions (Decision Alternatives) screen of Figure 8-16. Figure 8-16 shows an example of editing an action. The users can write the name of the action (e.g. operate at reduced equipment load), the role of the person that must perform the action, the delay of the action in hours (corresponding to the time period from its implementation until it starts taking effect) and the expected new time-to-failure after the implementation in hours. They can also insert the contextual elements affect the cost functions and their prior probabilities. In addition, they can edit the cost factors, the aggregation of which formulates the action cost function. To do this, they should click on the "Add Record" button (below "Edit Cost Factors) in order to add a cost factor or on the "Edit" button next to an existing cost factor. Moreover, you are able to see the details of each cost factor or to delete the ones that are not needed. In the example of Figure 8-16, the total action cost function is a linear function equal to 420 * t + 3100, since a, b and c are referred to the factors of an equation $a * t^2 + b * t + c$.

Figure 8-17 shows an example of a cost factor. In this case, the description of the cost factor inserted by the user is "Cost due to production loss" and the topic name that has been defined at the "Manage Topics" screen is "production loss". This topic has been previously mapped to a specific sensor by the System Administrator. Then, the user inserts the "Cost Factor Type", i.e. the polynomial order of the cost factor function, and the coefficients of the cost factor function. In the current example, the cost factor has been selected to be a first order polynomial and the coefficients b = 315 and c = 2325. Therefore, the cost factor function is 0*t^2+315*t+2325. Finally, the user confirms the sensor noise given during topics management by System Administrator and click on the "Submit" button and then on the "Set action cost from cost factor initial a, b, c" button, so that the changes are applied. After following the same procedure for all the cost factors that correspond to the specific action cost function, the fields that show the coefficients of the action cost function are automatically completed. In this example, the action “Operate at reduced equipment load” consists of two cost factors and their aggregation gives an action cost function of 420 * t + 3100.

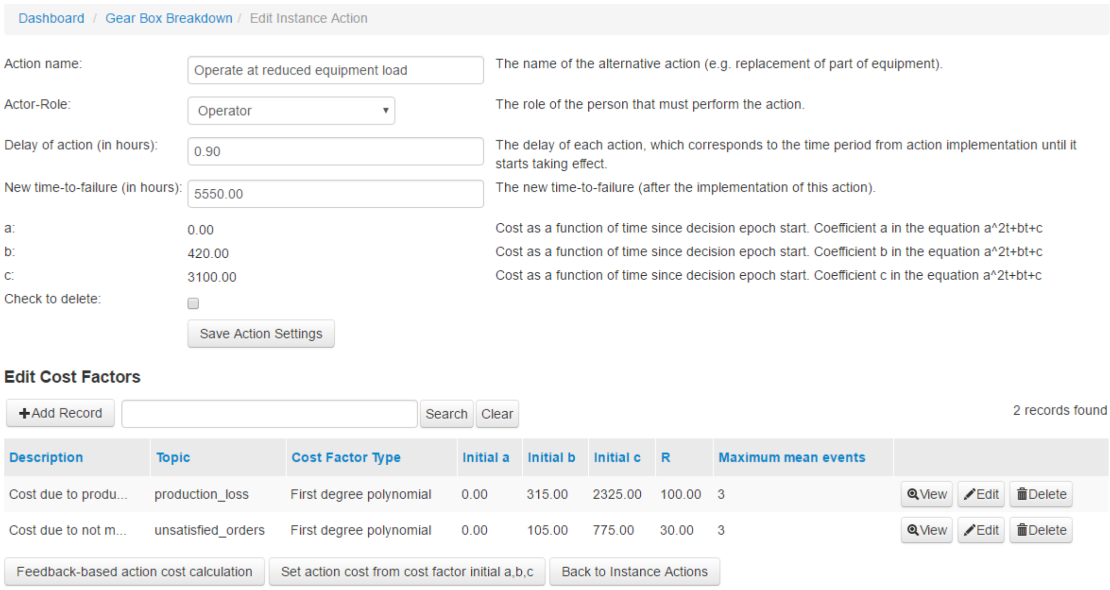

Figure 8-16: The “Edit Instance Action” screen for a specific action.

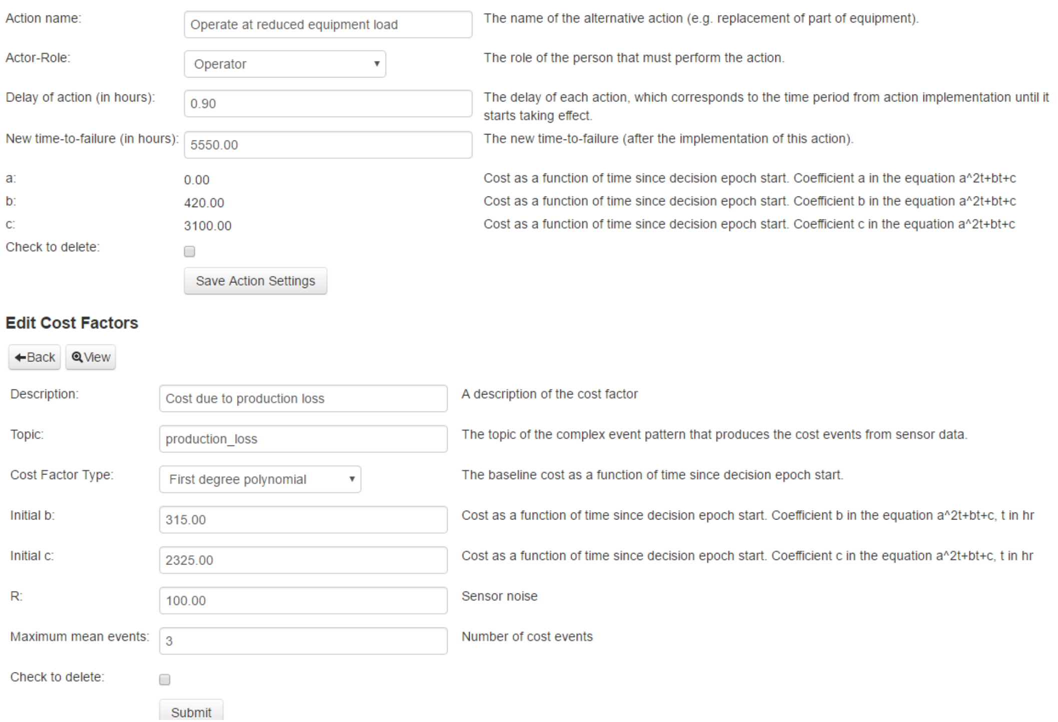

Figure 8-17: Edit Cost Factors.

The fourth component of the instance management screen is the 'Decision Method Settings' component in which, the users can specify method parameters before using the instance. After having editing all the alternative actions that correspond to the specific decision method instance, the users should click on the "Edit" button of the "Decision Method Settings" of the "Instance Management" screen. In this way, they are navigated to the "Instance Settings" screen, where they can modify the instance name and the instance expiration time, that have been edited at the beginning of the decision method instance configuration. Apart from this, they insert the cost of the undesired event that should be mitigated, the start date and time of the decision epoch and the duration of each decision epoch (e.g. time interval between two successive planned maintenances, shifts, etc.), as shown in Figure 8-18. After completing the associated parameters, click on the "Update decision method parameters" button so that the changes are applied.

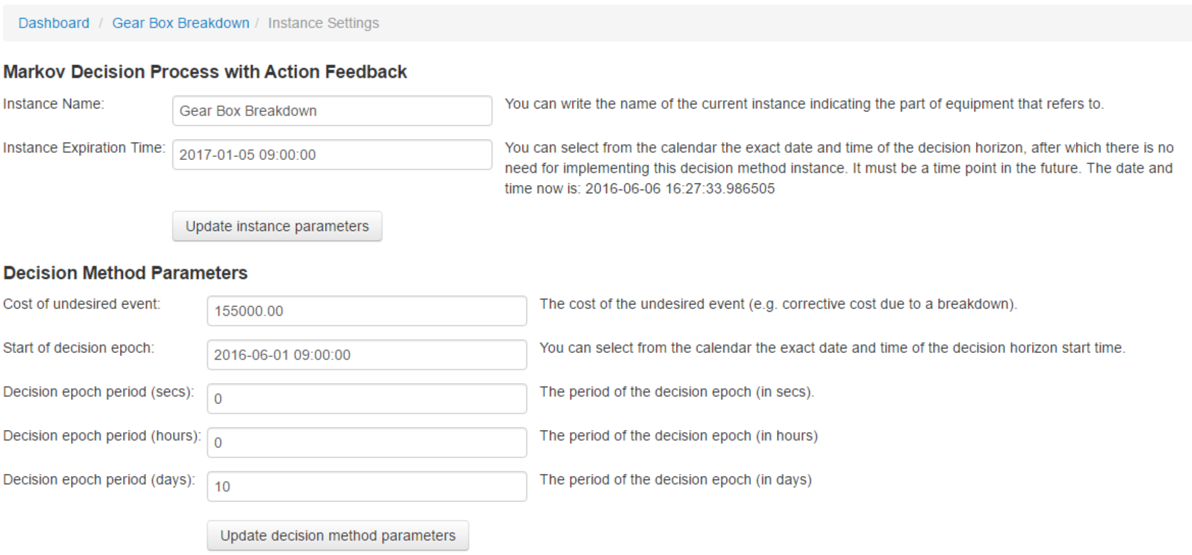

Figure 8-18: The "Instance Settings" screen.

After having completed the embodiment of domain knowledge in the PANDDA system, the specific instance is ready for use, as shown in Figure 8-19, and the 'My Instances' screen will be as shown in Figure 8-20. If any changes are required, these can be done by pressing the 'Manage' button. In this screen, there is a search area, facilitating the users to locate a specific instance in case their number is very large, as well as some filters allowing the presentation of Active, Expired or Inactive instances only.

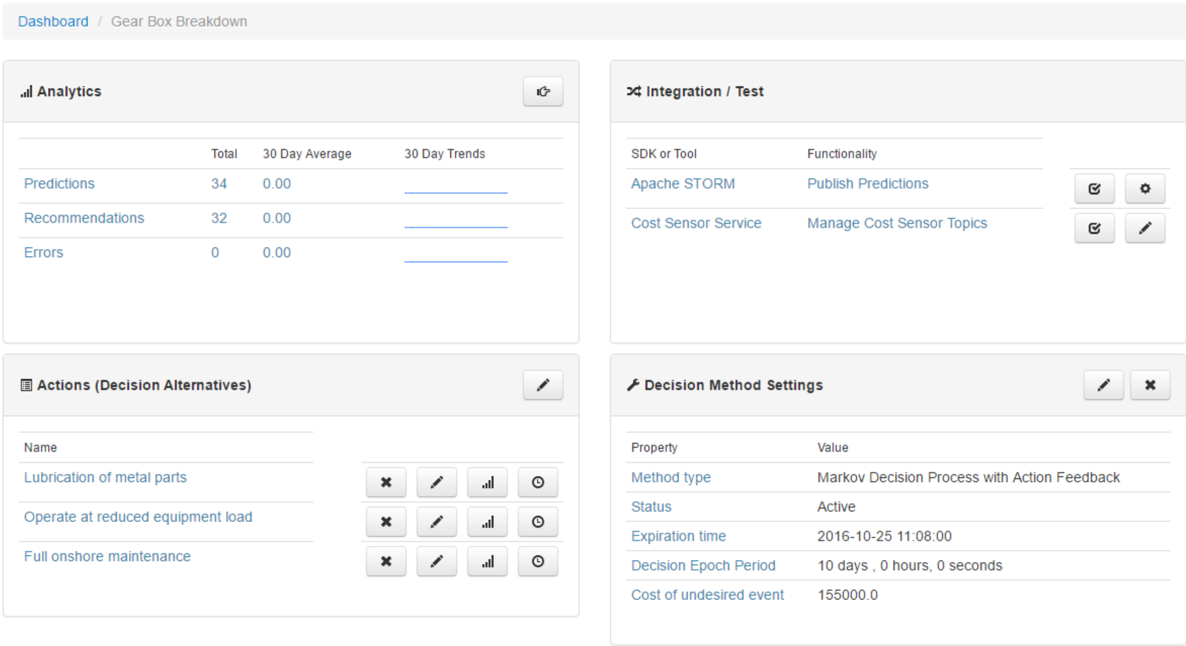

Figure 8-19: The updated instance management screen.

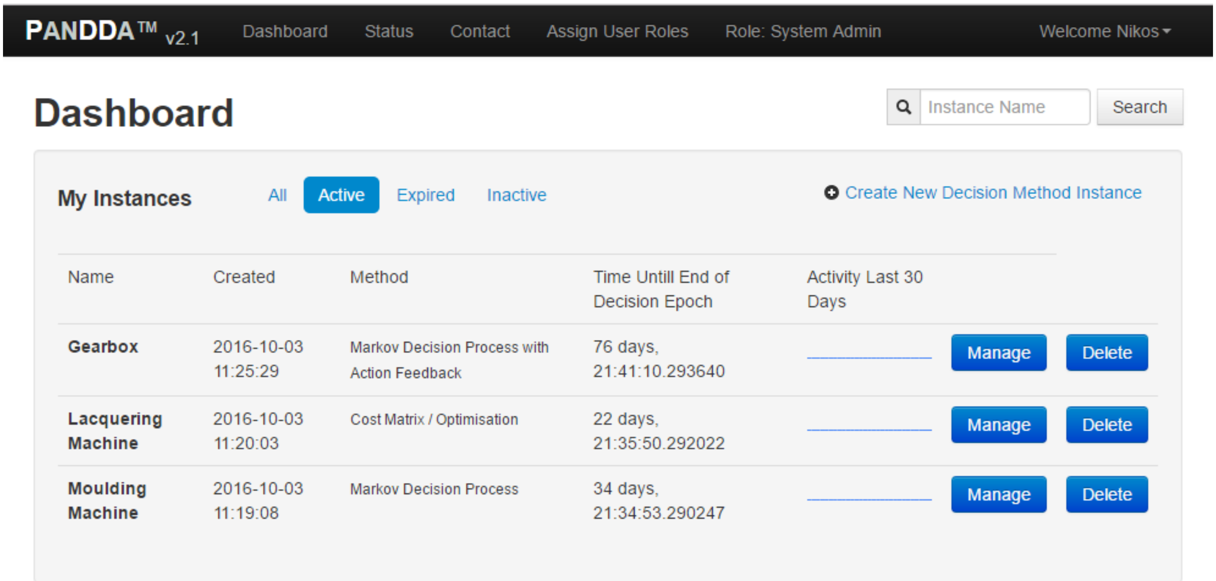

Figure 8-20: 'My Instances' screen for three instances

8.2.3 Providing a recommendation

When a prediction event triggers PANDDA, a recommendation is provided. The recommendation can be seen if the users click on the "Test" button of the Integration / Test component of the instance management screen that is shown in Figure 8-21. Then, they are navigated to the screen of Figure 8-22 where they are able to view a list about the predictions that generated recommendations. By clicking on the "View" button next to each element of this list, they can view the details for each provided recommendation. In the same screen, they are also able to simulate various prediction events.

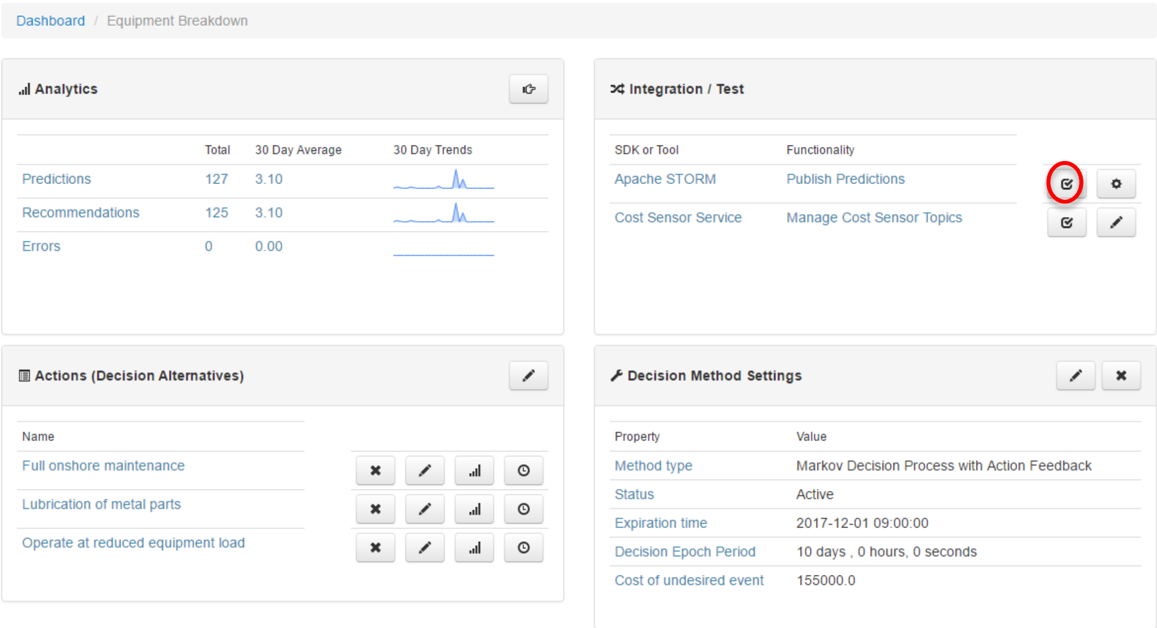

Figure 8-21: The "Test" button of the Integration / Test component of the instance management screen

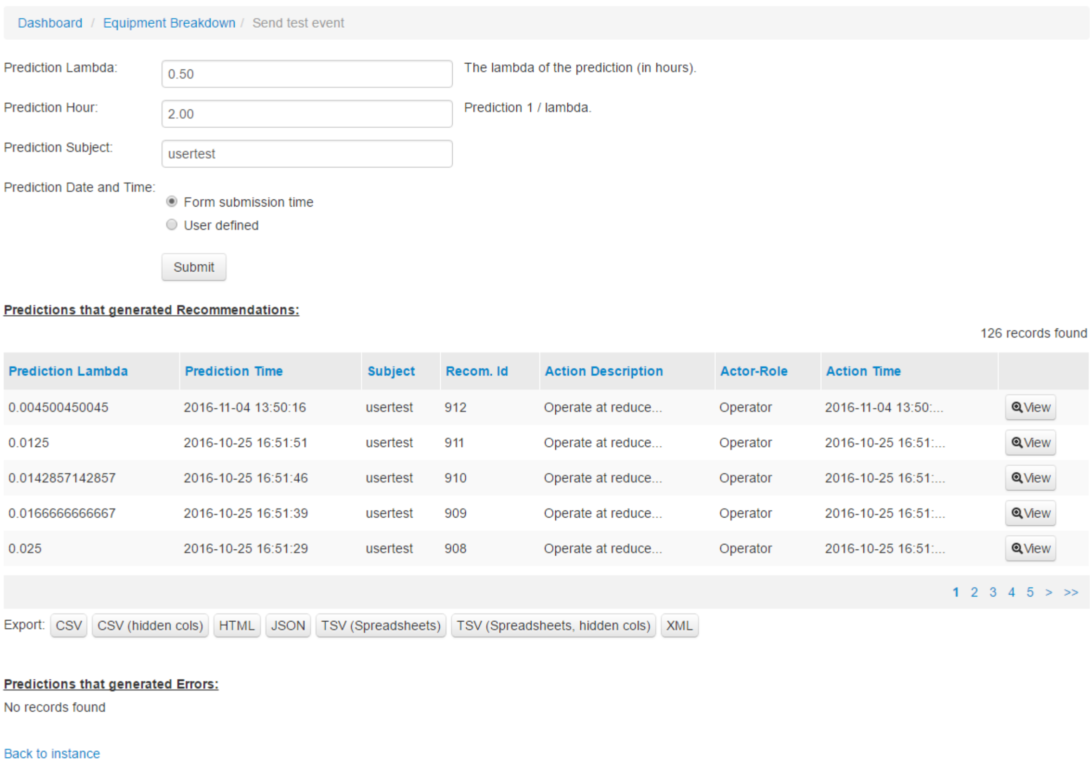

Figure 8-22: The "predictions that generated recommendations" list

8.2.4 *Sensor-enabled online cost monitoring*

At any time, the users can monitor the cost of an action in real time based on sensor measurements, by clicking on the "View Real-time Action Cost" next to the name of the action, at the "Actions (Decision Alternatives)" component of the "Instance Management" screen. In this way, they are led to the screen of Figure 8-23. They can view the baseline cost, i.e. the cost when the action has not been applied yet, or view the cost of the action during its implementation. The cost of the action is showed for all the decision epochs involving the implementation of this specific action. More specifically, PANDDA shows the actual cost based on the raw sensor measurements for each cost factor which is part of the action cost function. Both the measured (noisy) and the estimated (corrected) costs are presented since PANDDA filters out sensor noise.

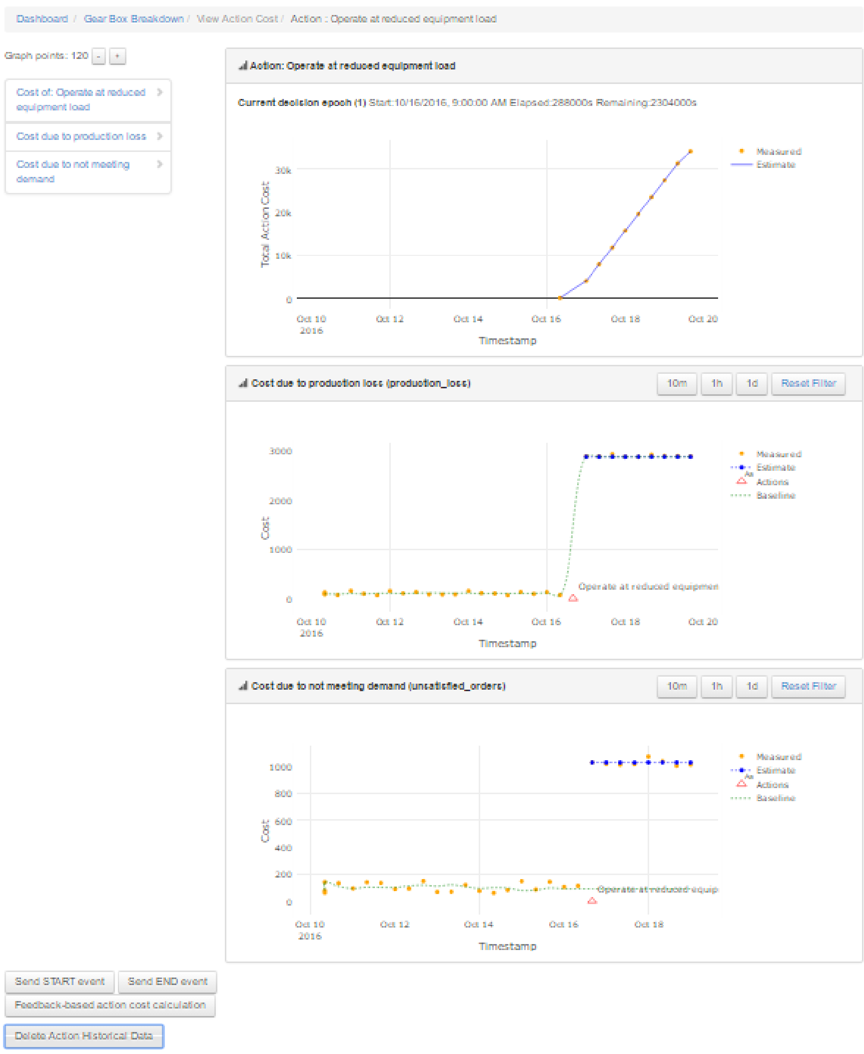

Figure 8-23: The screen for "View Real-time Action Cost"

At the same screen, PANDDA shows the total cumulative action cost for each decision epoch based on the sensor measurements of the various cost factors. For example, for monitoring the cost of the action “Operate at reduced equipment load”, PANDDA shows the raw cost measurements and estimations of two cost factors (cost due to production loss, cost due to not meeting demand) that are mapped to two topics – sensors (production_loss, unsatisfied_orders), as shown in Figure 8-24 and in Figure 8-25 respectively. The cumulative total action cost for these two cost factors is shown in Figure 8-26. In this example, the action has already been implemented twice (in two decision epochs) and it is currently being implemented for the third time.

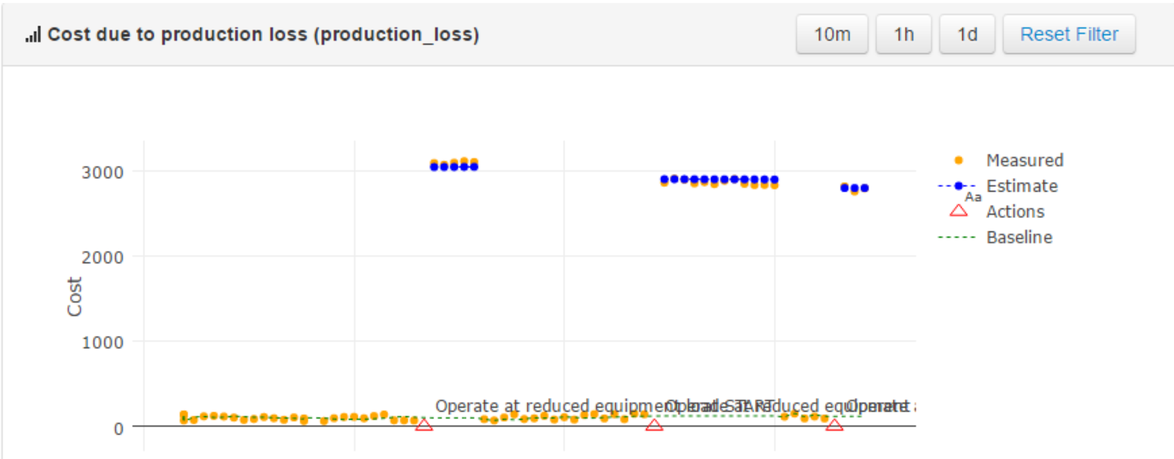

Figure 8-24: Real-time monitoring of the cost factor “Cost due to production loss”.

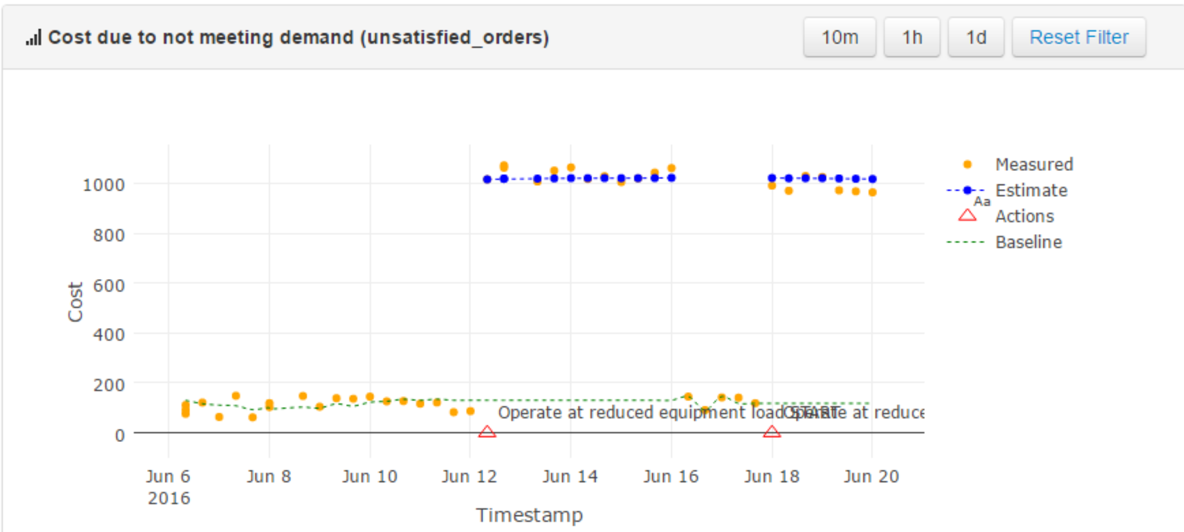

Figure 8-25: Real-time monitoring of the cost factor “Cost due to not meeting demand”.

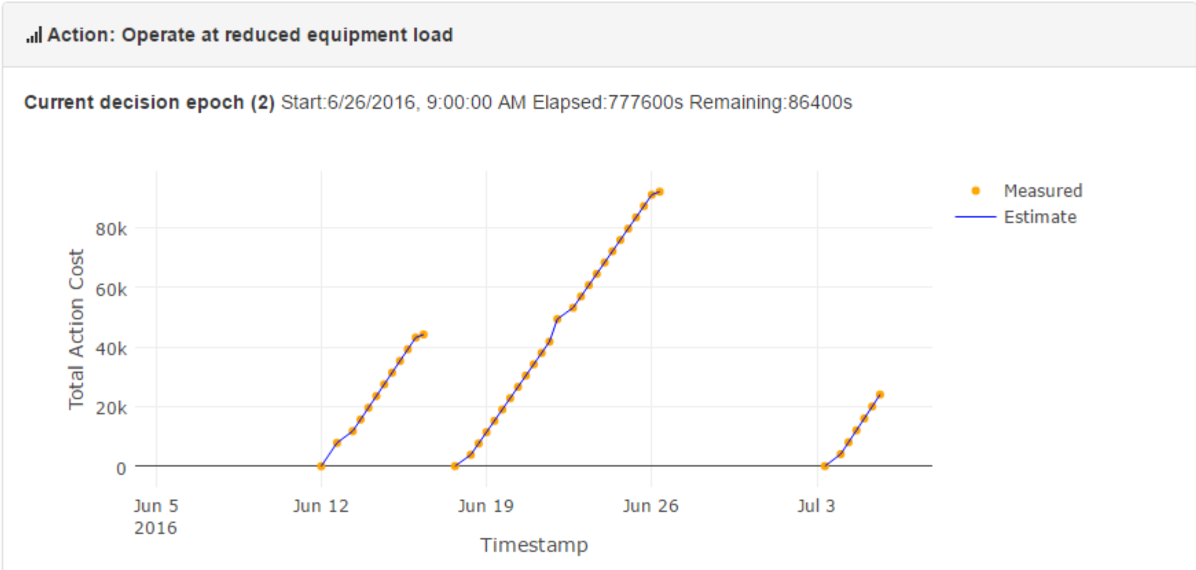

Figure 8-26:Real-time monitoring of the cumulative total action cost.

8.2.5 Sensor-enabled cost function update

At any time, the users can click on the "Feedback-based action cost calculation" button below the diagrams to be navigated to the "Calculate Action Cost from Feedback" screen of the specific action. The "Calculate Action Cost from Feedback" screen provides information about the action cost function and gives the possibility to calculate the refined cost function based on SEF. The first diagram of this screen is a 3-D diagram that presents the relationships among action cost, remaining time (until the end of decision epoch) and action start time, as shown in Figure 8-27. By moving the cursor on the points of the diagram, the user can see the values of the three axes.

At the same screen of PANDDA, below the diagram of Figure 8-27, there is the "Action Cost vs Remaining Time" diagram, as shown in Figure 8-28. In this diagram, the user is able to see the total cumulative action cost for each decision epoch in which the specific action has been implemented. By moving the cursor on the points of the diagrams, the user can also see the exact cumulative total cost value for a specific decision epoch. For example, in the case shown in Figure 8-28, there are cost data for 6 decision epochs and in the 4th decision epoch that this action was implemented, the prediction event was received approximately 160 hours before the end of decision epoch and costed 73873.84 euros. All the cost data can also be seen at the table existing below the "Action Cost vs Remaining Time" diagram and can be exported as a file as shown in Figure 8-29. In addition, there is a "Calculate Cost Function" button to see the estimated action cost function, as derived based on SEF.

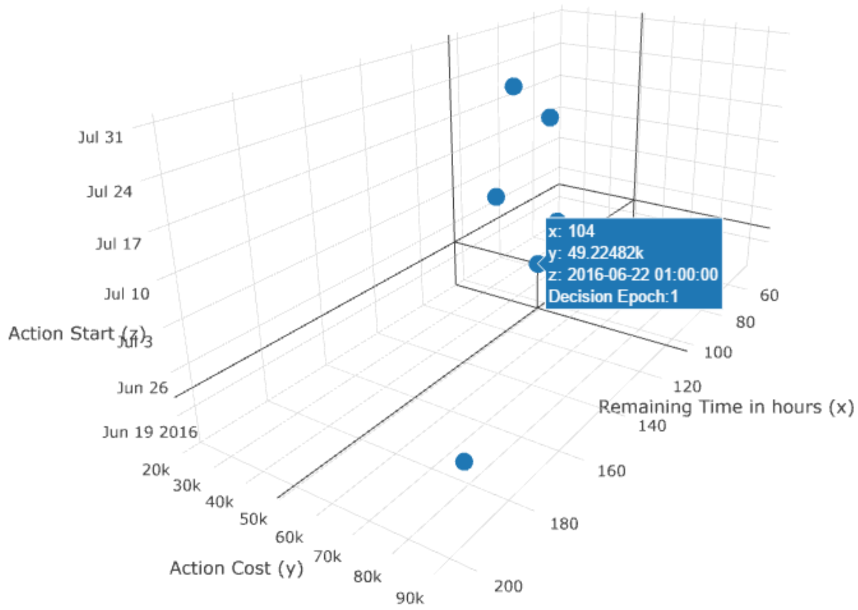

Figure 8-27: The "Action Cost vs Remaining Time vs Action Start Time" diagram.

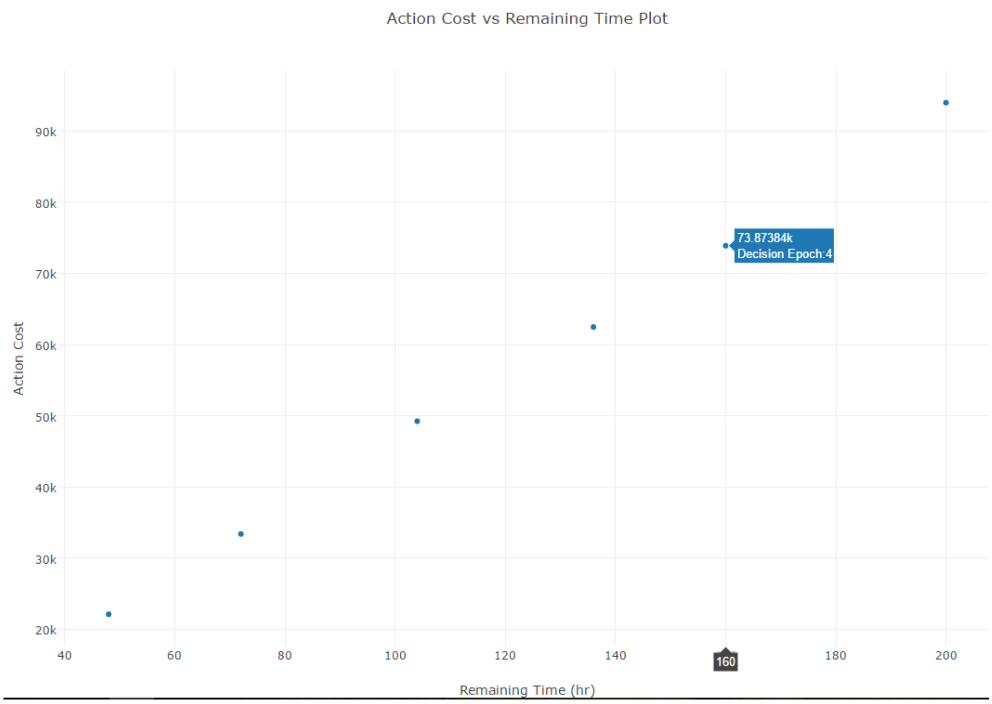

Figure 8-28: The "Action Cost vs Remaining Time" diagram.

Calculate Cost Function

Data: Search Clear 6 records found

Decision epoch number	Remaining Time (hr)	Cost	Estimated Cost	
5	136	62223.7513847	62459.3188185	Edit Delete
4	160	72982.1358653	73873.843276	Edit Delete
3	72	33128.6845839	33381.7403099	Edit Delete
2	200	91443.6405364	93980.7054945	Edit Delete
1	104	47540.4657826	49224.8217355	Edit Delete
0	48	22216.2432212	22103.3722485	Edit Delete

Export: CSV CSV (hidden cols) HTML JSON TSV (Spreadsheets) TSV (Spreadsheets, hidden cols) XML

Figure 8-29: The table presenting the action cost data for all the decision epochs.

By clicking on the “Calculate Cost Function” button, the “Action Cost vs Remaining Time” diagram is updated in order to show the estimated cost function based on SEF in comparison to the previous cost function and to the observed data, as shown in Figure 8-30. Below that diagram, two cost functions appear: the refined and the previous one. Then, the users are able to select whether you prefer the previous or the refined cost function to be taken into consideration by the decision method instance the next time it is going to be enacted online, by clicking on the relevant button. For example, if they select to use the refined function, a confirmation message appears, as shown in Figure 8-31.

If you do so, you will notice that the “Edit Instance Action” screen has also been updated and the coefficients of the action cost function have been changed according to the refined cost function derived from SEF. You can select to use the initial cost function at any time by clicking on the “Set action cost from cost factor, initial a, b, c” button. If you do so, the action cost function will be the aggregation of the cost factor functions inserted during configuration. You can always use the last refined cost function derived from SEF by clicking on the “Feedback-based action cost calculation” button. The aforementioned functionalities of PANDDA are shown in Figure 8-32.

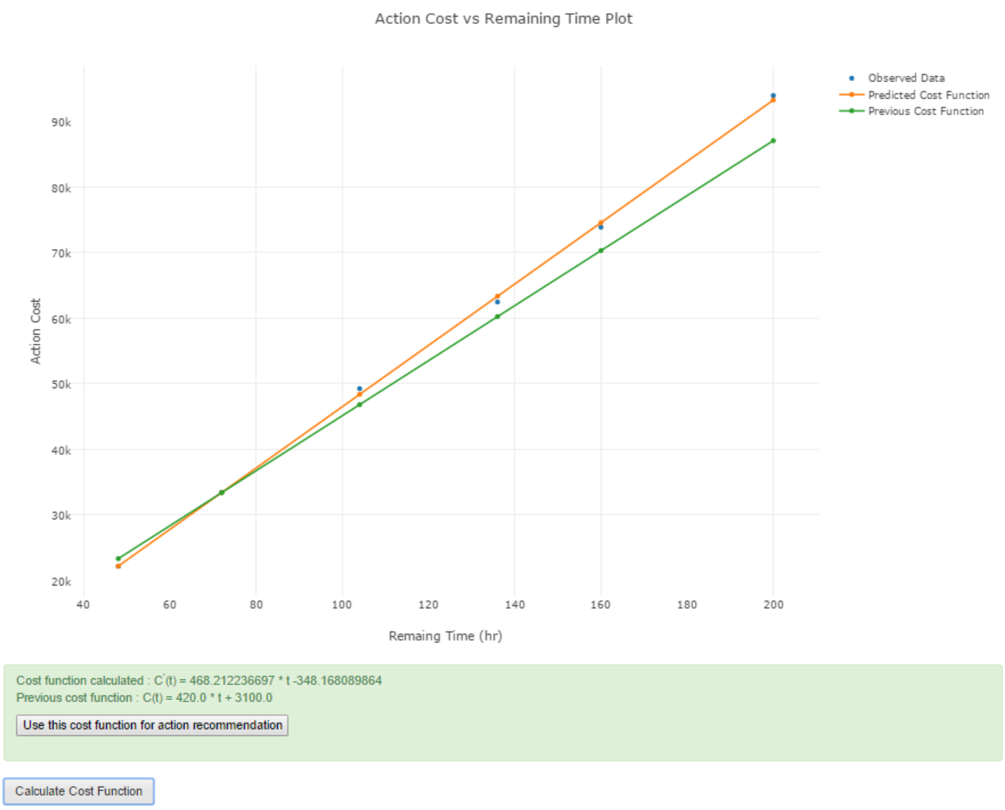

Figure 8-30: The updated "Action Cost vs Remaining Time" diagram showing the previous and the refined cost function.

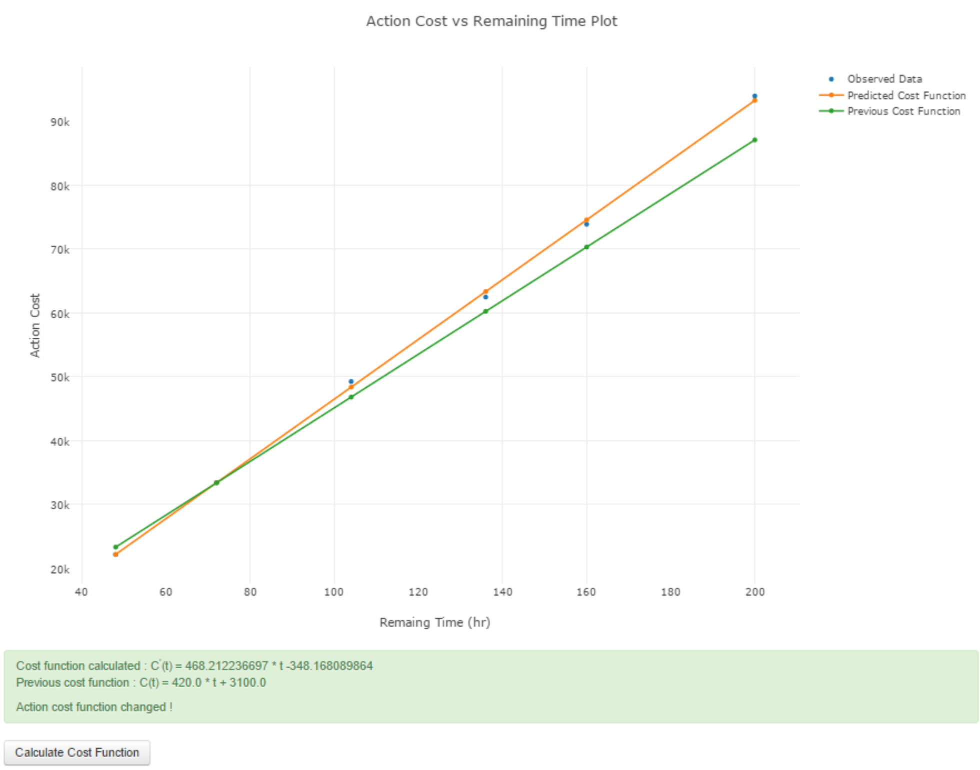

Figure 8-31: The action cost function has changed and the refined one is used.

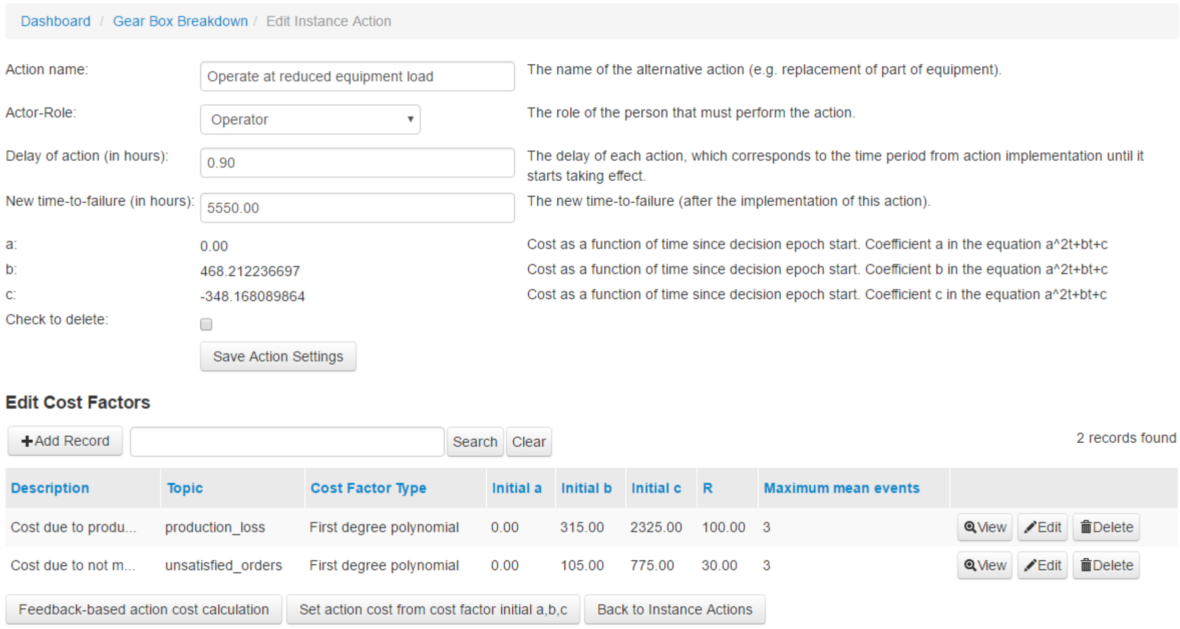

Figure 8-32: The "Edit Instance Action" screen after the selection of the refined action cost function.

8.2.6 Testing Decision Method Instances

The Analytics and the Integration / Test components of the instance management screen are addressed to the **System Administrator**. The Analytics component presents trends of the last 30 days for the predictions received, the recommendations provided, the errors and the latency. If the users want to see more details (e.g. about the results of predictions, the actions recommended, etc.) and diagrams, they can enter the Analytics space by pressing the relevant 'View' button on the screen.

The Integration / Test component of the instance management screen has a double role. First, the System Administrator integrates the specific decision method instance with the sensors that correspond to the cost factors of the action cost functions. Second, the System Administrator can test the decision method instance and see what the generated recommendation would be for various prediction events. In this way, they can test different scenarios by simulating different prediction events at the "Send Test Events" screen. The user navigates to this screen if he / she clicks on the "Test" button, next to "Apache STORM", which is found in the Integration / Test" component of the "Instance Management" screen. The results of all the conducted simulations are accessible at any time. Figure 8-33 shows an example where the user simulates a prediction event that the most probable value of the time-to-undesired event (taking into account that the probability distribution function of the occurrence of the undesired event is exponential) is in 222.2 hours, which corresponds to a λ value of 0.0045. This prediction event, in combination with all the other parameters of the decision method instance given by the user during configuration, leads to a recommendation, as shown in Figure 8-34.

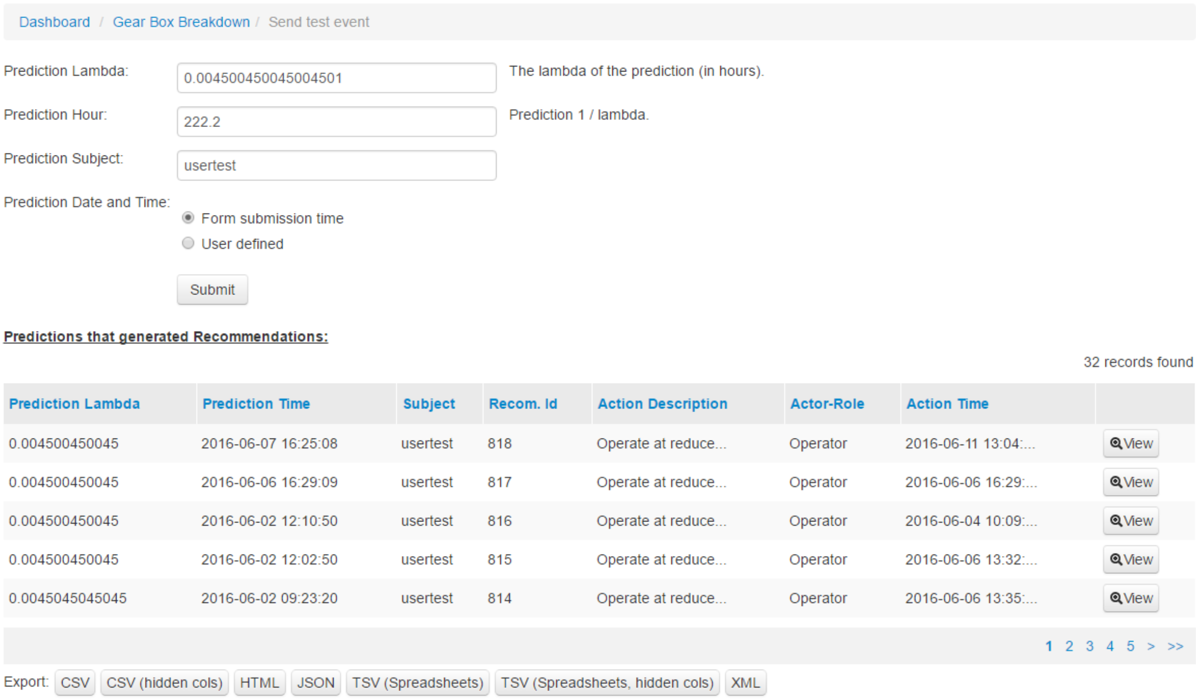

Figure 8-33: The "Send Test Even" screen.

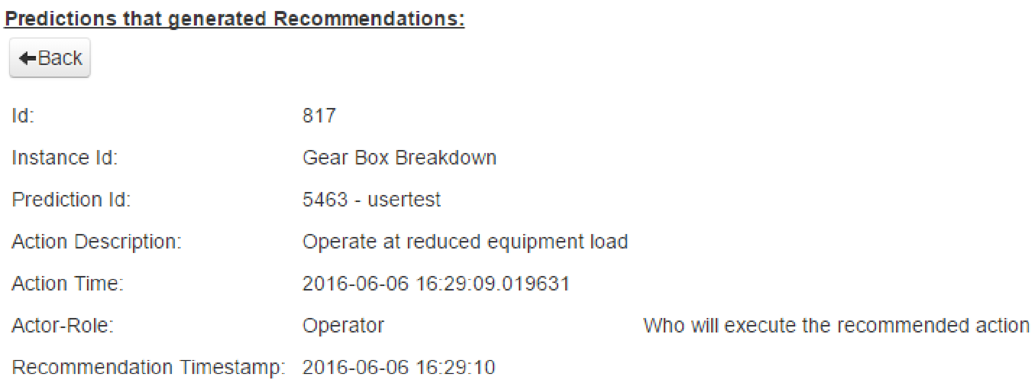

Figure 8-34: The resulting recommendation for a specific prediction event.

9 Deployment in Industrial Environment

This Chapter presents the deployment of PANDDA as part of the overall Proactive Maintenance information system in a real industrial environment. More specifically, it presents the functionalities of PANDDA in the context of real industrial scenarios in two pilot companies: MHWirth from oil and gas industry and HELLA from automotive lighting equipment industry. Based on the identified need of the companies to turn from reactive into proactive by adopting new technologies and systems in order to facilitate Proactive Maintenance implementation, the platform was deployed in their premises.

9.1 The MHWirth Business Case

9.1.1 Description of the MHWirth Business Case

Oil and gas projects are capital-intensive investments, with severe consequences in financial and environmental terms in case of breakdown (Telford et al., 2011). Since a typical production rate for an oil and gas corresponds to USD 500,000, the reduction of downtime is of great significance in the oil and gas industry taking into account that one hour of downtime costs around USD 20,000 (Telford et al., 2011). So, the business added value of proactive maintenance event-driven decision making in oil and gas industry is huge since cost, efficiency and safety are crucial aspects in this kind of industry (Payne, 2010). Although comparable industries such as automotive and aviation have recently started exploiting big data by analyzing them and processing them in suitable information systems, the oil drilling industry has not reached to that level yet.

MHWirth provides oilfield products, systems and services for customers in the oil and gas industry world-wide. The company's knowledge and technologies span from reservoir to drilling, production and through the life of a field. It brings together engineering and technologies for oil and gas drilling, field development and production. The company employs approximately 4,300 people in more than 20 countries. They apply the knowledge and create and use technologies that deliver their customers' solutions. The annual revenue is approximately EUR 1.1 billion (2013). The company is listed on the Oslo Stock Exchange. The company offers complete drilling packages, single drilling equipment and lifecycle services comprising installation and commissioning, maintenance and periodic overhaul of the installed base of machines around the world. Among the main customers are oil companies, rig owners and construction yards. Geographically, the main markets include the North Sea, Brazil and Asia, and project, sales and service organisations are located close to all main markets and customers. The main office is located in Kristiansand, Norway, and most of the drilling equipment is produced and assembled there. Figure 9-1 shows an oil rig owned by the pilot use case under consideration.

Figure 9-1: An MHWirth oil rig.

The Exploration and Production (E&P) cost within the oil and gas industry has had a considerable increase the latest 15 years, which is a major concern of the industry by limiting the number of oil and gas fields that can be exploited economically. All major oil companies are striving to kick off initiatives to reduce the costs related to the drilling process. It is a paradox that alongside an increasing level of automation on-board latest generation drilling rigs, the drilling efficiency is reduced. The push towards greater ocean depths, harsher environments and more advanced wells to be drilled has resulted in lower drilling efficiency. The average cost of drilling a well at the Norwegian continental shelf (NCS) exceeds EUR 58 millions; a number Statoil aims to reduce with 15-20%.

Cost focus and efficiency are the important topics for the drilling industry at the moment. To improve the understanding of the market situation within the sector, a high level review of the most usual process of ordering, building and operating a drilling rig might be appropriate. The process is initiated by a *rig owner company*, which puts forward a set of requirements for a new build to *a construction yard*. The yard is subsequently asking for quotes from *sub suppliers* for the drilling equipment package or single equipment. The main criterion for the yard when selecting sub suppliers is the equipment price, assuming that the prevailing customer requirements, standards and regulations are fulfilled. This lifecycle is shown in Figure 9-2.

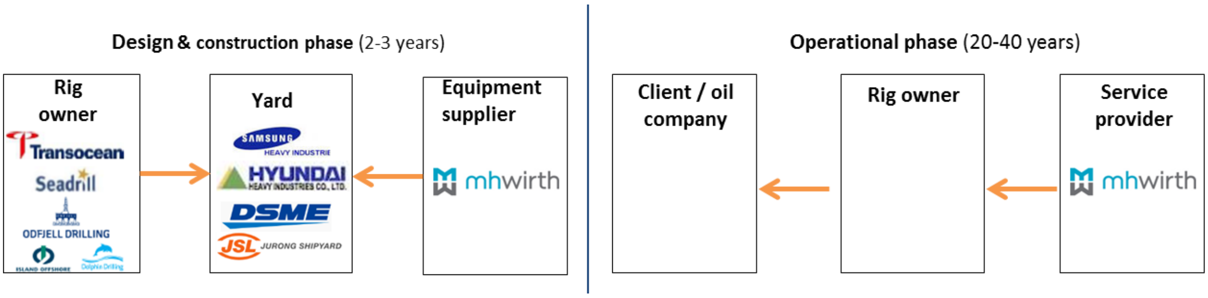

Figure 9-2: Typical drilling rig lifecycle with example stakeholders per phase.

The oil drilling company under consideration is a partner in both the design and construction phase and through the rig's lifecycle. The company aims to improve service offering to increase involvement in the important operational phase, which is considerable longer, has higher margins and provides more steady cash flow due to being less affected by cyclic economic conditions than the construction phase. To succeed in such a transition, the strategy has to be adopted already during design of equipment to improve data collection possibilities. Currently, the utilization of condition monitoring technology is hindered by the high cost of implementation during operation. Proactive customers looking into new contract regimes might speed up this process significantly. Requirements for machine availability in operation provide incentives for equipment suppliers for the transition towards Proactive Maintenance. In other words, the aim is a transition from reactive business and corrective maintenance strategy towards proactivity and higher predictability in operation and maintenance needs.

Riglogger™ is of particular importance as input source for data to the ProaSense system infrastructure. The Riglogger™ system is an infrastructure developed by the company for high speed, high capacity logging of operational data from the company's installed base of drilling equipment. Its topology is depicted in Figure 9-3.The Riglogger™ system continuously records up to 25,000 measuring points and captures events at a frequency of up to 50 Hz, ensuring a high data density (OSIsoft). It consists of two main components, one ppropri-etary solution for high capacity streaming of time series out of the equipment Programmable Logic Controllers (PLC) including decoding capabilities of the data streams at the receiving side. Secondly is an adapted version of the OSIsoft PI historian handling data reduction and storage. The OSIsoft suite contains also a variety of available interfaces for communication (e.g. towards third party systems), satellite replication of data and tools for data structuring and visualization.

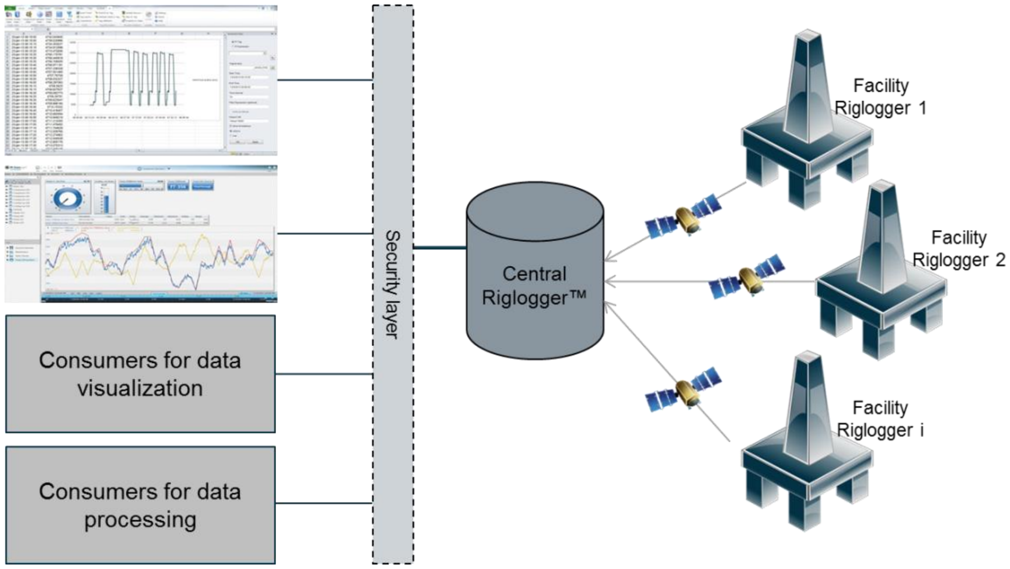

Figure 9-3: Riglogger high level topology and data flow.

The potential of ProaSense towards minimizing the weaknesses and avoiding the threats of the oil drilling company is explained through the SWOT analysis of Table 9-1. By analysing both the strength and weaknesses with respect to opportunities and threats, we get the elements of a strategy presented in Table 9-2. Table 9-1 and Table 9-2 were created in conjunction with the pilot company under examination.

Based on extensive root cause analysis using Bow Tie diagrams for various parts of equipment, the most critical ones were selected. Therefore, two use cases were selected: the gearbox and the swivel monitoring case. The gearbox and the swivel are critical components of the top drive, with severe consequences in terms of down time in case of damage.

It is therefore of interest to monitor these components in order to give early notifications if degradation or breakdown is expected in near future. As of today, the maintenance regime is inspection every 5 years. If a damage or unexpected wear should occur between the 5 year inspections it would probably not be detected, and a major failure would be the consequence. Figure 9-4 shows the motor and gearbox details.

Table 9-1: SWOT Analysis for MHWirth.

Strengths	**Weaknesses**
The company is a producer of equipment and systems with a large installed base, and a lifecycle service organisation in place. Being small in size, with short lines of command, and top management based in our own country, they can quickly introduce differentiating technology to the marketplace. Both the ProaSense project and the Riglogger™ are good examples of their ability to introduce modern IT technology to a conservative business. The Riglogger™ infrastructure is an enabler for offering next generation monitoring, analysis and decision support services. There is a defined will within their management to spend a higher fraction of our R&D budget on longer term, differentiating technologies. The workforce is highly skilled, capable of producing good quality, reliable designs and also with visionary abilities to make new, creative solutions that can be quickly introduced to the market. The reputation in the market is also good. We have over time established a base of loyal customers.	Modern IT concepts are not easy for the established sales force to sell to existing customers, who basically want one more unit similar to their existing installations. Understanding and utilizing the newly introduced concepts will require internal education of mid and top level management as well as the sales force. Because of the cost focus when bidding for new projects, it is difficult to introduce new and expanded instrumentation necessary for improved monitoring. Patenting and IP protection is historically a weakness within our organization. Being small, it is easy to destroy a good reputation with a very few failures or badly performing new constructions, and thereafter it would take a long time to re-build the good reputation. They have different customers during construction and the life cycle of the rig. At construction time the focus is on keeping equipment cost at a minimum. Organisational changes require new working processes. We are a project organisation trying to become a product organisation, with standardized deliveries, this requires changes. All new products, including Riglogger™ and the notifications based on ProaSense require engineers to think about Return on Investment (ROI) and create a valid business model.
Opportunities	**Threats**
The new emerging contract regime that calls for delivering equipment with guaranteed uptime, or even new business models such as leasing out equipment with free replacement in case of breakdown is a good opportunity to gain new contracts in the marketplace. Being able to offer modern differentiating technology that enables their customers to offer a better service to their end customers is a part of this new strategy. There is now a market pull for introducing improved condition monitoring (CM) and PHM of our equipment, coming from our end customers.	The new contract regime with guaranteed uptime means an increased risk if the failure rate of the equipment should rise above what is expected, or condition monitoring should fail to give correct early warnings of malfunction and wear. Even with their loyal customers, price is rapidly becoming more of an issue. As with most businesses, there are cycles in the market, and the recent sales boom will not last forever. Ownership of data is not clear. Other parties may harm their reputation by incorrectly analysing the data lacking our domain knowledge.

Table 9-2: Strategy derived from SWOT Analysis for MHWirth.

	Opportunities	Threats
Strengths	They aim to move forward based on their competency, good reputation and already established technological enablers, making use of the expressed will in top management to increase the percentage of R&D funding used for real differentiating technology that will enable their customers to perform better.	They should use their agility (their comparatively small size) to deploy low cost solutions (low-hanging fruit) and gain quick wins in CM/PHM. This strategy means to go for rapid deployment of low cost solutions based on making better use of already installed sensors and infrastructure.
Weaknesses	They should use the recent market pull to motivate the introduction of both simple and advanced monitoring and prognostics methods for enhancing equipment reliability. They should also strengthen their business mind set.	They should try to strengthen their knowledge level with respect to IP protection and the use of IT in meeting the new contract requirements and minimizing risks, as well as securing management support for introducing more advanced monitoring techniques.

Figure 9-4: Motor and Gearbox details.

9.1.2 Deployment in MHWirth

9.1.2.1 Use Case 1: Proactive Recommendation of Maintenance Action for the Gearbox

Use Case 1 deals with the "Gearbox Breakdown" DMI and aims to provide proactive recommendations about maintenance actions. First, the user configures a DMI through the PANDDA GUI. The desired recommendation is about the optimal maintenance action and the optimal time for its implementation. Therefore, the proactive MDP method is selected and the relevant DMI for the gearbox breakdown mitigation is created through the PANDDA GUI. The input parameters inserted by the user are shown in Table 9-3 and Table 9-4.

The Real-time Processing Layer of the platform receives in real-time readings of the oil temperature and Rounds Per Minute (RPM) of the drilling machine's gearbox. Assume that abnormal friction losses are detected on the basis of the observed data through complex patterns of oil temperature and RPM events characterized by an abnormal oil temperature rise (10% above normal) measured over 30% of the drilling period when drilling RPM exceeds a threshold. This pattern is a strong indication that the gearbox breakdown of the drilling equipment starts to occur.

Table 9-3: Input by the user of actions parameters for gearbox breakdown instance

Alternative actions	**Cost functions** *(in hours)*	**Delays** *(in hours)*	**Time-to-failure after implementation** *(in hours)*
a1: Lubrication of metal parts	400*(T-t)	0.9	2400
a2: Operate at re-duced equipment load	420*(T-t)	0.9	5550
a3: Full onshore maintenance	550*(T-t)	0.9	10000

Table 9-4: Input by the user of gearbox breakdown instance

Cost of undesired event	155,000 euros
End of decision epoch	10 days

Table 9-5: Real-time input and output of gearbox breakdown instance

Predicted probability distribution	Exponential with λ = 0.0045
Recommended action	a2: Operate at reduced equipment load
Recommended implementation time	in 124 hours

The detection event triggers the prognostic model that generates a real-time prediction. Three hours after the start of decision epoch for the “gearbox breakdown” DMI, PANDDA is triggered by a prediction event that there is an exponential probability distribution function for the occurrence of gearbox breakdown with a most probable time-to-failure equal to 222.2 hours, i.e. $\lambda = 0.0045$. The “gearbox breakdown” DMI is enacted online and provides a recommendation that the optimal action is “Operate at reduced equipment load” and the corresponding optimal time for implementing this action is in 124 hours. This real-time input and output of PANDDA is shown in Table 9-5. This action at this time is the one that leads to the minimum expected loss, as shown in Figure 9-5.

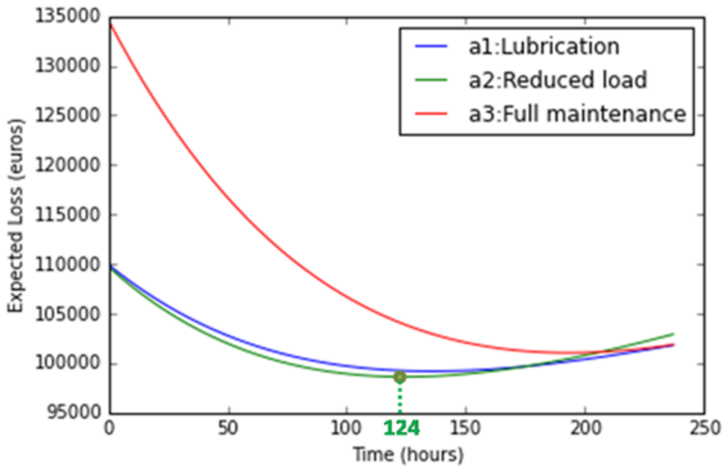

Figure 9-5: Output of proactive MDP for the oil and gas scenario

9.1.2.2 Use Case 2: Proactive Recommendation of Maintenance Action for the Swivel

Another DMI corresponds to the swivel of the oil and gas drilling equipment. The aim is to move dynamically swivel replacement according to the swivel health state. Therefore, the desired output of PANDDA is the optimal time for the implementation of a predefined action. At design time, the user creates an appropriate DMI by selecting the Proactive Expected Loss Rate optimization method and by inserting the required domain knowledge shown in Table 9-6.

Table 9-6: Input by the user of swivel breakdown instance

Cost of undesired event	100,000 euros
Cost of swivel replacement	800 * (T – t)
End of decision epoch	280 hours

Similarly to the previous scenario, 126.15 hours after the start of decision epoch for the "swivel breakdown" DMI, PANDDA is triggered by a prediction event that there is an exponential probability distribution function for the occurrence of swivel breakdown with a most probable time-to-failure equal to 153.84 hours, i.e. λ = 0.0065. The "swivel breakdown" decision method instance is enacted online and provides a recommendation that the optimal time for swivel replacement is in 73.65 hours. This real-time input and output of PANDDA is shown in Table 9-7. At this time, this action leads to the minimum expected loss, as shown in Figure 9-6.

Table 9-7: Real-time input and output of the decision method

Predicted probability distribution	Exponential with λ = 0.0065
Recommended action	Replace swivel
Recommended implementation time	in 73.65 hours

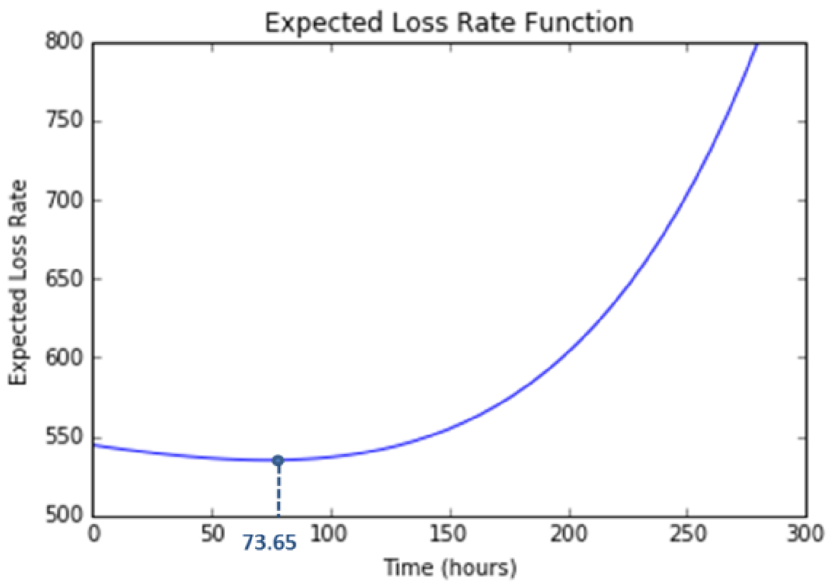

Figure 9-6: Output of Expected Loss Optimization method for the oil and gas scenario

9.1.2.3 Use Case 3: Proactive Recommendations of Joint Maintenance and Logistics Actions for the Gearbox

For the machine's gearbox DMI, friction losses are detected with the use of complex event patterns of lube oil temperature and RPM events characterized by an abnormal oil temperature rise measured over a percentage of the drilling period when drilling RPM exceeds a threshold. This pattern, learned at the offline phase, is an indication that the gearbox may be at a dangerous state. Therefore, the prognostic model generates a prediction about the reliability distribution function of the gearbox. This prediction triggers PANDDA which provides a proactive recommendation about the optimal maintenance action and the optimal time of applying it as well as the optimal order of spare parts along with the optimal time for their ordering.

At design time, the user interaction is realized with a GUI of the web-based application enabling the user to insert the required domain knowledge per equipment instance. In the current scenario, there are four alternative maintenance actions (lubrication of metal parts, operate at reduced equipment load, offshore maintenance, full onshore maintenance) with different degrees of restoration and their associated orders of spare parts (lube oil, no ordering, gearbox, Derrick Drilling Machine- DDM), as shown in Table 9-8. The time-to-failure after the implementation of the maintenance action indicates the degree of restoration. The actions a1, a2 and a3 are implemented on the oil rig (offshore), while onshore maintenance, which corresponds to perfect ("good-as-new") maintenance, requires its movement onshore.

Table 9-8: The domain knowledge inserted during user configuration.

<table>
<tr><td colspan="2">Cost of failure (Euro)</td><td>350,000</td></tr>
<tr><td colspan="2">Decision horizon (hours)</td><td>240</td></tr>
<tr><td colspan="3">Maintenance actions</td></tr>
<tr><td rowspan="4">Time-to-failure after implementation (hours)</td><td>a1: Lubrication of metal parts</td><td>1,240</td></tr>
<tr><td>a2: Operate at reduced equipment load</td><td>2,050</td></tr>
<tr><td>a3: Offshore maintenance</td><td>2,960</td></tr>
<tr><td>a4: Onshore Maintenance</td><td>3,220</td></tr>
<tr><td colspan="3">Spare parts orders</td></tr>
<tr><td rowspan="4">Lead time (hours)</td><td>o1: Lube oil</td><td>5</td></tr>
<tr><td>o2: Swivel hook</td><td>8</td></tr>
<tr><td>o3: Gearbox</td><td>24</td></tr>
<tr><td>o4: DDM</td><td>48</td></tr>
</table>

At some time, a prediction event about an exponential distribution function of the failure occurrence with a parameter λ = 0.045 triggers the decision algorithm. The joint maintenance and logistics proactive decision model are formulated for each alternative action as shown below:

$$EL^{a_i}(t) = \left[1 - \left(1 - e^{-\lambda t}\right)\right] * \left\{C_{a_i} + \left(1 - e^{-\lambda \Delta t}\right) * C_f + \left[1 - \left(1 - e^{-\lambda \Delta t}\right)\right] * \left[C_{e_i}(T - t - \Delta t) + \left(e^{(t+\Delta t)(\lambda - \lambda')} - e^{-\lambda' T + \lambda(t+\Delta t)}\right) * C_f\right]\right\} + \left(1 - e^{-\lambda t}\right) * C_f$$

$$EL^{o_i}(t) = \left[1 - \left(1 - e^{-\lambda(t+L)}\right)\right] * \left\{C_{sp} + \left(1 - e^{-\lambda \Delta t}\right) * C_s(T - t - L) + \left[1 - \left(1 - e^{-\lambda \Delta t}\right)\right] * \left(e^{(t+L+\Delta t)(\lambda - \lambda')} - e^{-\lambda' T + \lambda(t+L+\Delta t)}\right) * C_s(T - t - L - \Delta t)\right\} + \left(1 - e^{-\lambda(t+L)}\right) * C_s(T)$$

Although there is an indication of the most probable time-to-failure (parameter λ), the exponential degradation leads to high uncertainty in considering the deterministic value itself. Handling the PDF instead can lead to more accurate and reliable results. The Expected Loss Functions are shown in Figure 9-7 and their optimization results in the recommendation: Conduct offshore maintenance for gearbox replacement in 85.47 hours and order the Gearbox in 42.36 hours. These recommendations are exposed to the user through the GUI.

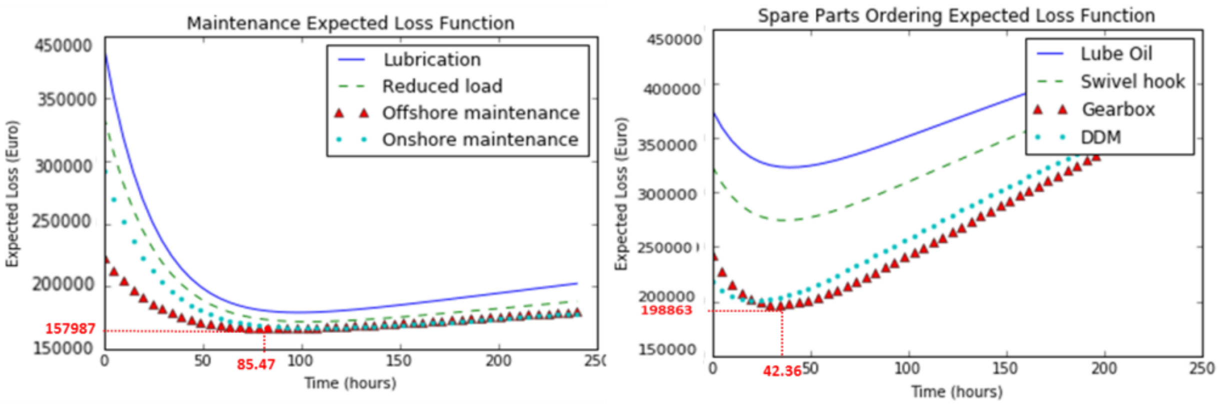

Figure 9-7: The expected loss functions for (a) maintenance, and (b) logistics (ordering of spare parts).

9.1.2.4 Use Case 4: Continuous Improvement of Proactive Recommendations about Gearbox

The user creates a DMI about the gearbox through the PANDDA GUI. When a prediction event triggers PANDDA, a proactive recommendation that minimizes the expected loss for the manufacturing enterprise is provided. The scenario demonstration is presented in three phases: (i) DMI configuration, (ii) proactive mitigation of gearbox breakdown and (iii) improvement of recommendations through SEF.

DMI Configuration

Senior engineers inserted the values of the input parameters shown in Table 9-9, Table 9-10 and Table 9-11. Time is expressed in hours. Table 9-9 shows three alternative actions in our application scenario. The action cost functions have been defined by the users to be linear, all of them have the same delay and each one of them has a different time-to-failure after its implementation (i.e. how much the action prolongs the gearbox lifetime). Each action consists of 2 cost factors, each one of which contributes to a different percentage to the total action cost as shown in Table 9-10. Table 9-11 shows additional parameters needed as input for the proactive decision methods.

Table 9-9: Input of Action Parameters

Alternative actions	Cost functions (Euro)	Delays (hours)	Time-to-failure after implementation (hours)
a1: Lubrication of metal parts	400*(T-t)	0.9	2400
a2: Operate at reduced equipment load	420*(T-t)+3100	0.9	5550
a3: Full onshore maintenance	550*(T-t)	0.9	10000

Table 9-10: Cost Factors for each Cost Function

Alternative actions	Cost Factors	
a1: Lubrication of metal parts	Cost of lube oil	Personnel cost
a2: Operate at reduced equipment load	Cost due to production loss	Cost due to not meeting demand
a3: Full onshore maintenance	Cost due to production loss	Cost of transport

Table 9-11: Input of the Decision Method

Parameter	Value
Cost of undesired event	155,000 euros
Start of decision epoch	01-06-2016
Decision epoch period	10 days
Instance expiration time	05-01-2017

Proactive mitigation of gearbox breakdown

Three hours after the start of decision epoch for the "gearbox breakdown" DMI, PANDDA is triggered by a prediction event that there is an exponential probability distribution function for the occurrence of gearbox breakdown with a most probable time-to-failure equal to 222.2 hours, i.e. $\lambda = 0.0045$, as shown in Table 9-12. The "gearbox breakdown" DMI (which incorporates the proactive MDP decision method) is enacted online and provides a recommendation that the optimal action/time pair is to "Operate at reduced equipment load" after 124 hours, as shown in Table 9-12. This action timing pair is the one that leads to the minimum expected loss, as shown in Figure 9-8.

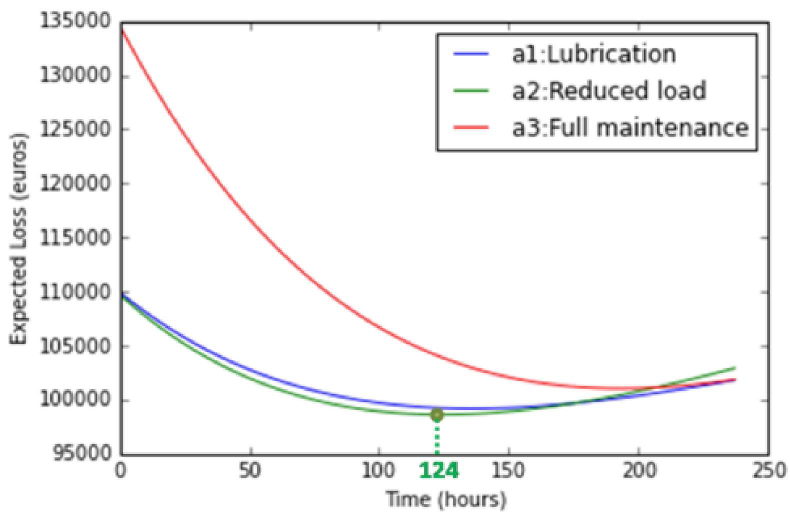

Figure 9-8: The expected loss for the three alternative actions.

Table 9-12: Real-time Input and Output of the Decision Method

Parameter	Value
Predicted probability distribution	Exponential with λ = 0.0045
Recommended action	a2: Operate at reduced equipment load
Recommended implementation time	in 124 hours

Improvement of recommendations through SEF

In the current scenario, feedback from 2 sensors related to the following cost factors (the aggregation of which formulates the cost function of the action "a2: Operate at reduced equipment load") is gathered:

- Cost due to production loss: Cost factor function = 300*(T-t) +2100
- Cost due to not meeting demand (penalty for unsatisfied orders): Cost factor function = 120*(T-t) +1000

"Cost due to production loss" cost factor is mapped to a flow monitoring sensor which measures the productivity i.e. the volume of oil gathering during oil drilling per time unit. This cost factor is calculated by multiplying productivity by the associated cost and adding the constant costs. In our example, production loss due to the action implementation is 600 gallons per hour with a cost of 0.5 Euro per gallon (300 Euro per hour), while there is a fixed cost of 2100 Euro which corresponds to the production loss of the time required for changing the production process. According to IEC 60770, flow sensors have an accuracy of 99.5 % in terms of Full Scale Output (FSO), or equally, an uncertainty of 0.5 % FSO. FSO is the resulting output signal or displayed reading produced when the maximum measurement for a given device is applied (Aberer et al., 2007). When an instrument has an accuracy specified as % FSO, the error has a constant value no matter where the flow is in the flow range (in contrast to the percentage of reading where the error is always the same percentage of the actual flow). FSO is the resulting output signal or displayed reading produced when the maximum measurement for a given device is applied (Aberer et al., 2007).

The "cost due to not meeting demand (penalty for unsatisfied orders)" cost factor represents the penalty per time unit because an order is not ready on time and is retrieved from the ERP system. In the case examined and based on the historical data and the customer's requirements, the cost due to not meeting demand is 120 Euro per hour, while there is an additional fixed penalty of 1000 Euro. In this case, the noise in cost is attributed to incorrect data entered in the ERP system (low data quality) which affects the cost function and is estimated by analyzing historical data associated with data entries. In the case examined, low data quality is caused by the percentage of incorrect entries for late orders in the ERP system and by the percentage of actual orders from customers.

After several iterations of breakdown predictions and mitigating action implementations, the user can see the refined cost function in comparison to the initial configuration. In this case, after 7 decision epochs in which action "a2: Operate at reduced equipment load" was recommended and implemented, the refined cost function (derived from the aggregation of the two cost factors) is 428*(T-t) + 3300, as shown in Figure 9-9. In other words, the cost function derived from SEF is higher than the one configured by the user and consequently, the recommended action and the recommended implementation time may be different. In a subsequent decision epoch and by using the a2 cost function derived from SEF, the optimal action is now "a1: Lubrication of metal parts" and the optimal time is in 119 hours with a different resulting expected loss, since the global minimum is different, as shown in Figure 9-10.

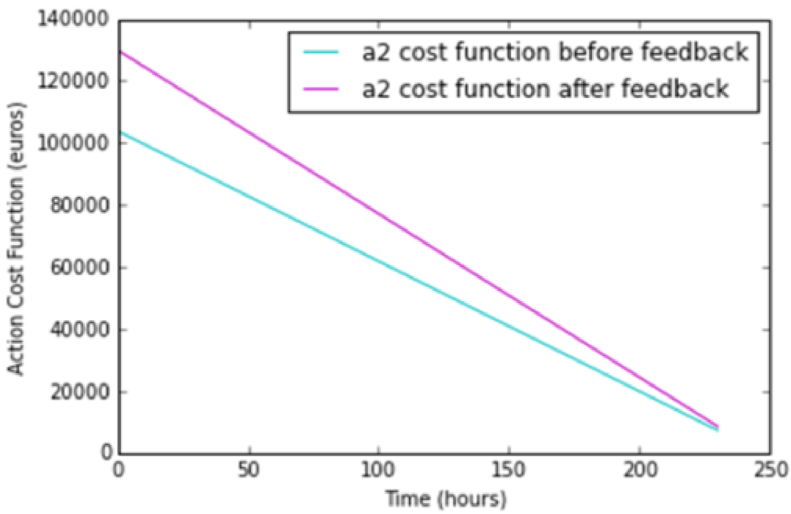

Figure 9-9: The cost functions of a2 before and after feedback.

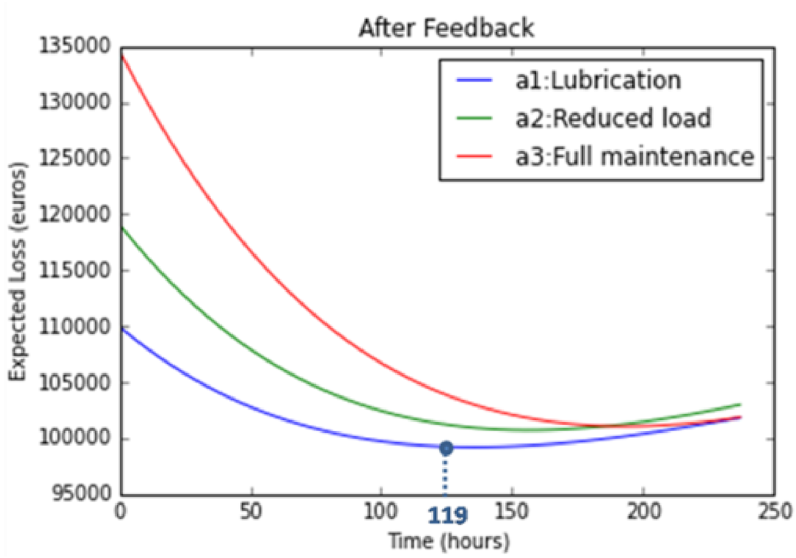

Figure 9-10: The expected loss of the alternative actions after SEF.

9.1.2.5 Use Case 5: Context-aware Proactive Recommendation for a Maintenance Action

The user creates a DMI about the gearbox breakdown. They insert the time until the next planned replacement of the gearbox (which corresponds to the end of decision epoch). Furthermore, they insert the contextual elements that affect the cost functions, that is the location of the rig (L) and the availability of service engineer (A) on the rig. These contextual elements take binary values and, depending on the value, they correspond to different cost functions with a specific probability. The probabilities of the values of the contextual element “location of the rig” indicate the amount of time that the rig will be near shore or far from shore respectively, since the location of the rig may change until the next planned gearbox replacement (end of decision epoch). This information can be collected from the production plan. Similarly, the probabilities of the values of the contextual element “availability of service engineer” indicate the amount of time that the service engineer will be on the rig until the next planned gearbox replacement (end of decision epoch). This information can be collected from the resource plan. Both plans are available at the Enterprise Resource Planning (ERP) system of the company. The aforementioned contextual elements are probabilistic because they are used in order to enrich proactive decision making rather than reactive. Therefore, the user cannot be sure about the values of the contextual elements at the time when the system will recommend the implementation of the action, i.e. the gearbox replacement, because they do not know in advance the recommended optimal time.

The probabilities that the user inserts at the configuration stage are required for the initialization and are derived by historical data analysis. Moreover, the user inserts the alternative values for each cost function. The cost function of breakdown is a linear function of time (per week) and includes also the cost of corrective actions that are required in case of a breakdown, while the cost function of planned replacement is a constant function. In the first case, there is high uncertainty about the duration of the equipment being down and of the corrective actions, since it is an unexpected event. In the second case, the duration of the equipment being down and of the planned replacement is known, since it is a planned action. Table 9-13 shows the input of the user based on which the BN is created. The BN gives the probability of a specific cost conditioned the context given. The BN that is created according to the user's input is shown in Figure 9-11.

Table 9-13: User's Input for Context and Costs

Context	Probability	Cost function of	Cost function of planned
Rig location			
Near shore	0.75	18,000*t	10,000
Far from shore	0.25	27,000*t	21,000
Availability of service engineer			
Available	0.83	18,000*t	10,000
Non available	0.17	27,000*t	21,000

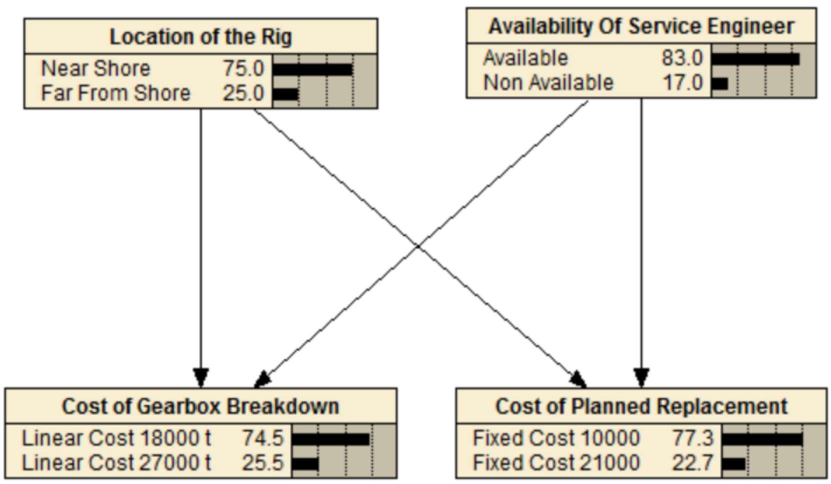

Figure 9-11: The Bayesian Network created according to the user's input.

Based on the result of the BN, the cost risk function gearbox breakdown and the cost risk function of planned gearbox replacement are calculated as shown below.

$$C_{ue} = 18000 * t * P(C_{ue} = 18000 * t | CE1 = L \cap CE2 = A) + 27000 * t * P(C_{ue} = 27000 * t | CE1 = L \cap CE2 = A)$$

$$C_{pa} = 10000 * P\left(C_{pa} = 10000 | CE1 = L \cap CE2 = A\right) + 21000 * P\left(C_{pa} = 21000 | CE1 = L \cap CE2 = A\right)$$

So, the cost risk function of gearbox breakdown and the cost risk function of the planned gearbox replacement are calculated respectively:

$$C_{ue} = 18000 * t * 0.745 + 27000 * t * 0.255 = 20{,}295 * t$$

$$C_{pa} = 10000 * 0.733 + 21000 * 0.227 = 12{,}097$$

After having calculated the cost risk functions according to the probabilities of the context, this output feeds into the proactive event-driven decision method. In this case, since the user wants to know the optimal time of applying a pre-defined action, the system uses the equation that calculates the expected cost rate based on the prediction event, i. e. the probability distribution function of the occurrence of the gearbox breakdown. So, 3 weeks before the end of decision epoch, a prediction event that the probability distribution function of the gearbox breakdown is exponential with the parameter λ = 1 / time-to-breakdown equal to ½ is received and the Expected Loss Rate is formulated accordingly.

$$Expected\ \ Loss\ Rate\ = \frac{20295 * t * \left(1 - e^{-0.5 * t}\right)}{2} + \frac{12097 * e^{-0.5 * t}}{3}$$

The equation is minimized by using the Brent's method and provides the recommendation that the optimal time for gearbox replacement is in 0.19 weeks, that is in 1.33 days, with an expected cost rate of 3842 euros per week or 549 euros per day, as shown in Figure 9-12. When the action implementation is finished, the SEF functionality gathers and processes data from the ERP system (software sensor) containing the values of the associated cost function accompanied with the context within the action of gearbox replacement took place. Thus, at that time, the rig was situated near shore and the service engineer was available on the rig, while the cost of the action was 10,000 euros. Considering this information, SEF updates the Bayesian cost risk calculation module (BN and cost risk functions) accordingly.

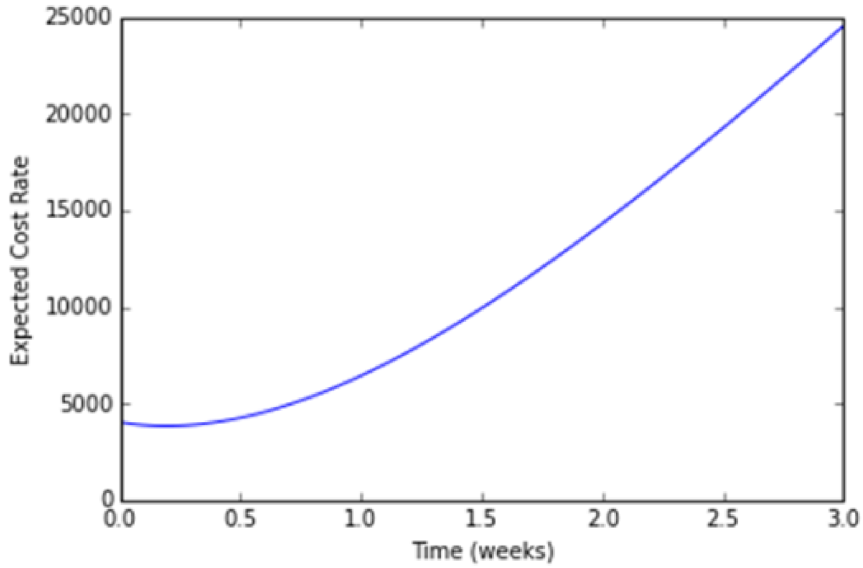

Figure 9-12: The expected loss rate in the time period between the prediction event and the end of decision epoch.

9.1.2.6 Use Case 6: Context-aware Proactive Recommendation of Joint Maintenance and Logistics Actions

User Configuration

The domain knowledge is inserted at design time through the PANDDA GUI. This domain knowledge has to do with the input parameters of the proactive decision model for joint maintenance and logistics optimization as well as the contextual elements affecting the costs along with their prior probabilities. In the current scenario, there are four alternative maintenance actions (lubrication of metal parts, operate at reduced equipment load, off-shore maintenance, full onshore maintenance) with different degrees of restoration and their associated orders of spare parts (lube oil, no ordering, gearbox, Derrick Drilling Ma-chine- DDM), as shown in Table 9-14. The time-to-failure after the implementation of the maintenance action indicates the degree of restoration. The actions a1, a2 and a3 are im-plemented on the oil rig (offshore), while onshore maintenance, which corresponds to per-fect ("good-as-new") maintenance, requires its movement onshore.

Table 9-14: The domain knowledge inserted during user configuration.

<table>
<tr><td colspan="2">Cost of failure (Euro)</td><td>350,000</td></tr>
<tr><td colspan="2">Decision horizon (hours)</td><td>240</td></tr>
<tr><td colspan="3">Maintenance actions</td></tr>
<tr><td rowspan="4">Time-to-failure after implementation (hours)</td><td>a1: Lubrication of metal parts</td><td>1,240</td></tr>
<tr><td>a2: Operate at reduced equipment load</td><td>2,050</td></tr>
<tr><td>a3: Offshore maintenance</td><td>2,960</td></tr>
<tr><td>a4: Onshore Maintenance</td><td>3,220</td></tr>
<tr><td colspan="3">Spare parts orders</td></tr>
<tr><td rowspan="4">Lead time (hours)</td><td>o1: Lube oil</td><td>5</td></tr>
<tr><td>o2: Swivel hook</td><td>8</td></tr>
<tr><td>o3: Gearbox</td><td>24</td></tr>
<tr><td>o4: DDM</td><td>48</td></tr>
</table>

Context-aware Model Initialization

Similarly to the previous use case, there are two contextual elements that are known to affect the specific equipment: the location of the rig and the availability of service engineer on the rig. The historical data needed for extracting the prior probabilities of the BN exist in the production plan and in the resource plan respectively of the ERP system of the company. According to this knowledge, the context-aware model is initialized. Therefore, two BNs are created: the first one deals with maintenance costs while the second one deals with inventory costs, as shown in Figure 9-13 and in Figure 9-14.

The context-aware costs that feed into the proactive decision model for joint maintenance and logistics optimization through Context-aware Reasoning are calculated according to the following equation:

$$C_n(t) = \sum_{i=1}^{i=2} C_{n,i}(t) * P\big(C_n(t) = C_{n,i}(t) | CE_1 \cap CE_2\big)$$

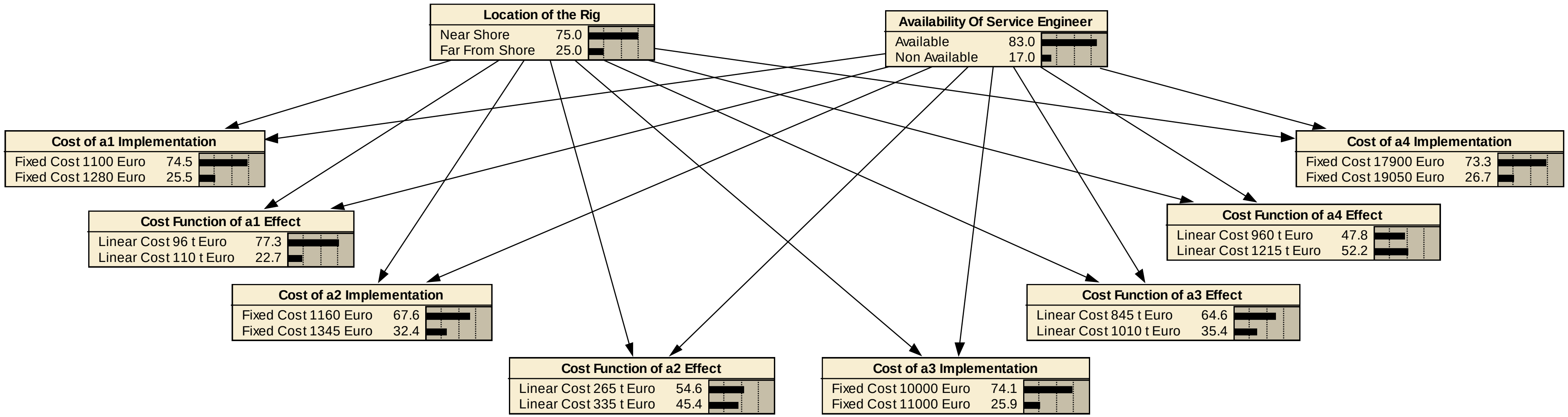

Figure 9-13: The initialization of the BN for the context-aware model for maintenance-related costs.

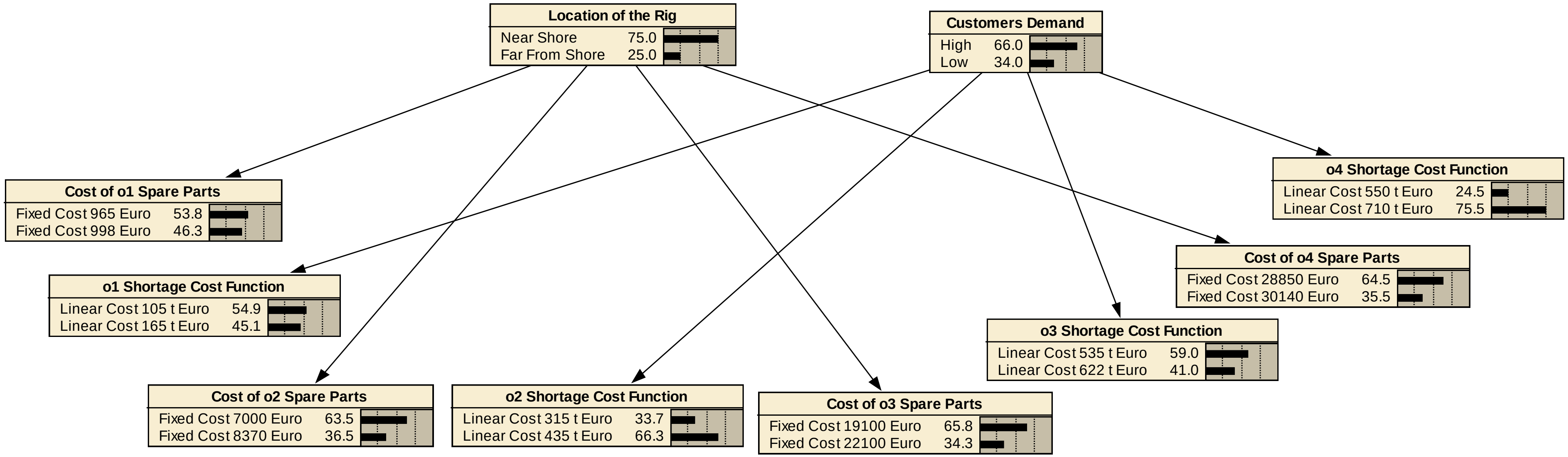

Figure 9-14: The initialization of the BN for the context-aware model for logistics-related costs.

Proactive decision model for joint maintenance and logistics optimization

At some time, a prediction event about an exponential distribution function of the failure occurrence with a parameter λ = 0.045 (i.e. expected time-to-failure is in 22 hours) triggers the decision algorithm. Taking into account the context-aware costs instead of the costs themselves, the joint maintenance and logistics proactive decision model are formulated as shown below:

$$EL^{a_i}(t) = \left[1-\left(1-e^{-\lambda t}\right)\right] * \left\{C_{a_i} + \left(1-e^{-\lambda \Delta t}\right) * C_f + \left[1-\left(1-e^{-\lambda \Delta t}\right)\right] * \left[C_{e_i}(T-t-\Delta t) + \left(e^{(t+\Delta t)(\lambda-\lambda\prime)} - e^{-\lambda' T+\lambda(t+\Delta t)}\right) * C_f\right]\right\} + \left(1-e^{-\lambda t}\right) * C_f$$

$$\begin{aligned} EL^{o_i}(t) = {} & \left[1-\left(1-e^{-\lambda(t+L)}\right)\right] \\ & * \left\{C_{sp} + \left(1-e^{-\lambda \Delta t}\right) * C_s(T-t-L) + \left[1-\left(1-e^{-\lambda \Delta t}\right)\right]\right. \\ & \left. * \left(e^{(t+L+\Delta t)(\lambda-\lambda\prime)} - e^{-\lambda' T+\lambda(t+L+\Delta t)}\right) * C_s(T-t-L-\Delta t)\right\} \\ & + \left(1-e^{-\lambda(t+L)}\right) * C_s(T) \end{aligned}$$

Although there is an indication of the most probable time-to-failure (parameter λ), the exponential degradation leads to high uncertainty in considering the deterministic value itself. Handling the probability distribution functions instead can lead to more accurate and reliable results. The Expected Loss Functions are shown in Figure 9-15 and their optimization results in the recommendation: Conduct offshore maintenance for gearbox replacement in 85.47 hours and order the Gearbox in 42.36 hours. These recommendations are exposed to the user through the GUI.

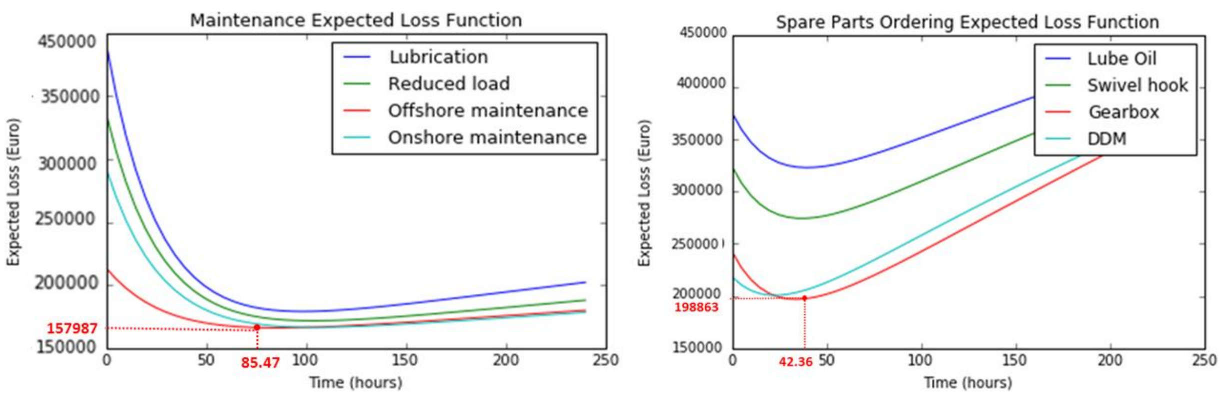

Figure 9-15: The expected loss functions for (a) maintenance, and (b) logistics (ordering of spare parts).

Sensor-Enabled Feedback and Context-aware Model Reasoning

As long as more executions of the proposed algorithm are conducted, more data are gathered through SEF in order to update the context-awareness mechanism and thus, to improve the generated recommendations. In each effect node, the X-means clustering algorithm creates clusters and separates the feedback values to them. In each algorithm trigger, the centroid of each cluster is taken into account. Figure 9-16 shows an example of an effect node ("Cost of a1 Implementation") after 40 executions, where there are two clusters. The real-time cost information is exposed to the user through the GUI.

This real-time update of visualization is achieved due to the server-push and event-based publish/subscribe and a highly scalable real-time graphing system which is able to store thousands of time-series per second and compute metrics on them. After many executions of the proposed model, the expected loss functions on the basis of the updated cluster centroids for the same prediction event are derived as shown in Figure 9-17. In this case, the recommendation is to conduct full, onshore maintenance in 98.26 hours and to order the DDM in 49.12 hours. This recommendation leads to a lower expected loss compared to the recommendation shown in Figure 9-15, while it is more accurate since it is based on sensor data instead of human's subjective estimations.

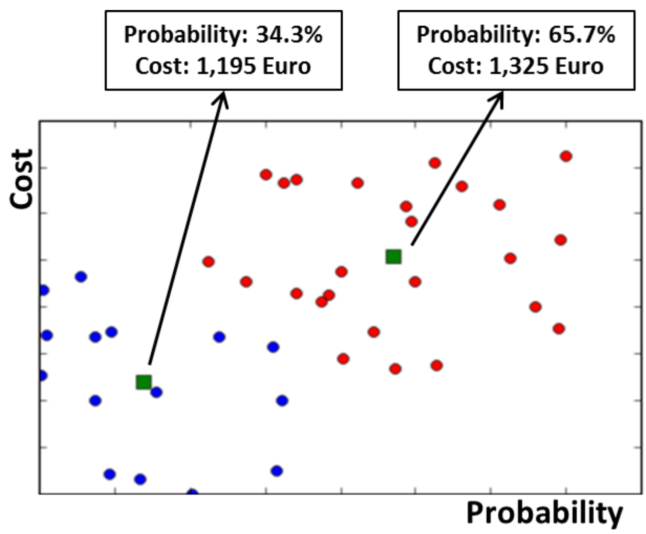

Figure 9-16: An example of the BN effect node "Cost of a1 Implementation" after 40 executions.

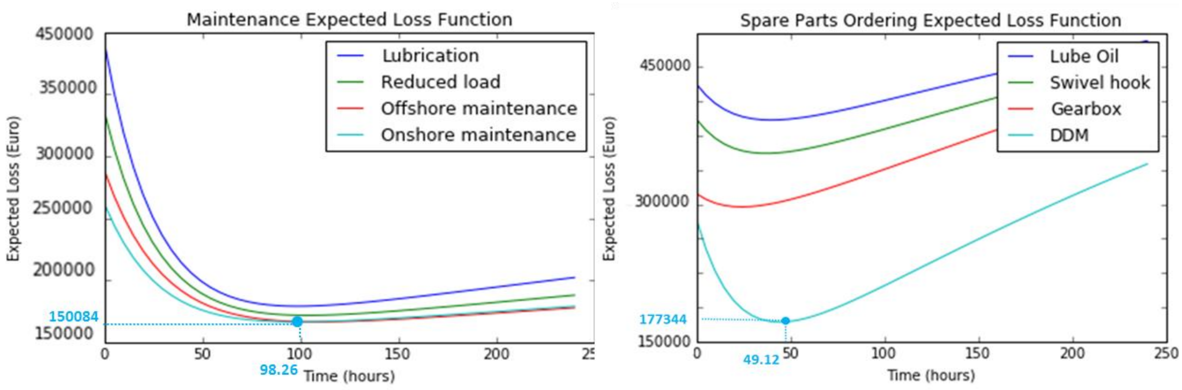

Figure 9-17: The expected loss function after many executions for (a) maintenance, and (b) logistics.

9.2 The HELLA Business Case

9.2.1 Description of the HELLA Business Case

HELLA Saturnus Slovenija is based in Ljubljana and is part of a corporate group, an international headquartered in Germany. It is one of the biggest Slovenian exporters. Company's core business is the development and production of a wide range of top-level lighting equipment products for motor vehicles: headlamps, auxiliary fog lamps, daytime running lamps and single- and multifunction lamps. It now employs approximately 1,900 people and generates more than EUR 200 million of revenues. The work is organized in 3 daily shifts in approximately 300 days per year. The factory produces 2.5 million headlamps and 6.5 million fog lamps annually.

The production process includes different process steps from supplier deliveries, warehousing, plastics injection moulding, surface treatment, metalizing, preassembly of groups and finish goods assembly. There are several inline measurement processes involved that gather information of the quality of parts exiting particular phases. In addition data is also being collected at the particular process level. There are additional parameters influencing the effectiveness of the production line like ambient information, material structure information, personnel working at the line etc. The installed monitoring functionality mainly shows the current status and some most important trends that are being influenced mainly by the tool wear and machine configuration.

The company has established fully automated plastic cover lens production facility including injection moulding and lacquering. Following the path of innovation, it intends to deploy the ProaSense solution with the aim to lower the scrap rate and boost productivity on the same facility. It expects to directly benefit from ProaSense results with the expansion to new markets and new products and new technologies (e.g., laser-based automotive lighting). By introducing new technologies that will complement the existing setup, it expects to gain significant competitive advantage against major competitors.

The optimization of the manufacturing lines certainly tackles company's strategy for energy positive business and nature preservation policy. Integrating different data with the inline data will enable the company to understand better the dependencies between different factors that will lower the scrap rate, waste and energy consumption costs that are currently high because of intensive production. Furthermore, the implementation of a sensor network on top of the existing infrastructure, as shown in Figure 9-18, will enable the company continue the company development towards intelligent manufacturing.

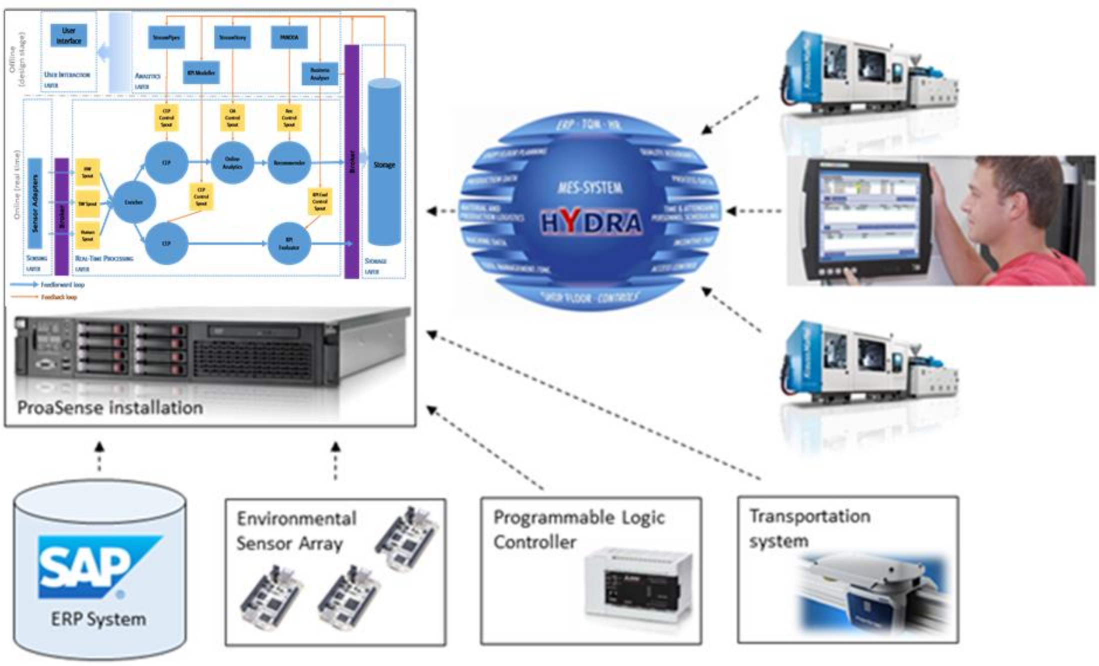

Figure 9-18: The ProaSense system in HELLA.

The potential of ProaSense towards minimizing the weaknesses and avoiding the threats of the automotive lighting equipment company is explained through the SWOT analysis of Table 9-15. By analysing both the strength and weaknesses with respect to opportunities and threats, we get the elements of a strategy presented in Table 9-16. Table 9-15 and Table 9-16 were created in conjunction with the pilot company under examination.

Table 9-15: SWOT Analysis for HELLA.

Strengths	Weaknesses
The company is a producer of high quality and high volume automotive lighting equipment in a very competitive market segment. It has always been innovative and is therefore still a key player. It has operators that are experts in moulding and lens production.	Part of the scrap production and downtime is due to the fact that our customers demand (norms) high quality, regarding functionality and appearance. Downtime appears during maintenance of complex production processes, which results in a lower overall equipment efficiency (OEE). Customers demand more complex products that will increase the scrap rate and downtime. Currently they are not able to analyse the machines that are offline in the production process. They have only aggregated data on, e.g., the scrap rate.
Opportunities	**Threats**
Automotive lighting equipment is more and more advanced and new technologies are being introduced (i.e. LED, OLED, LASER, etc.). With their innovative approach, they could be one step in front of the competitors and offer their customers cutting edge products. They have the opportunity to put all of our production machines online, reduce cost and reduce downtime, so they deliver even better performance and keep their competitive advantage.	Introducing new technologies into automotive lighting equipment also headlamp production process becomes even more complex. In the future, they might not be able to master the process with reasonable OEE. Other competitors that can produce more precise products, for lower costs.

Table 9-16: Strategy derived from SWOT Analysis for HELLA.

	Opportunities	Threats
Strengths	Considering company's innovative approach, the knowledge that is available on offline analysis of the moulding machine, it foresee that ProaSense is the logical next step.	With the use of the experts on moulding in combination with the new feedback ProaSense can give them, we will be able to deliver higher quality products with less scrap produced and less downtime.
Weaknesses	ProaSense will enable them to master even more complex production processes and improve OEE. Also ProaSense will enable to give more feedback that is not possible today. Even before a breakdown/downtime or scrap occurs.	ProaSense will enable them to at least maintain current OEE value with introduction of even more complex products and production processes.

Based on extensive root cause analysis using Bow Tie diagrams for various defects, the most critical ones were selected. Therefore, the automotive lighting equipment industry use case focuses on the cover lens component of the headlamp components. Headlamps consist of several components. The components are assembled during the assembly process. Workers first place components on a headlamp-specific stand and then the robots assemble and glue the headlamp. In order to enable semi-automated headlamp assembly process, a stable production process of components must be ensured. Next to headlamp assembly process, shown in Figure 9-19, the company also incorporates the production and treatment of all plastic components. It uses more than 60 different raw plastic materials for component production, each with its own properties. Component treatment includes metalizing and lacquering. One of headlamp components is the cover lens. Cover lens production process consists of two main steps: moulding and lacquering. The moulding process ensures the correct geometry of the lens while lacquering ensures the resistance to outer vehicle environment. An example of a cover lens is shown in Figure 9-20.

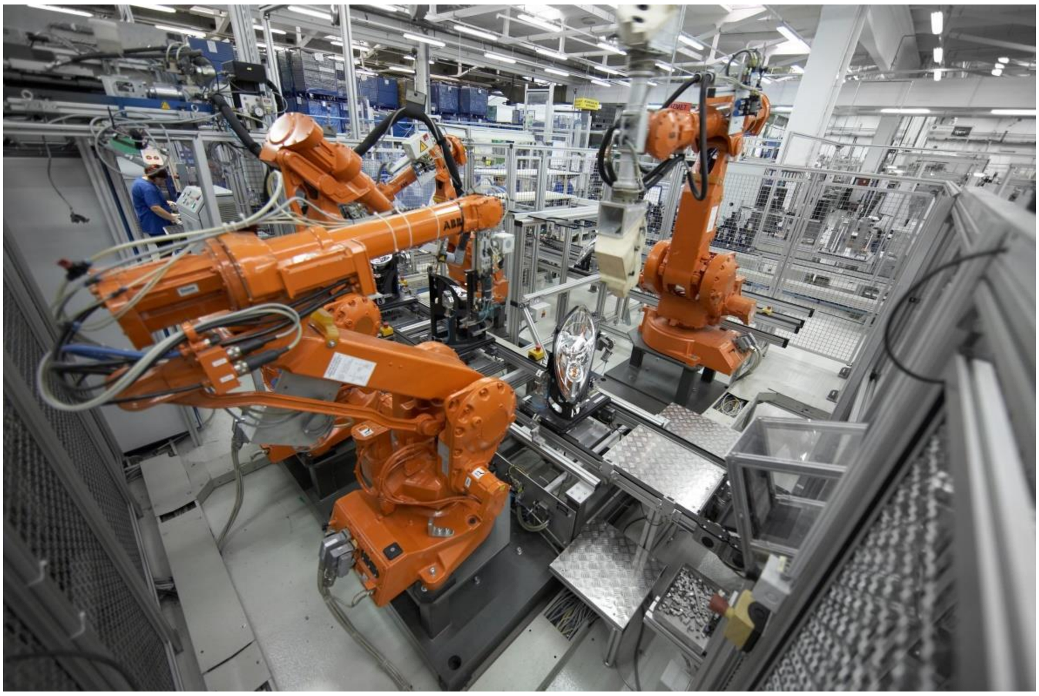

Figure 9-19: The headlamp assembly process.

Cover lens is one of the most important components of headlamp for two reasons. The first reason is that cover lens completes the outer surface of the vehicle, which is very important to our customers. The customers spend a lot of effort in completing the outer surface also for the best aerodynamic properties of the vehicle. Therefore only a small range of geometrical deviation for the cover lens is allowed according to the customer requirements. Additionally, the cover lens is during the assembly process glued to the housing. Since the headlamp must be watertight, the cover lens must fit into the housing's gluing channel hence the geometry of the lens must be constant i.e. the production process must be stable. The second reason for special focus on cover lens is that since it represents the outer surface it has to be free of all decorative defects. Decorative defects on injection moulding parts are not avoidable and result in scraped parts. Scrap rate depends on the stability of the production process. Scrap related expenses are to some extent also covered by the customers. Expenses that are not covered by the customers are naturally covered by the company.

Figure 9-20: Example of a cover lens.

9.2.2 Deployment in HELLA

9.2.2.1 Use Case 1: Proactive Recommendation of Joint Maintenance and Logistics Actions and Supplier Selection

The production process includes the production of the headlamps' components and their assembly with automated transporting. The manufacturing process of cover lens collects a multitude of data through sensors established in moulding and lacquering phase of the production line. The reliability of this manufacturing process is critical, due to the volume of production and the complexity of the products. These processes gather many data about the various production phases mostly through embedded quality assessment equipment using sensors and measuring devices. Since the volume of the production process is high and the equipment for the production of complex parts is expensive, the improvement in detecting, predicting and eliminating failures or mitigating their impact can be measured in tens of thousands of Euros. For example, a reduction of the scrap rate by just 1 %, would result in savings of the order of magnitude of 100,000 euros per year.

One of headlamp components is the cover lens. Cover lens production process consists of two main steps: moulding and lacquering. The moulding process ensures the correct geometry of the lens while lacquering ensures the resistance to outer vehicle environment. The undesired event that should be mitigated is the level of scrap rate exceeding 25%. The “as-is” situation of the company is a time-based maintenance strategy including cleaning of the moulding machine from dust and conducted every Monday and Thursday at 9:00. The automotive lighting equipment company aims to turn from time-based cleaning of the moulding machine into condition-based and, at the same time, to be able to order the spare parts just-in-time and to decide proactively about the portfolio of its suppliers. In this way, the maintenance and inventory costs can be reduced by eliminating, at the same time, the risk of a high scrap rate and of a shortage cost of spare parts.

First, the user configures a DMI through the PANDDA GUI. The desired recommendation is about the optimal time for cleaning (mitigating action) and the optimal time for ordering the spare parts (prerequisite action). Furthermore, the selection of maintenance spare parts suppliers is required. Therefore, the user inserts the available number of suppliers and the budget dedicated to the purchasing department. Therefore, the Joint Expected Losses optimization method is selected along with the supplier selection method and the relevant DMI for the moulding machine scrap rate is created through the PANDDA GUI. The input parameters of the joint maintenance and logistics decision model inserted by the user are shown in Table 9-17.

The planned maintenance cost is 325 euros and lasts for 1 hour, while the failure cost, that is the cost due to scrap rate (which also includes the cost of corrective actions), is 85 euros per hour. So, there is a fixed planned maintenance cost equal to 325 euros and a linear increasing failure and corrective cost equal to 85*t. The shortage cost is 140 euros per hour, the holding cost is 65 euros per hour and the lead time L is equal to 2 hours. Next planned maintenance (cleaning of the moulding machine) is in 10 hours.

Table 9-17: Input by the user of the moulding machine scrap rate instance

Corrective cost of mitigating action	85 euros / hour
Planned cost for mitigating action	325 euros
Cost due to early undesired event	140 euros / hour
Cost due to undesired event not occurring	65 euros / hour
Planned time for implementation	10 hours
Lead time	2 hours

At the Real-time Processing Layer of the platform, real-time smart sensing of dust level and environmental factors such as humidity and temperature is conducted. These factors are known to affect the operation of the moulding machine and thus, the level of cover lens scrap rate. At some time, abnormal levels of dust, humidity and temperature are detected on the basis of the observed data that indicate the deterioration of the moulding machine's operation. Therefore, there is a real-time prediction about the scrap rate exceeding 25% 5 hours after the start of the decision horizon. This prediction event that the remaining life distribution is exponential with expected time-to-failure equal to 4 hours (λ=0.25) triggers PANDDA which is enacted online and provides the recommendation shown in Table 9-18. Figure 9-21 shows the following resulting expected loss functions.

$$C_m(t) = (85 * t) * (1 - e^{-0.25*t}) + (85 * t + 325) * \left(1 - e^{-0.25*(5-t)}\right) + 325 * \left(e^{-0.25*t} + e^{-0.25*(5-t)} - 1\right)$$

$$C_o(t) = (140 * t) * \left(1 - e^{-0.25*(t+2)}\right) + 140 * t * \left(1 - e^{-0.25*(5-t-2)}\right) + 75 * t * \left(e^{-0.25*(t+2)} + e^{-0.25*(5-t-2)} - 1\right)$$

Table 9-18: Real-time input and output of the moulding machine scrap rate instance

Predicted probability distribution	Exponential with λ = 0.25
Recommended mitigating action	Clean the moulding machine
Recommended prerequisite action	Order spare parts (i.e. moulds)
Recommended implementation time for the mitigating action	In 3.54 hours
Recommended implementation time for the prerequisite action	In 1.32 hours

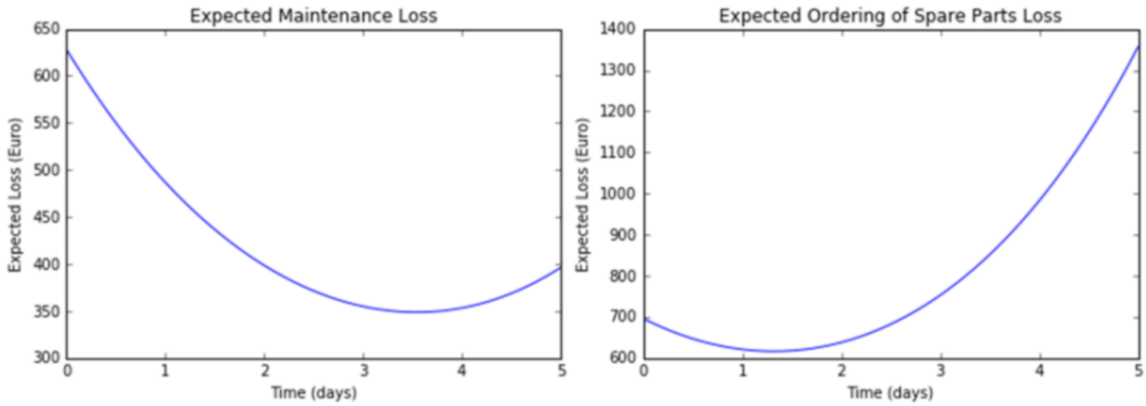

Figure 9-21: The result of the optimization algorithm for the optimal time for (a) mitigating action (moulding machine cleaning); (b) prerequisite action (ordering of spare parts).

The optimization of the aforementioned equation gives a recommendation that the optimal time for maintenance (cleaning of the moulds) is in 3.54 hours with a cost of 348.8 euros. The optimization of Equation 6 gives a recommendation that the optimal time for ordering the spare parts is in 1.32 hours with a cost of 616.6 euros. On the basis of this recommendation, the negotiation of the company with 4 suppliers, the available purchasing budget and the last update of the prediction of spare parts' prices (shown in Figure 9-22), the optimal portfolio of suppliers can be recommended. This information along with suppliers-related data (inventory level, scheduled production plan, capacity, etc.) is continuously updated in ERP through EDI.

The proactive supplier selection method provides the 'Markowitz bullet' and its 'efficient frontier' shown in Figure 9-23. Based on this, the optimal portfolio of suppliers is recommended, as shown in Table 9-19. The recommended portfolio actually presents the percentages of the available purchasing budget that should be dedicated to each one out of the available suppliers in order to optimize the costs and to eliminate the risk of not delivering the spare parts or of their delivering late. Therefore, the 14 % of the approved purchasing budget should be allocated to Supplier A, the 38 % to Supplier B, the 26 % to Supplier C and the 22 % to Supplier D.

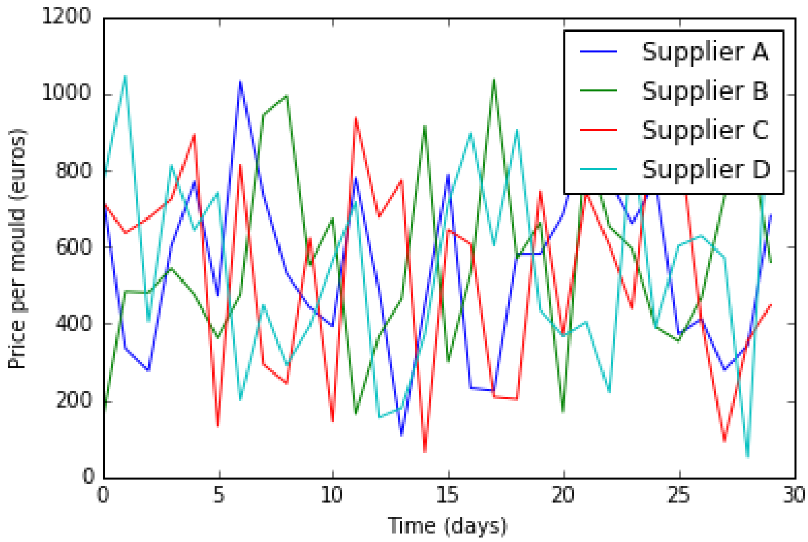

Figure 9-22: The prices prediction in the course of time until the decision horizon.

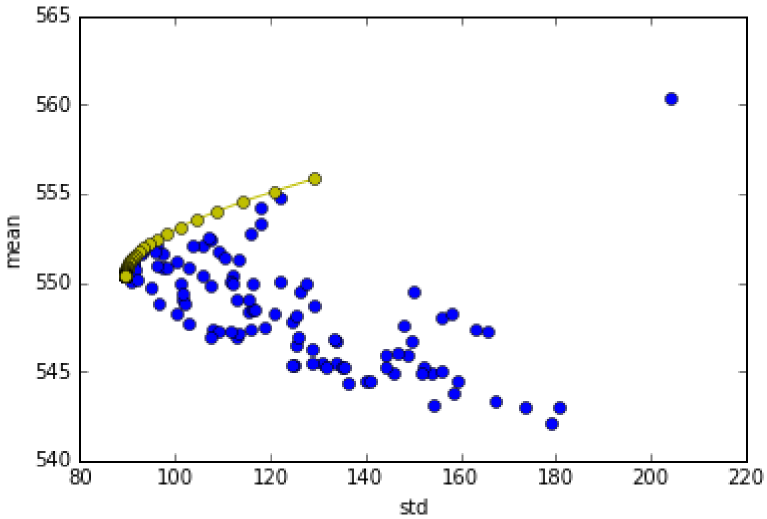

Figure 9-23: The Markowitz bullet and its Efficient Frontier for the portfolios.

Table 9-19: The optimal portfolio of suppliers.

Supplier A	Supplier B	Supplier C	Supplier D
0.14	0.38	0.26	0.22

10 Evaluation

In this Chapter, the evaluation results are presented. First, the PANDDA system is evaluated by industrial users through questionnaires and free text for expressing their views. In addition, the system performance is evaluated in order to prove its efficiency and scalability. Moreover, results from extensive simulation experiments of the functionalities are presented. Due to the large timescales and manufacturing processes' lifecycle, it is not usually possible to evaluate sufficiently new systems, algorithms, methods and approaches in the context of a real industrial environment, during the actual operation of manufacturing processes. For this reason, a simulated computational environment was created in order to evaluate the proposed approach, system, algorithms and methods for cases that did not arise during the evaluation period. To this end, comparative and sensitivity analyses show the added value of the proposed approach.

10.1 System Evaluation by Industrial Users

The PANDDA system was evaluated by the users in the aforementioned manufacturing companies in two ways: first, through a web survey incorporating questionnaires about the usability, the usefulness and the installation of the system; second, in a qualitative way, by reporting their views, conclusions and lessons learned. The evaluation was conducted in two iterations in the context of an agile mode of software development. The results of the first round of evaluation were taken into account for improvements in the already existing functionalities and the development of new ones. Moreover, they were considered for improvements regarding systemsperformance and usability.

10.1.1 Questionnaire-based Evaluation

10.1.1.1 Methodology of Evaluation Results Analysis

In order to enable evaluators to become familiar with the system before performing the evaluation, manuals and videos demonstrating typical user interactions with the components were developed for each one of them. Moreover, scripting trials were described for each component, so that the users are guided to the screens and functionalities. A web survey incorporating the questionnaires was developed. The questionnaire regarding PANDDA's usefulness and usability is available at Appendix A: Questionnaire for Evaluation of PANDDA.

Most of the questions were formulated in the form of statements to which the participants were asked to specify their level of disagreement or agreement on a 5-point Likert scale from 1 (strongly disagree) to 5 (strongly agree), while there were also some yes/no and optional free text questions. As part of the methodology, a Positive Feedback Indicator (PFI) was defined for indicating positive/negative feedback on the level of question with respect to usability and usefulness.

However, Likert scales produce ordinal data (i.e. data that can be ranked), which cannot yield mean values. Therefore, A value of PFI less than 50% indicates a question that received more negative than positive feedback, since most of the responses were bad, and vice versa. A PFI value of 50% was also considered problematic. A similar approach was followed to calculate a PFI for yes/no questions.

10.1.1.2 Analysis of Questionnaire-based Results

Regarding the questionnaire about usefulness, with the exception of the question PP4 which was a free text question, the rest of the aforementioned questions were formulated in the form of Likert scale statements. The evaluation results were positive, although the users' level of expertise related to maintenance decision support applications was quite low, as a result of the fact the proactive decision making is an underexplored area in the academic and the industrial realms. Figure 10-1 shows the Positive Feedback Indicators for all questions, which are above the threshold (i.e. 50%).

Regarding the questionnaire about usability, with the exception of the question PP4 which was a free text question, the rest of the aforementioned questions were formulated in the form of Likert scale statements. Figure 10-2 shows the Positive Feedback Indicators for all the questions. The Positive Feedback Indicators of the questions PP1 ("My level of expertise related to maintenance decision support applications is high.") and PP3 ("The meaning of decision method instance in the PANDDA system is understandable to me.")

are the only ones below 50%. While the results of PP1 are not negative with respect to the Usability of PANDDA (since they are more general questions that aim to identify the current status with respect to experience and expertise maintenance decision support in the organizations and to enable the explanation of the results), the results of PP3 show a lack of understanding for the 'decision method instance' concept.

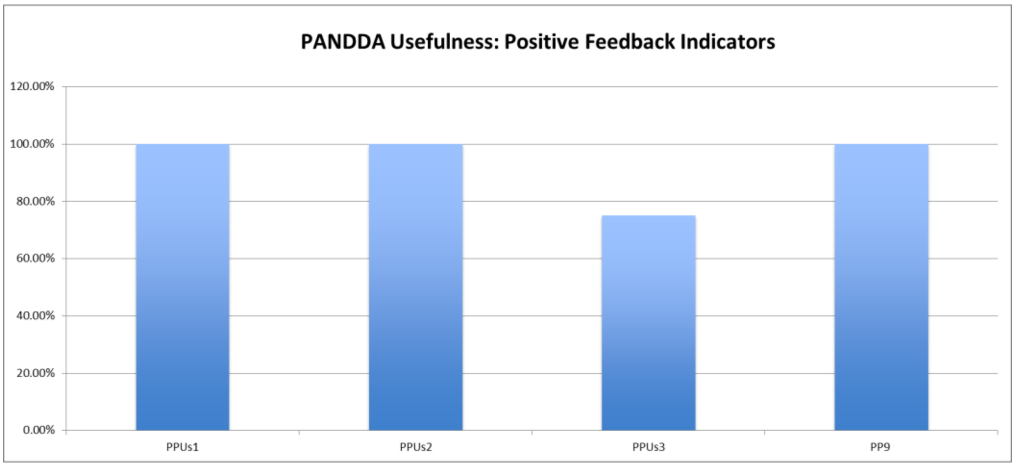

Figure 10-1: The PFI for the usefulness questions.

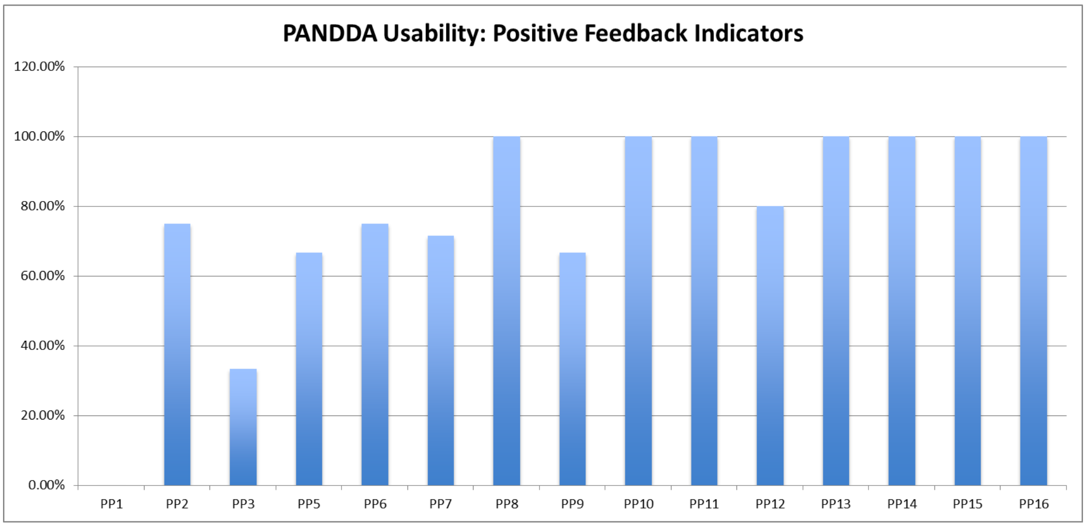

Figure 10-2: The PFI for the usability questions.

10.1.2 Qualitative Evaluation: The Pilots' Views

10.1.2.1 MHWirth

The company's goal was to utilize data stored in Riglogger™ to support the development of new, value-adding services. This has been materialized by focusing on advanced condition monitoring methods for critical machines and performance monitoring. Monitoring methods for the Top Drive are essential for revealing degradation and upcoming issues. These techniques are also seen as major differentiators for the company compared to its competitors. Combined with the customer increasing interest in the field of condition monitoring, patents proving new functionality and overall focus on Big Data are three considerable selling points for new MHWirth drilling equipment. Capabilities within decision support and cost optimization of both maintenance tasks and best time of execution are also highly relevant for company's customers. Furthermore, the new infrastructure possesses cost optimization abilities which were not previously understood as relevant for the exploitation of Riglogger™ data. Hence, a major effort was required to provide sufficient relevant contextual data covering cost scenarios for equipment failures to complete the proactivity principle. Still, overall maintenance optimizing functionality and integrated notification functionality has proven useful to comply with company's overall target of reducing equipment's downtime for critical machinery. The system provides accurate and reliable information about equipment condition to both improve and simplify the decision-making by drilling operators and maintenance planners.

Hosting modern ICT solutions within an existing operating IT environment has proven a considerable undertaking throughout the development and testing. The fact that the infrastructure is Linux based and build on open-source components have challenged company's corporate privacy and security policies considerably. Combined with network and software operations outsourced to a third party vendor have made required granting processes for exceptions difficult and time consuming. However, the use of Docker and well documented the result become successful installation executed with only limited time delays.

The system is perceived both flexible and robust to handle a wide range of applications. Outlining use cases across the phases of the proactivity principle has matured the organization establishing new requirements for the collection of context data to be used in optimization of maintenance recommendations, both related to type of activity and timing to minimize equipment lifecycle cost. The company possesses considerable amount of Riglogger™ data expected to have great value for both reactive and proactive service providing. Still, complete exploitation of the Proactive Maintenance system assumes high quality contextual information such as PDFs for top drive breakdowns. Getting hold of such kind of information has proven highly challenging. Another challenge experienced through the project is documenting the results of reducing equipment downtime economically in business KPIs. The recent customer interest into performance based contracts both for the drilling rig owners towards the oil companies and the contracts between the rig owner and the equipment supplier are highly interesting and need to be pursued. A shift in the conservative and traditional business model will greatly affect the pace, which the company needs to control equipment operational status to support future incentives for both rig owners and oil companies.

The system development process facilitated through the outlining of use cases have revealed new understanding internally of the challenge in cooperating across business domains. Adopting key competency within both drilling equipment engineering and computer science is critical to succeed in big data analytics. MHWirth has through the work with Proasense acquired increased understanding of the complexity in system architecture and revealed the need of dedicated system users with both business optimization competency and technical equipment understanding to utilize the full potential of such analytical tools. Hence, the need of organizational development in the field of business analytics is provided as input to the company's business development strategy aiming to exploit the full potential of the real-time data currently collected in Riglogger™.

10.1.2.2 HELLA

"Zero defects" is one of the most important goals of every company that wants to be among the TOP3 in its industry. Defects in the company's use case consist of downtimes and scrap rate. The production process includes different process steps from supplier deliveries, warehousing, plastics injection moulding, surface treatment, metalizing, preassembly of groups and finish goods assembly. Several inline measurement processes are involved in gathering information of the quality of parts in exiting particular phases. In addition, data is collected at the particular process level. There are additional parameters influencing the effectiveness of the production line like ambient information, material structure information, personnel working at the line etc. This is why there is a clear need to understand the wider context and to be aware of the overall situation in the shop floor. In particular, the aim was to have a model that can be used for predicting potential errors and to provide proactive solutions ahead. The installed monitoring functionality mainly shows the current status and some most important trends that are being influenced mainly by the tool wear and machine configuration. Before the Proactive Maintenance system, the company had never had the chance to work on a system to identify new correlations between different factors, which influence various defects. To cope with such huge amount of data, a configurable system is required. With the Proactive Maintenance system, the company defined a use case and started to prove correlations between scrap rate and its root causes. The correlations identified will first hand allow us to avoid clear defect-causing combinations (where applicable, e.g., never use certain set of machine-product) and, second-hand, allow us to adapt the process to avoid predicted defects (e.g., set different injection moulding parameters for the night shift on moulding machine no. 2 for product no. 23).

Large enterprises usually have several small subsidiary companies all over the world. There are different kind of corporate governments, which can lead subsidiary companies from headquarter (centrally) or leave the subsidiary company to be guided by the local management. The company is becoming more and more centrally guided. Consequently, as a subsidiary, it has limitations due to central management decisions, but also through technology and new system implementations. The company's headquarters are preparing guidelines on which system will be used for ERP, MES or SCADA and which the rest of the companies over the world need to follow.

In case one of subsidiary company wants to implement a new system, it must first check with headquarters if the system in question has enough potential for first local and then worldwide distribution. This is where the company had a strong limitation. With the new perspective of Industrie 4.0 and the Internet of Things, there is a need for integration of new sensing equipment. Considering also the support for legacy equipment, investment is significant and has to be approved by the headquarters. For administration and further adaptation of big complex systems, the company has to create a new position in organisation. At least one person with certain competences has to be recruited and further trained. There are costs involved that need to be considered and approved. In order to build a reliable process model, there is a need for a big amount of historical data.

For deployment, the company had to stop the production process leading into downtime. It chose the use case of thermoplastic cover lens production, which represented a huge challenge. The department is fully automated and is running on a four-shift model. The company also fully booked capacity on injection moulding machine, therefore additional downtimes are not acceptable. This is one of the key reasons why large enterprises are slowly adopting new technologies. Security of the corporate data is one of the most important requirements for new enterprises. This has to be considered when adopting new technologies.

Reduction of costs in every corner, complex data analyzing, etc., are the guidance for every company that wants to survive on a market more aggressive every year. The company can extend project on complete component production area. There are similar problems on injection moulding machines as well as on lacquering line and metallization. Next to the existing use cases from component production process and the data gathered there, the company intends to extend the use of the Proactive Maintenance system to final headlamp production. Final product (headlamp, rear lamp, fog lamp, etc.) consists of several components, some of them (optical system/group, e.g., reflector) being responsible for correct light distribution.

Reflector is first being moulded, then it might be lacquered (depends on the material of reflector) and finally metalized. After the final product is assembled, it is automatically tested for optical properties (light intensity, position of cut-off line (COL), sharpness of COL, etc.) and evaluated as OK/NOK. NOK parts result as scrap. The main (non) quality contributors are moulding and lacquering processes (parameters) and should therefore be monitored and adapted. In the future, the company aims to further increase the quality of the final product (average light intensity in photometric points reserve over legal values) and/or reduce the price of the product.

10.2 System Performance Evaluation

In order to properly test the PANDDA application, test cases simulating user interactions with the web-based PANDDA application were developed. The data for this type of testing, has originated from testing scripts that simulate user interactions. Such scripts have been developed with the use of the Selenium toolbox (www.selenium.org). Selenium script language provides many options for locating UI elements and comparing expected test results against actual application behavior, while it also allows executing the tests on multiple browser platforms and check for potential problems that may arise.

The PANDDA test cases were developed and executed using the Selenium IDE (Integrated Development Environment) in particular, which is implemented as Mozilla Firefox plugin. It has a recording feature, which records user actions as they are performed and then exports them as a reusable script in one of many programming languages that can be later executed. Additional verification commands can be added manually to the recorded scripts. A screenshot of Selenium IDE containing a PANDDA test suite (on the left panel) and a PANDDA test case (on the right panel) can be seen in Figure 10-3.

The test suite has been executed many times from different platforms and browsers, revealing some bugs that were successfully resolved. Finally, the example test suite has been successfully executed, as can be seen in Figure 10-4. This means that the PANDDA configurator behaves as expected. Moreover, the PANDDA system performance was tested in terms of its latency with the following hardware specifications: Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz, Ubuntu 16.04, 4 Cores VM, 8GB RAM. As shown in Figure 10-5, the time needed to process the real-time data and information of actions is an almost linear function of their number.

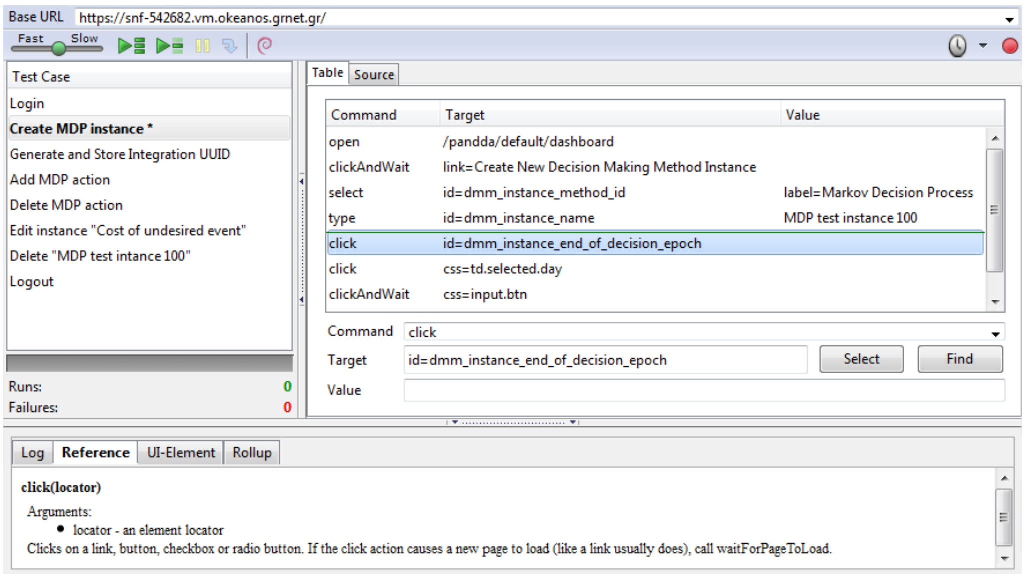

Figure 10-3: PANDDA Selenium test suite and test case

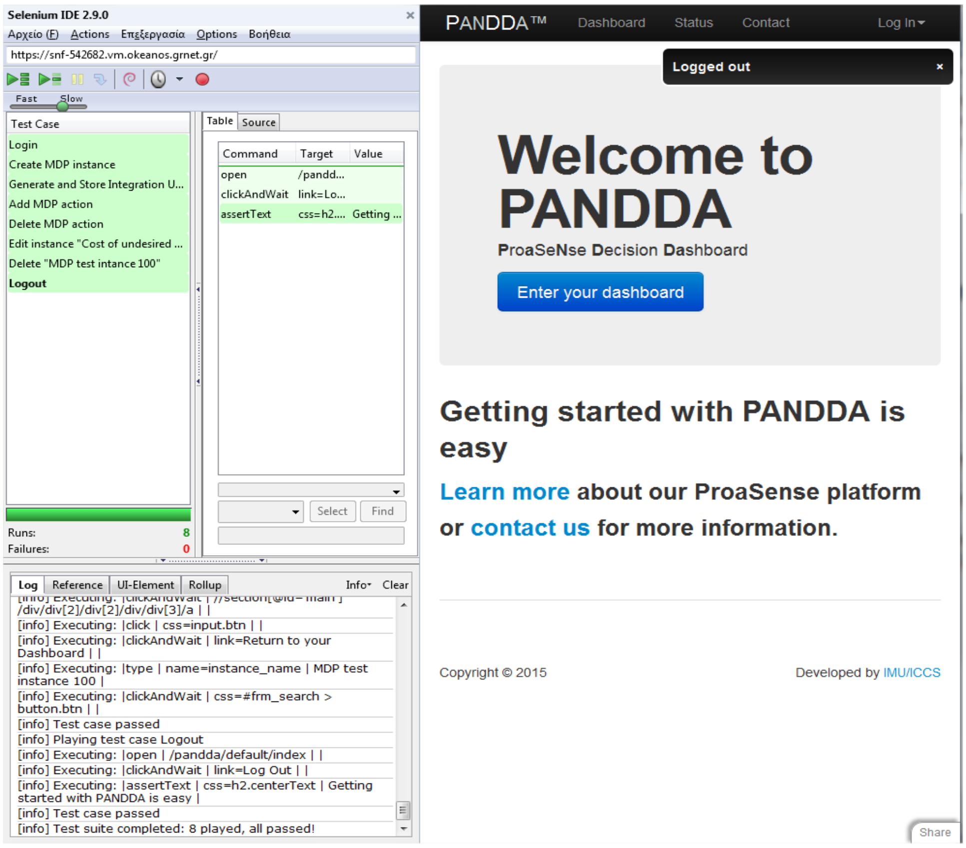

Figure 10-4: An example of a successful PANDDA test suite execution.

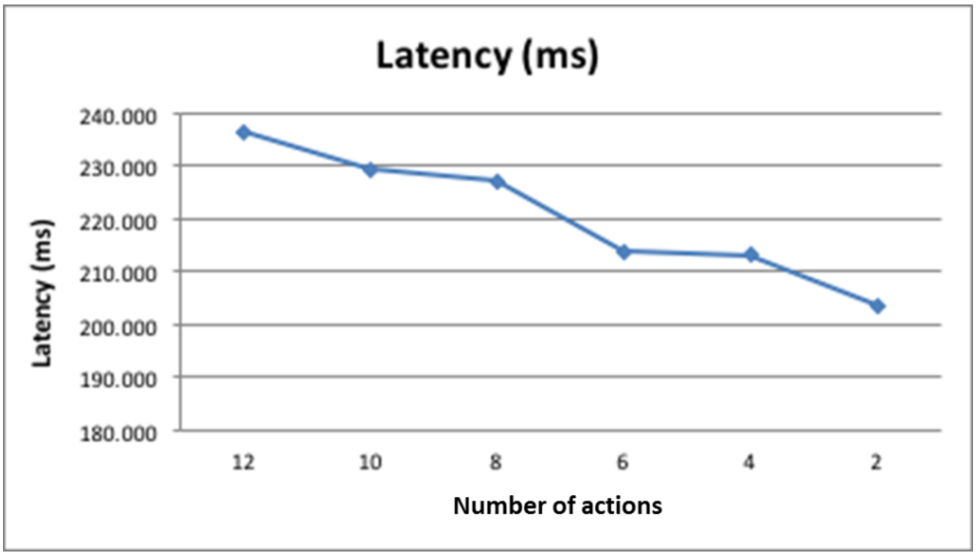

Figure 10-5: Latency of PANDDA

10.3 Sensitivity and Comparative Analysis of the Implemented Functionalities

Due to the large timescales and manufacturing processes' lifecycle, it is not usually possible to evaluate sufficiently new systems, algorithms, methods and approaches in the context of a real industrial environment, during the actual operation of manufacturing processes. For this reason, a simulated computational environment was created in order to evaluate the proposed approach, system, algorithms and methods for cases that did not arise during the evaluation period. To this end, comparative and sensitivity analyses show the added value of the proposed approach.

10.3.1 Proactive Decision Making

10.3.1.1 Sensitivity Analysis of Proactive Decision Making

All the developed decision methods deal with uncertainty in order to provide real-time, event-driven proactive recommendations. We conducted a sensitivity analysis that examines the output recommendations of the aforementioned proactive event-driven decision methods for various input parameters. Sensitivity analysis with plots showing the output for various values of input parameters (e.g. under scenarios of various cost structures of costs of predictive and corrective actions) has been widely used for testing and validating decision models in the manufacturing domain (Wu et al., 2007; Elwany, and Gebraeel, 2008; Engel et al., 2012; de Almeida, et al., 2015; Wang et al., 2015). The advantage of this method is that it can also deal with arbitrary ranges of values (e.g. when there are not constraints about the minimum and maximum values of input parameters) and gives a direct visual indication of sensitivity (Hamby, 1994; Paruolo et al., 2013). For each method and each parameter we change, we keep all the other parameters required by the user constant and we plot two diagrams. The first one presents the optimal expected loss of the recommended action for various values of the input parameters examined as a function of the prediction event parameter for all the possible values of prediction event parameters. We assume that the prediction includes an exponential probability distribution function and thus, the prediction event parameter is λ = 1 / expected time-to-failure. Therefore, the diagrams have a x-axis referring to expected time-to-failure. The second one presents the recommended (optimal) action implementation time as a function of the prediction event parameter for the various values of the examined input parameter. In the current sensitivity analysis, we examine various cost structures between the action costs and the cost of undesired event as well as all the input parameters with respect to the predictions in terms of the optimal expected loss as well as of the optimal implementation time of the recommended action. All the diagrams were derived from the Python programming language. The functionalities of PANDDA were extracted from the web2py application and was used for the design of the simulation experiments described below.

Figure 10-6 shows the results of sensitivity analysis for the Proactive ELR optimization method. Figure 10-7 shows the results of sensitivity analysis for time-to-failure after the implementation of the action. For this input parameter, the output recommendation is sensitive when the prediction about the expected time-to-failure (for an exponential probability distribution function) is between 0 and 35. When a prediction is referred to a longer time period, they do not affect the output expected loss. Figure 10-8 shows the results of sensitivity analysis for action cost function. Both Figures correspond to the MDP for proactive systems method for one action. Figure 10-9 shows the effect of sensitivity analysis in two alternative actions for the same method and how the recommendation is affected by different action cost functions. Figure 10-10 and Figure 10-11 show the results for the Joint Expected Losses optimization method. The mitigating action corresponds to the maintenance action and the prerequisite action corresponds to the spare parts ordering.

For some parameter values, the associated curves lay outside the optimal implementation time constraint of 68 days, which is the decision horizon (Figure 10-6, Figure 10-9, Figure 10-10). This means that they are not going to be recommended because their implementation would lead to a loss greater that the cost of undesired event and thus, it is more worthy to run the equipment to failure. Our sensitivity analysis and simulation experiments show also that input parameters related to cost are crucial for the recommendation. In other words, proactive decision making is highly sensitive with respect to its input parameters and especially, to the action cost-related input parameters. More specifically, costs have a major effect in all the proactive decision methods (Figure 10-6, Figure 10-10, Figure 10-11), although they do not significantly affect the prerequisite action of the Joint

Expected Losses optimization method (Figure 10-11). In addition, the time that the prediction event is received is important since it also affects significantly the recommendation and the resulting expected loss, while the earlier it is received, the more time allows to the user to prepare the action implementation. Time-to-failure after action implementation affects proactive decision making much less, while changes in delays are slightly influence the resulting recommendations.

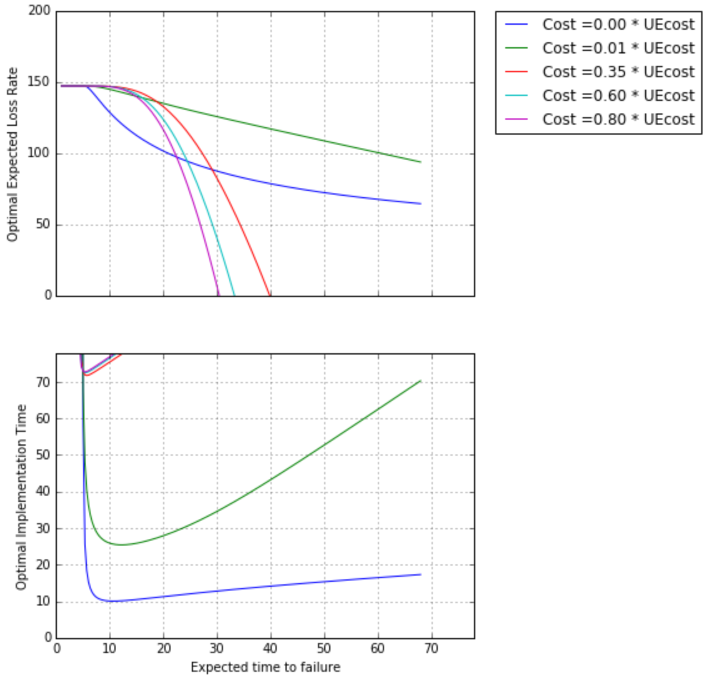

Figure 10-6: Results of sensitivity analysis of action cost function for the "Proactive ELR" method

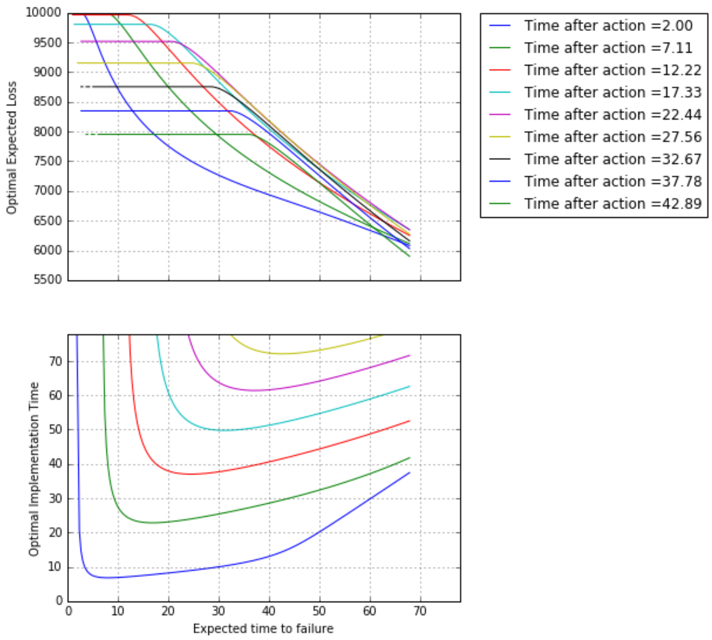

Figure 10-7: Results of sensitivity analysis for time-to-failure for the "Proactive MDP" method.

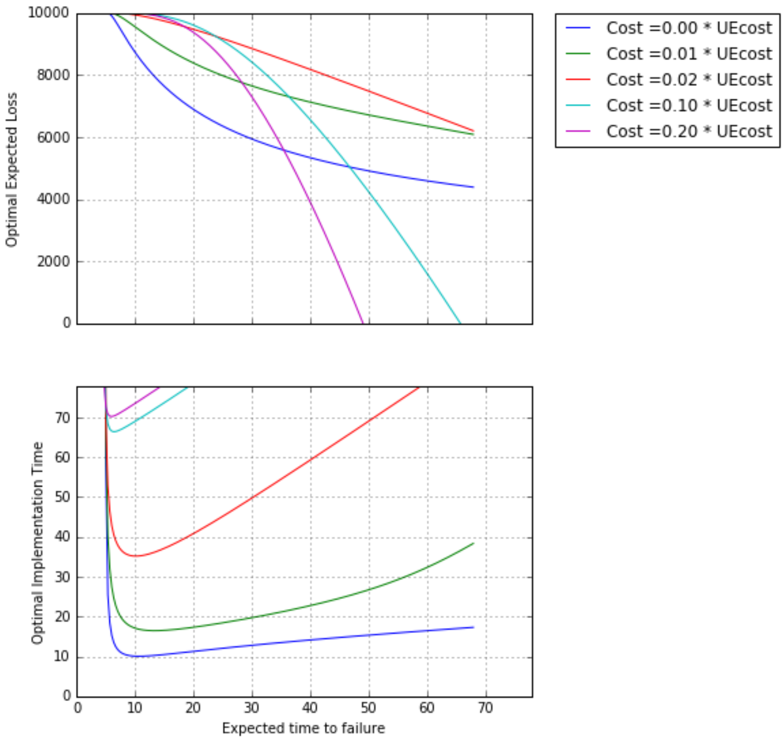

Figure 10-8: Results of sensitivity analysis for action cost function for the "Proactive MDP" method.

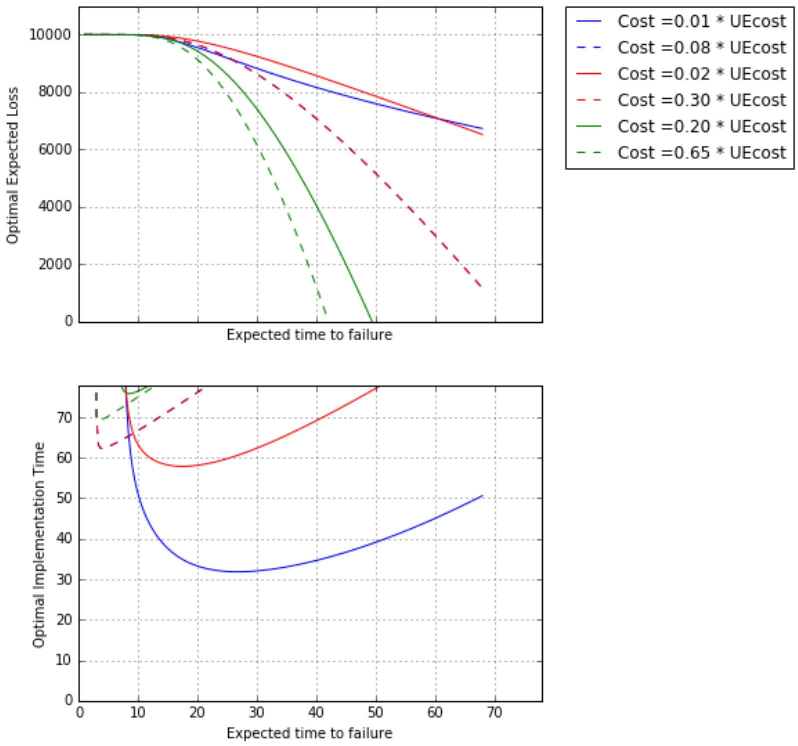

Figure 10-9: Results of sensitivity analysis of action cost function for 2 alternative actions for the "Proactive MDP" method.

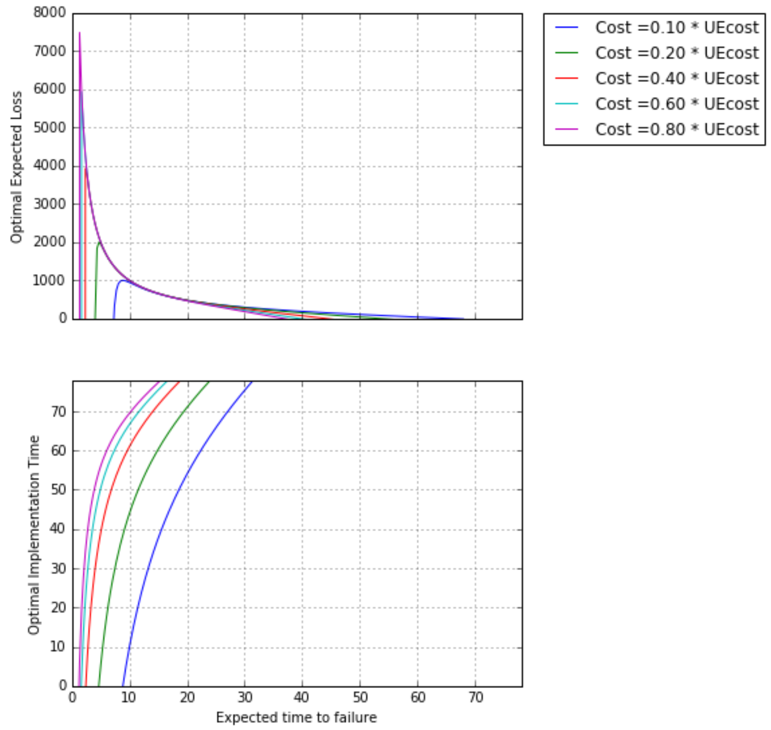

Figure 10-10: Results of sensitivity analysis of mitigating action cost function

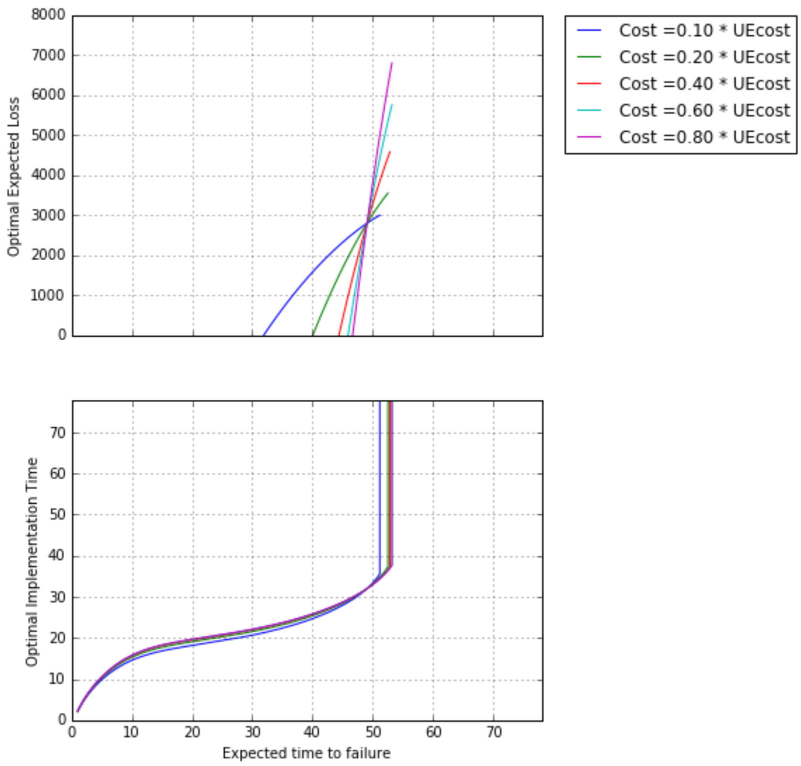

Figure 10-11: Results of sensitivity analysis of prerequisite action cost function

The sensitivity analysis regarding the joint maintenance and logistics decision model was conducted in two ways: (i) with respect to the prediction events; and (ii) with respect to the costs. As far as the (i) way is concerned, in the context of the sensitivity analysis, we simulated several prediction events for investigating the resulting recommendations and the associated expected loss. Table 10-1 shows some indicative results of the sensitivity analysis. It should be not-ed that, since the decision horizon is in 240 hours after the prediction event trigger, the recommended time of 240 means that the action should be performed as has been planned. According to the results, the recommendations can significantly change according to the prediction events. In addition, the earlier a failure is predicted and the proactive decision model is triggered, the less the expected loss is, while the decision maker has more time at their disposal to be prepared and align other manufacturing operations. This conclusion also means that there is a need for reliable and accurate predictive algorithms, with minimized false alarms (false positive and false negative) in order to early predict upcoming undesired events (e.g. equipment failures). In this way, proactive decision models will be able to provide recommendations that lead to a more optimized business performance.

Table 10-1: Results of sensitivity analysis with respect to the prediction events.

Parameter	Maintenance			Spare Parts Ordering		
Predicted time-to-failure	Recom-mended action	Recom-mended time	Resulting Expected Loss	Recom-mended order	Recom-mended time	Resulting Expected Loss
10	a1	2.03	335,434.84	o1	0.00	205,031.98
20	a3	36.87	312,544.27	o3	15.12	199,433.39
50	a3	77.92	234,124.91	o3	39.82	174,861.74
100	a3	104.32	198,063.57	o3	87.87	126,523.99
150	a2	135.22	181,133.28	o2	118,11	120,475.45
200	a4	240.00	169,045.21	o4	189.34	101,366.56
240	a4	240.00	156,217.91	o4	193.21	96,661.94

As far as the (ii) way is concerned, in order to conduct sensitivity analysis of the proactive decision model for joint maintenance and logistics optimization, we simulated four scenarios of cost struc-tures between the action cost and the failure cost as well as between the shortage cost and the spare parts costs given a specific prediction. Figure 10-12 and Figure 10-13 show two indicative plots for the maintenance and logistics expected loss functions respectively (for one maintenance action and one spare parts order), while Table 10-2 and Table 10-3 present the resulting optimal expected loss and the optimal implementation time for the specific action. Similarly to other proactive decision algorithms, this proactive decision model is sensitive to its cost-related input parameters, since the expected loss functions are changed and they can lead to different recommendations.

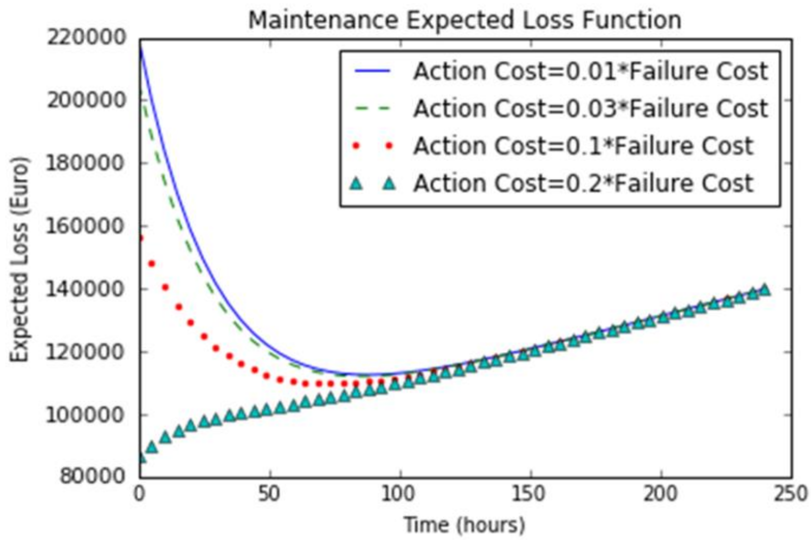

Figure 10-12: Four cost structures for the maintenance expected loss function.

Table 10-2: Results of the cost structures for the maintenance expected loss function.

Action Cost	Action Expected Loss *(Euro)*	Optimal Action Implementation Time *(hours)*
0.01 * Failure Cost	117,211.42	87.78
0.03 * Failure Cost	117,032.12	85.28
0.1 * Failure Cost	115,146.47	73.65
0.2 * Failure Cost	88,654.72	0

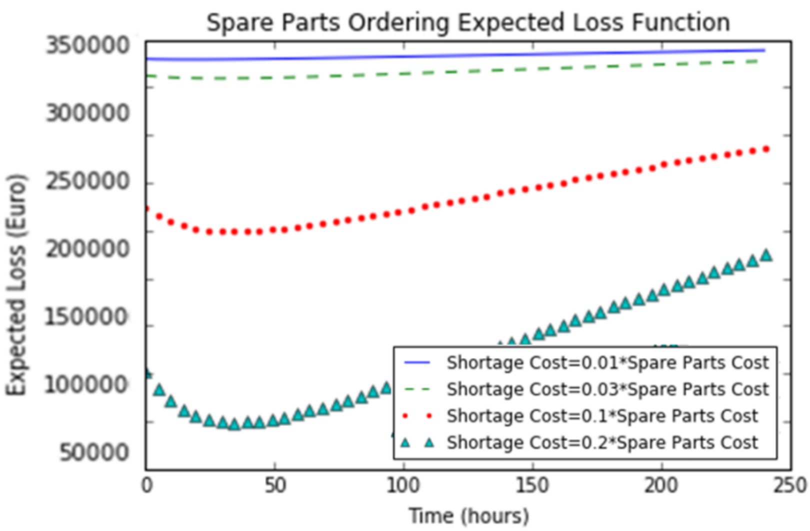

Figure 10-13: Four cost structures for the spare parts ordering expected loss function.

Table 10-3: Results of the cost structures for the spare parts ordering expected loss function.

Shortage Cost	Action Expected Loss *(Euro)*	Optimal Action Implementation Time *(hours)*
0.01 * Spare Parts Cost	345,871.02	18.21
0.03 * Spare Parts Cost	331,124.39	29.06
0.1 * Spare Parts Cost	200,009.99	34.97
0.2 * Spare Parts Cost	50,004.86	35.61

As far as the proactive supplier selection method is concerned, extensive simulation experiments were conducted with various simulated predictions about the spare parts' prices. Moreover, the simulation experiments were conducted for various numbers of available suppliers and for various past portfolios according to simulated historical data. Figure 10-14 shows some indicative examples of these simulation experiments. More specifically, it shows the Markowitz bullet and the efficient frontier when there are 200, 400, and 600 past portfolios of suppliers based on historical data existing in the company's information systems. Moreover, it shows the results when there are 4, 6 and 8 available suppliers. The corresponding portfolios of suppliers for these experiments are shown in Table 10-4.

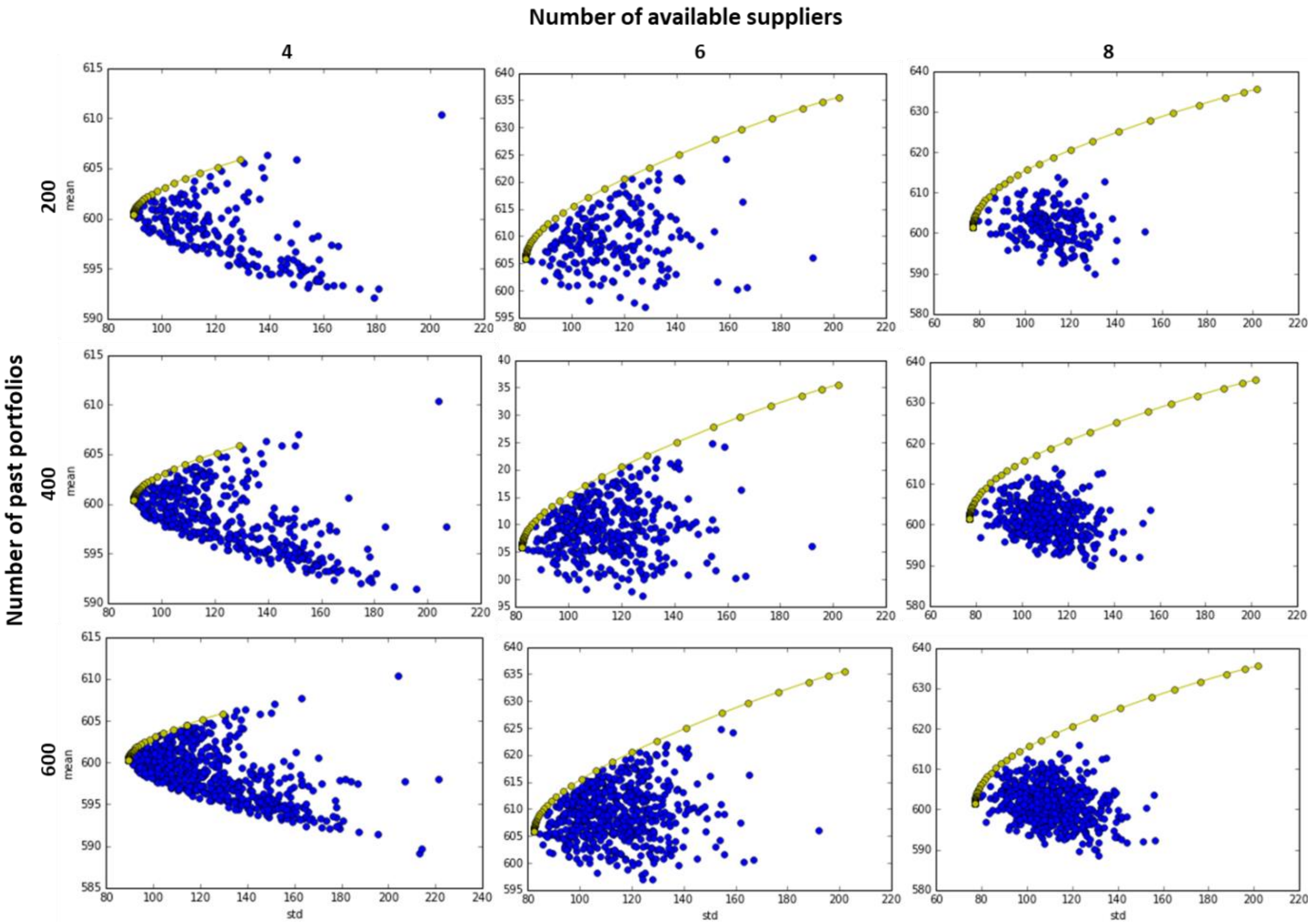

Figure 10-14: The simulation experiments for various numbers of past portfolios and suppliers.

Table 10-4: The resulting portfolios of suppliers based on the experiments

Number of past portfolios		Number of available suppliers: 4	6	8
Number of past portfolios	200	***Supplier A***: 0.14 ***Supplier B***: 0.38 ***Supplier C***: 0.26 ***Supplier D***: 0.22	***Supplier A***: 0.09 ***Supplier B***: 0.29 ***Supplier C***: 0.24 ***Supplier D***: 0.20 ***Supplier E***: 0.12 ***Supplier F***: 0.06	***Supplier A***: 0.10 ***Supplier B***: 0.29 ***Supplier C***: 0.22 ***Supplier D***: 0.18 ***Supplier E***: 0.09 ***Supplier F***: 0.02 ***Supplier G***: 0.07 ***Supplier H***: 0.03
	400	***Supplier A***: 0.16 ***Supplier B***: 0.39 ***Supplier C***: 0.23 ***Supplier D***: 0.22	***Supplier A***: 0.08 ***Supplier B***: 0.33 ***Supplier C***: 0.25 ***Supplier D***: 0.18 ***Supplier E***: 0.10 ***Supplier F***: 0.06	***Supplier A***: 0.08 ***Supplier B***: 0.31 ***Supplier C***: 0.24 ***Supplier D***: 0.20 ***Supplier E***: 0.08 ***Supplier F***: 0.01 ***Supplier G***: 0.06 ***Supplier H***: 0.02
	600	***Supplier A***: 0.17 ***Supplier B***: 0.41 ***Supplier C***: 0.21 ***Supplier D***: 0.21	***Supplier A***: 0.07 ***Supplier B***: 0.36 ***Supplier C***: 0.28 ***Supplier D***: 0.15 ***Supplier E***: 0.09 ***Supplier F***: 0.05	***Supplier A***: 0.07 ***Supplier B***: 0.32 ***Supplier C***: 0.25 ***Supplier D***: 0.21 ***Supplier E***: 0.07 ***Supplier F***: 0.01 ***Supplier G***: 0.05 ***Supplier H***: 0.02

10.3.1.2 Comparative Analysis of Proactive Decision Making

10.3.1.2.1 Proactive Decision Methods for Maintenance Actions

In order to validate the effectiveness the Proactive MDP and the Proactive ELR decision methods, we conducted several executions and we compared the resulting expected losses of the Proactive MDP and the Proactive ELR decision methods with the results of two policies: a “no action” policy due to the lack of predictions and an “immediate action implementation” policy, when there are predictions but not automated decision making. We calculated the average cost and its standard deviation obtained over 100 executions. Each decision method was applied in the context of a different scenario according after interaction with the users in the real industrial environment where we deployed our system. The results are shown in Table 10-5. According to the results, our proposed approach for recommending and applying a proactive mitigating action leads always to a significantly lower loss compared to applying no mitigating action (and thus, applying corrective actions after the equipment breakdown) and to applying an immediate mitigating action (as soon as the user becomes aware of a prediction).

Table 10-5: Average Loss Comparison for 2 scenarios

Scenario	No action	Immediate action	Proactive action
1 Proactive MDP	155000 ± 350 euros	103540 ± 850 euros	96880 ± 70 euros
2 Proactive Expected Loss	750 ± 15 euros	630 ± 10 euros	535 ± 20 euros

10.3.1.2.2 Proactive joint maintenance and logistics decision models

I conducted a comparative analysis for the proactive joint replacement and logistics decision method in the context of the industrial scenario that was validated. We compare the expected losses of this method with those obtained in two scenarios: a reactive scenario of having no prediction (with corrective actions and emergency ordering of spare parts when the failure occurs) and another one where there is a prediction algorithm but not a decision making algorithm. In the first case, corrective maintenance actions last for 5 hours due to the lack of root causes knowledge, while emergency, unplanned ordering of spare parts requires a lead time of 3 hours and a fixed extra cost of 200 euros.

In the second case, due to the failure prediction, either corrective actions are implemented when the failure actually occurs, or immediate preventive actions are applied with a maintenance cost of 325 euros and an inventory cost of 420 euros (due to the lead time of 3 hours and the extra cost), that is a total cost of 945 euros. These values of cost derive from expert knowledge or historical data. However, this deterministic estimation is not realistic due to the stochastic nature of degradation and therefore, the uncertainty at the decision making process. So, a more accurate estimation for this scenario could be obtained if we used the equations of the proposed decision model for t=0, which results in a cost of 1323.8 euros (probabilistic estimation). These results are shown in Table 10-6. In order to further validate our proposed approach, we conducted sensitivity analysis through simulations of prediction events for 5 different manufacturing scenarios. We calculated the average total cost and its standard deviation obtained over 100 executions for the "no prediction", the "only prediction (probabilistic estimation)" and the "proposed approach" policies, as shown in Table 10-7.

The results show that this method can significantly reduce downtime and costs related to maintenance and inventory of spare parts by enabling the transformation of the company from reactive to proactive. More specifically, the "as-is" situation of the company is that it conducts a time-based maintenance, while, if a failure occurs in the interval between two successive time-based maintenances, the appropriate corrective actions are applied, based on breakdown maintenance principles.

Our approach eliminates the probability of an unexpected failure occurring and therefore, it contributes to costs minimization and to the change of company's maintenance management strategy. The company can select either to adopt a Proactive Maintenance strategy (by abolishing the time-based maintenance) or to combine Proactive Maintenance and time-based maintenance principles, e.g. by enlarging the time intervals.

Table 10-6: Results of comparisons.

Approach	Maintenance Loss (Euro)	Inventory Loss (Euro)	Total Loss (Euro)
No prediction	425	620	1045
Only prediction			
Deterministic estimation	325	620	945
Probabilistic estimation	625.5	698.3	1323.8
Proactive approach	348.8	616.6	965.4

I also conducted a comparative analysis for the proactive joint maintenance and logistics decision method in the context of the industrial scenario that was validated. We compared the results of the proposed decision model for the aforementioned scenario with three cases: (i) the case of not having a prediction and therefore, of applying corrective maintenance and inventory-related actions (reactive approach), (ii) the case of having a preventive policy with time-based maintenance and scheduled ordering, and (iii) the case of having prediction but not proactive recommendations and therefore, of applying a preventive action immediately when the prediction is provided (myopic approach).

Table 10-7: Results of extensive comparative analysis.

Scenario	Total Cost (Euro)		
	No prediction	Only prediction	Proposed approach
1	1,286 ± 95	1,494 ± 112	1,015 ± 92
2	823 ± 46	796 ± 44	608 ± 39
3	3,674 ± 115	3,293 ± 124	2,686 ± 122
4	534 ± 32	512 ± 34	371 ± 28
5	50,000 ± 365	48,950 ± 632	28,733 ± 347

Table 10-8: Results of comparative analysis for the joint maintenance and logistics decision model.

Approach	Maintenance Action	Logistics Action	**Total Expected Loss** (maintenance and inventory)
Reactive	Onshore maintenance after oil rig moving	Immediate emergent ordering of DDM	1,492,000 Euro
Preventive	Onshore maintenance after oil rig moving	Scheduled ordering of DDM 48 hours before maintenance	1,021,430 Euro
Myopic	Operate at reduced equipment load when spare part arrives	Immediate ordering of swivel hook	825,000 Euro
Proactive	Offshore maintenance in 85.47 hours	Ordering of gearbox in 42.36 hours	356,850 Euro

In the first case, corrective maintenance actions last more than planned ones due to the lack of root causes knowledge, while emergency, unplanned ordering of spare parts requires a higher lead time along with a cost penalty due to the unplanned distribution. In the second case, there is the cost for time-based maintenance along with the risk of an unexpected failure between time intervals. In the third case, due to the failure prediction, immediate orders of spare parts are applied and preventive maintenance actions are implemented after the required lead time. However, there is the probability of a failure occurring before the spare parts arrived. The cost values for the comparative analysis have been derived from expert knowledge in combination with historical data analysis. The results are shown in Table 10-8.

Moreover, we conducted simulations of prediction events in the context of 5 real case studies, based on the configuration of 5 associated equipment instances by the users in the oil drilling company. For each scenario, we simulated 100 executions by sending prediction events. In all the scenarios, the expected loss of the proposed approach is significantly lower comparing to the reactive, preventive and the myopic approach leading to optimized business performance, as shown in Table 10-9. In the case of myopic policy, actions may be applied at some time according to domain knowledge, something which is not quantifiable and is constrained by the subjectivity of human decision making process.

Table 10-9: Results of comparative analysis for several executions in five scenarios.

	Total Expected Loss for each approach *(Euro)*			
Scenario	**Reactive**	**Preventive**	**Myopic**	**Proactive**
1	1,491,360 ± 185,150	1,019,344 ± 143,229	827,635 ± 93,234	346,355 ± 71,566
2	874,362 ± 41,275	705,627 ± 39,631	596,122 ± 46,988	333,245 ± 37,461
3	122,644 ± 12,476	104,497 ± 9,762	93,532 ± 11,855	50,769 ± 11,450
4	30,550 ± 3,122	24,566 ± 3,099	22,550 ± 3,044	12,915 ± 2,988
5	446,500 ± 23,110	411,433 ± 20,087	315,000 ± 19,750	191,235 ± 16,814

10.3.1.2.3 Proactive Decision Method for Supplier Selection

We compared our approach with two scenarios under several executions: a reactive scenario, having no prediction (with corrective actions and emergency spare parts ordering when the failure occurs) and one where there is a prediction algorithm but not automated decision making. In the first case, corrective maintenance actions last more than predictive ones due to the lack of root causes knowledge, while emergency, unplanned ordering of spare parts requires a longer lead time and leads to a penalty cost due to unplanned distribution. In the second case, due to the failure prediction, either corrective actions are implemented when the failure actually occurs (with the previously referred costs and lead time), or immediate preventive actions are applied, according to a cost-benefit analysis. These results are shown in Table 10-10.

Table 10-10: Results of comparative analysis.

Approach	Maintenance Loss	Inventory Loss	Supplies Loss	Total Loss
No prediction	1,466 ± 58	1,013 ± 27	1,195 ± 34	3,674 ± 119
Only prediction	1,355 ± 112	905 ± 89	1,069 ± 121	3,329 ± 322
Proposed Approach	823 ± 46	708 ± 38	802 ± 44	2,333 ± 128

10.3.2 Continuous Improvement of Proactive Decision Making

We created a computational environment that simulates baseline and action-related cost data from a number of sensors related to cost factors. The computational environment generates and sends to PANDDA cost events derived from simulated sensors, which can have either uniform (i.e. measurement provided at regular intervals) or non-uniform (i.e measurement provided at irregular intervals) sampling. Associated costs are generated according to the normal (Gaussian) distribution based on a mean value and a standard deviation derived from the configured sensor noise, since typical industrial sensor noise is Gaussian (Abramovich et al., 2016). Each sensor is mapped to a specific cost factor as it has been defined during the DMI configuration. The simulation-based evaluation was required due to the long life-times of oil and gas industry’s equipment and the long maintenance intervals.

10.3.2.1 Assessing the impact of user input inaccuracies

In this Section, we present the results of SEF with respect to the action cost functions estimations. More specifically, based on the aforementioned industrial pilot case, we conducted simulation experiments in order to compare a "true" action cost function (i.e. the one guiding the simulation) with the (i) cost function initially configured by the user through the PANDDA GUI ("initial estimate"), (ii) cost function estimated by a partial SEF approach, i.e. by excluding noise filtering and (iii) cost function derived from our full SEF approach. For the "initial estimate", the input of the users was used, while the simulated cost functions were derived from a generator that considers the probabilistic nature of noise on the basis of 100 enactments. The comparison among the methods was done in terms of curve fitting evaluation metrics (Vohnout, 2003). We used the coefficient of determination R^2 (i.e. the proportion of the variance in the dependent variable that is predictable from the independent variable) and the standard error of the estimate (measuring the accuracy of predictions). For each method, we calculated the mean and the standard deviation of each case. The results are presented in Table 10-11 and show that SEF lead to more accurate estimations of action cost functions.

Table 10-11: Comparative Analysis of Action Cost Function Estimation

Cost function	R^2	Standard Error
True	1 ± 0	0 ± 0
Initial estimate	0.783 ± 0.081	1.872 ± 0.115
Based on noisy data	0.911 ± 0.022	1.387 ± 0.108
Based on SEF	0.996 ± 0.003	0.163 ± 0.021

10.3.2.2 Assessing the impact of SEF on user input inaccuracies

In this section, we show the impact of the SEF approach on the generated action-time pair recommendations and the corresponding expected loss. We do so, by comparing the aforementioned outputs of a DMI when considering the cost function derived by user estimations versus by SEF, without considering sensor noise. Table 10-12 shows the comparative results for eight different cases, each one corresponding to the receipt of the prediction event at different times compared to the end of decision epoch, i.e. the decision horizon after which there is no reason of taking a decision about the specific DMI (e.g. next planned maintenance). The results show that SEF has a big impact on the generated proactive recommendations, since user subjectivity is eliminated.

Table 10-12: Comparative Results before and after SEF for One DMI

Parameter	Results Before SEF			Results After SEF		
Remaining Time *(hours)*	Recom-mended action	Recom-mended time *(hours)*	Resulting Ex-pected Loss *(Euro)*	Recom-mended action	Recom-mended time *(hours)*	Resulting Expected Loss *(Euro)*
170	a1	48.43	55,466.18	a1	48.43	55,466.18
180	a1	52.36	57,648.39	a1	61.92	71,822.44
190	a2	63.45	63,766.41	a2	69.27	81,979.28
200	a2	78.47	70,653.91	a1	89.95	92.874.56
210	a2	88.11	76,497.62	a1	103.86	94,837.29
220	a2	104.79	85,447.32	a3	134.15	107,248.66
230	a2	118.85	98,226.65	a3	169.21	123,217.98
240	a3	154.88	114,378.54	a3	178.74	125,191.63

In the current experiment, the user has underestimated the input action cost, something which las led to a significantly lower expected loss with respect to the one derived by applying the SEF approach and therefore, to wrong recommendations about the optimal action and the optimal time of its implementation. The refined by SEF cost function, leads to a more reliable expected loss and therefore to a better recommendation. The benefit of our approach with respect to inaccuracies in user's cost function estimations is multiplied taking into account that modern industries own a large amount of complex equipment, each part of which corresponds to a different DMI. By applying our approach, each one is maintained according to the associated proactive recommendations instead of by conducting time-based full equipment maintenance. To demonstrate the multiplication effect, we followed the same procedure with extensive number of prediction events for 10 DMIs corresponding to the most crucial parts of the oil rig's equipment. The results are presented in an aggregated form in Table 10-13. The results show that the industry underestimates the expected maintenance losses, something which causes obstacles to an efficient overall operational planning and business performance. In other words, without SEF, resources that have been allocated in other operations will need to be used to maintenance operations in order to cover the difference between the user's expectation and the actual losses. SEF leads to more accurate cost estimations and thus, to more reliable recommendations enabling business performance planning and optimization.

Table 10-13: Aggregated Results of SEF Impact for Ten DMIs

DMI	% changes in recommended action	% changes in recommended time	% changes in expected loss	Average difference in expected loss *(Euro)*
1	55.67 %	63.33 %	63.33 %	1,235.44
2	71.16 %	86.75 %	83.37 %	5,689.85
3	43.15 %	45.73 %	40.45 %	692.57
4	64.33 %	82.78 %	80.92 %	4,327.68
5	77.49 %	81.66 %	81.66 %	8,478.14
6	69.33 %	85.18 %	89.91 %	11,265.49
7	61.11 %	65.33%	65.33 %	852.86
8	36.55 %	41.93 %	46.27 %	502.89
9	80.22 %	92.61 %	92.61 %	22,105.76
10	67.93 %	70.08 %	72.12 %	2,937.16

10.3.2.3 Assessing the impact of SEF on sensor inaccuracies

We have conducted extensive simulations in order to examine the output of SEF for various sensor noise levels, to compare these results among them and to those derived based on user estimated cost function and to discuss their difference with the 'true' action cost function. In addition, we compare the MSE of the derived points of the cost function with and without noise removal. Moreover, we examine the impact of the different action cost functions to the proactive recommendations. Figure 10-15 shows the improvement of MSE when using noise filtering for different cost functions and levels of sensor noise.

In each diagram, the X axis represents the Cost Noise, i.e. the part of cost attributed to the sensor noise. The Cost Noise is directly derived from sensor noise and, at the same time, indicates the sensitivity of proactive decision making with respect to its input cost parameters. The range of Cost Noise has been derived from the sensor accuracy in terms of FSO. The Y axis shows the MSEs of both the noisy cost function and the cost function calculated by our approach, compared to the true one. For each cost function, we also show the value of the ratio derived from the MSE of corrected data to the MSE of noisy data, i.e. the Corrected-to-Noisy Data Ratio (CNDR) for indicative Cost Noise levels. For example, a CNDR of 0.3922 means that the MSE of corrected data is equal to the 39.22% of the MSE of noisy data.

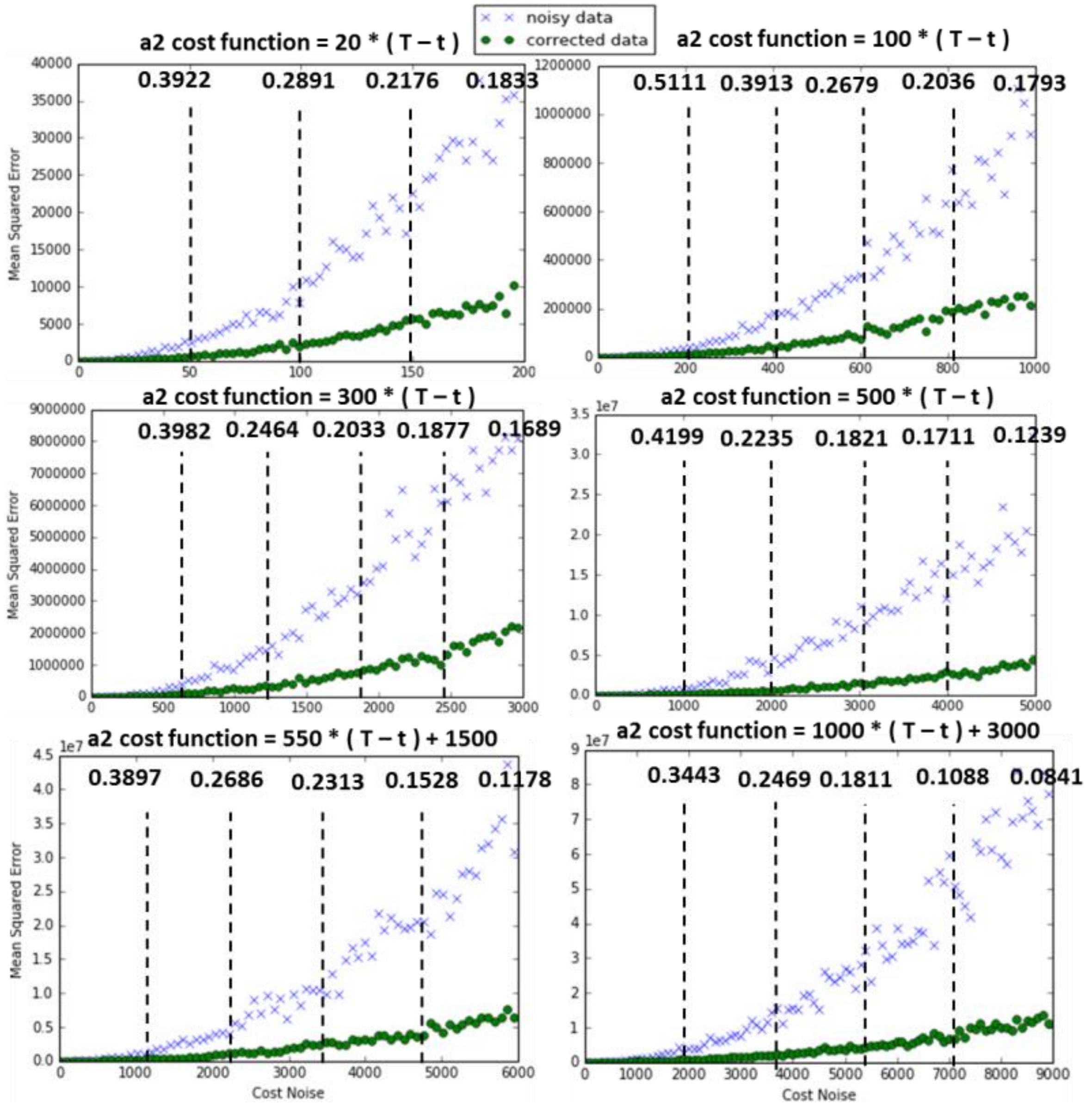

Figure 10-15: Results of simulation experiments showing the impact of noise in SEF.

According to the experimental results shown in Figure 10-15, MSE of the noisy cost function, i.e. of that derived on the basis of noisy measurements is significantly higher than the one of the cost function calculated by our approach, for various cost functions and levels of sensor noise. We show six indicative cost functions in order to investigate the impact of sensor noise to different cost functions. Although the MSEs of the cost functions calculated by our approach increases with the sensor noise, that increase is significantly smaller in relative terms than the one observed to the MSE of the noisy cost functions. In addition, higher Cost Noise levels lead to a lower CNDR, since the MSE of corrected data becomes significantly lower than the MSE of noisy data and thus, the added value of our approach increases.

A large noise level causes less accurate action cost function estimation and therefore, a less reliable recommendation provided by the proactive decision algorithm. In each point of Figure 10-15, for both the noisy and the corrected data, an action-time pair was recommended based on the particular cost function used in the proactive decision method. Table 10-14 shows the percentages of the cases where the recommended action-time pairs changed when the noise was filtered from the data points shown in Figure 10-15. Especially sensitive to sensor noise are the recommended times leading to deviations of several hours regarding the optimal implementation time. Moreover, Table 10-14 shows the average CNDR for each cost function. The results of the right column show that higher cost values lead to more noisy data (since the MSE of the noisy data is significantly higher) and therefore, the added value of noise filtering increases.

Table 10-14: Comparative Results between Noisy and Corrected Data

a2 cost function (Euro)	% changes in recommended action	% changes in recommended time	Average CNDR
20 * (T – t)	12.37%	100%	0.2676
100 * (T – t)	19.18%	100%	0.2312
300 * (T – t)	42.91%	100%	0.2060
500 * (T – t)	64.45%	100%	0.1881
550 * (T – t) + 1500	69.27%	100%	0.1787
1000 * (T – t) + 3000	78.49%	100%	0.1173

10.3.2.4 Evaluation of Proactive Maintenance decisions after SEF

Although proactive decision methods for maintenance lead to less expected losses, they are subjected to high uncertainty due to their dependence on the stochastic degradation process (Van Horenbeek et al., 2013). SEF improves the generated recommendations and provides more reliable results through more accurate cost estimations. Therefore, in this section, we conduct a comparative analysis with the aim to evaluate the results of these decision methods when they incorporate the proposed SEF mechanism. More specifically, we compare the expected losses of a proactive policy after their improvement from SEF with the results of Breakdown Maintenance (BM), Time-Based Maintenance (TBM), CBM with a myopic policy (decision making based on expert knowledge) and with a proactive policy without SEF. We calculated the average expected loss and its standard deviation obtained over 100 executions for 4 scenarios each one of which corresponds to a different part of equipment and decision method. The results are presented in Table 10-15. Based on them, the costs of the Scenarios 1 and 2 have been underestimated, while those of the Scenarios 3 and 4 have been overestimated.

Table 10-15: Comparative Analysis of Maintenance Expected Losses

Maintenance Strategy	Scenario 1 *(Euro)*	Scenario 2 *(Euro)*	Scenario 3 *(Euro)*	Scenario 4 *(Euro)*
BM	155,000 ± 967	1,000,000 ± 7,345	13,955 ± 209	28,565 ± 209
TBM	128,731 ± 565	829,274 ± 7,345	12,142 ± 181	22,882 ± 190
Myopic policy	112,540 ± 850	738,936 ± 7,114	10,127 ± 272	19,931 ± 182
Proactive policy	69,179 ± 341	494,638 ± 5,697	8,938 ± 610	16,798 ± 326
Proactive policy with SEF	78,213 ± 293	522,214 ± 4,815	8,059 ± 591	15,989 ± 202

10.3.3 Context-Awareness in Proactive Decision Making and Action Implementation

For the current experiments, the context-aware model was applied on the basis of the joint maintenance and logistics decision model. We compared the results of our approach for the oil and gas industry scenario with three cases: (i) the case of not having a prediction and therefore, of applying corrective maintenance and inventory-related actions (reactive approach), (ii) the case of having prediction but not proactive recommendations and therefore, of applying a preventive action immediately when the prediction is provided (myopic approach), (iii) the case of having the proactive joint maintenance and logistics decision model without the context-awareness mechanism (proactive approach) and (iv) the case of having the full proposed methodology (context-aware proactive approach). In the first case, corrective maintenance actions last more than planned ones due to the lack of root causes knowledge, while emergency, unplanned ordering of spare parts requires a higher lead time along with a cost penalty due to the unplanned distribution. In the second case, due to the failure prediction, immediate orders of spare parts are applied and preventive maintenance actions are implemented after the required lead time. However, there is the probability of a failure occurring before the spare parts arrived. The comparison with the third case supports the argument that context-awareness can enable business performance optimization. The cost values for the comparative analysis have been derived from expert knowledge in combination with historical data analysis. The results are shown in Table 10-16.

Table 10-16: Results of comparative analysis for the aforementioned scenario

Approach	Maintenance Action	Logistics Action	**Total Expected Loss** (maintenance, inventory and supplies cost)
Reactive	Onshore maintenance after oil rig moving	Immediate ordering of DDM	1,492,000 Euro
Myopic	Gearbox replacement when spare part arrives	Immediate ordering of gearbox	825,000 Euro
Proactive	Operate at reduced equipment load in 95.22 hours	Ordering of swivel hook in 84.23 hours	482,355 Euro
Context-aware proactive	Offshore maintenance in 85.47 hours	Ordering of gearbox or gears in 42.36 hours	376,850 Euro

Moreover, we conducted simulations of prediction events in the context of 5 real case studies, based on the configuration of 5 associated equipment instances by the users in the oil drilling company. For each scenario, we simulated 100 executions by sending prediction events. In all the scenarios, the expected loss of the proposed approach is significantly lower, as shown in Table 10-17.

Table 10-17: Results of comparative analysis for several executions in five scenarios

	Total Expected Loss for each approach *(Euro)*			
Scenario	**Reactive**	**Myopic**	**Proactive**	**Context-aware proactive**
1	1,491,360 ± 185,150	827,635 ± 93,234	482,355 ± 71,566	376,810 ± 53,392
2	874,362 ± 41,275	596,122 ± 46,988	333,245 ±	281,245 ±
3	122,644 ± 12,476	93,532 ± 11,855	50,769 ± 11,450	42,712 ± 8,120
4	30,550 ± 3,122	22,550 ± 3,044	12,915 ± 2,988	9,675 ± 2,336
5	446,500 ± 23,110	315,000 ± 19,750	191,235 ±	122,651 ±

Table 10-18: Results of sensitivity analysis with respect to the context-aware model after reasoning

Parameter	**User-defined Costs**			**Context-aware Costs**		
Predicted time-to-failure *(hours)*	**Recom-mended actions**	**Recommend-ed times** *(hours)*	**Total Ex-pected Loss** *(Euro)*	**Recom-mended actions**	**Recom-mended times** *(hours)*	**Total Ex-pected Loss** *(Euro)*
10	a1, o1	0.00, 2.03	502,493	a1, o1	0.00, 4.01	491,249
20	a1, o1	15.12, 36.87	486,278	a2, o2	15.12, 39.66	379,116
50	a2, o2	39.82, 77.92	414,351	a2, o2	47.12, 91.23	322,735
100	a3, o3	87.87, 104.32	305,742	a2, o2	66.34, 89.91	224,927
150	a4, o4	118,11, 135.22	261,318	a3, o3	107.83, 128.45	181,429
200	a4, o4	189.34, 240.00	245,633	a4, o4	198.88, 240.00	169,658
240	a4, o4	193.21, 240.00	244,773	a4, o4	199.31, 240.00	167,945

Moreover, the results show that even when a prediction exists, the myopic approach does not always result in lower expected losses comparing to the reactive one. So, in this case, actions may be applied at some time according to domain knowledge, something which is not quantifiable and is subjected in the subjectivity of human decision making process. On the other hand, a proactive approach results always in lower expected losses comparing to the reactive and the myopic approach, while the implementation of the proposed approach can lead in an even more optimized business performance in terms of maintenance and spare parts inventory.

In order to conduct sensitivity analysis of the context-aware model, we simulated several prediction events and we compared the resulting recommendations and their associated expected losses between the case of not considering context-awareness (user- defined costs) and the case of considering context-awareness (context-aware costs). The results are shown in Table 10-18.

10.4 Discussion of Evaluation Results

Proactive Maintenance was proved to lead to optimized expected losses in maintenance operations and thus, to an improvement of an overall business performance. The large amounts of real-time data generated by sensors are exploited with the use of event-driven information systems. These information systems incorporate complex algorithms and technologies with the aim to combine these real-time data with historical records and expert knowledge in order to provide menaninful insights about potential problems in a proactive manner. In this way, manufacturing enterprises are able to take advantage of the full potential of big data in an IoT-based industrial environment. Moreover, Proactive Maintenance contributes to the Industry 4.0 concept and RAMI 4.0 with respect to maintenance operations. The holistic view of Proactive Maintenance requires the integration of scalable and efficient event-driven information systems incorporating detection/ giagnostic, prediction and proactive decision making algorithms in the Real-time Processing layer as well as legacy data analytics and FMECA in the Batch Processing layer. Since the least explored area in this information pipeline is “Proactive Decision Making”, the current thesis focused on developing approaches, methods, algorithms and an associated information system to enable data analytics maturity going a step further.

Proactive decision making leads always to a significantly lower loss compared to reactive (breakdown maintenance: corrective actions after a failure occurs) and preventive policies (time-based maintenance: a certain set of actions are apllied in specific time intervals) as well as to myopic policies (according to which a real-time prediction is available, but the type and the time of actions are decided by the expert according to their domain knowledge). The latter often includes immediate implementation of actions as soon as the user becomes aware of the prediction. In addition, the proactive recommendations can significantly change according to the prediction events. The earlier a failure is predicted and the proactive decision model is triggered, the less the expected loss is, while the decision maker has more time at their disposal to be prepared and align other manufacturing operations. The evaluation results show that proactive decision making in the context of Proactive Maintenance leads to lower losses by 29% to 77% with respect to breakdown maintenance policy, by 22% to 65% with respect to time-based maintenance policy and by 7% to 61% with respect to myopic policy. The reason why the range of the latter comparison is so wide is that expert knowledge and estimates vary in each case and is totally subjective. These amounts become even more important for high-revenue, capital-intensive industries.

Although proactive decision methods for maintenance lead to less expected losses, they are subjected to high uncertainty due to their dependence on the stochastic degradation process. Moreover, proactive decision making is highly sensitive with respect to its input parameters and especially to those related to cost. In this sense, input parameters related to cost are crucial for the reliability of proactive recommendations. The SEF mechanism enables the **continuous improvement of proactive decision making** by providing more accurate estimations of proactive decision methods' input parameters and thus, to more reliable recommendations. To do this, it eliminates the inaccuracies derived from the user, from the sensors (due to sensor noise) and from the legacy data systems (due to low data quality). The evaluation results show that the Standard Error of SEF is 91% lower than the expert initial estimate and 88% lower than the incorporation of sensor-driven approaches without noise filtering techniques. Moreover, the proposed SEF mechanism results in more reliable proactive recommendations in terms of both the recommended action and the recommended time, something which leads to a more accurate estimate of the maintenance expected losses by 9% to 88%, depending on the noise level and the cost function that is used as input parameter to the proactive decision method. This fact is also important for the reliability of the comparison of Proactive Maintenance with the other maintenance strategies.

Context-awareness in proactive decision making results in lower expected losses due to the higher accuracy in proactive decision methods' input parameters leading to an even more optimized business performance. However, context-awareness increases the sensitivity of proactive decision making. To this end, the SEF mechanism acquires even higher importance, since not only it leads to more reliable context-aware proactive recommendations, but also it support continuous learning of the context-aware model. In this way, the context-aware model is able to handle uncertainty, while it eliminates the impact of inaccuracies to the reliability of proactive recommendations due to high sensitivity of proactive decision making. The evaluation results show that the proposed approach for context-awareness in proactive decision making leads to differences in expected losses with respect to proactive decision making without context by 12% to 37%, while the context-aware proactive approach still leads to much lower losses comparing to breakdown maintenance policy, time-based maintenance policy and myopic policy. Moreover, context-aware proactive decision making is sensitive to the time window between the time that a prediction is received and the time of the predicted future failure.

11 Lessons Learned and Managerial Implications

The adoption of the Proactive Maintenance strategy and the deployment of the Proactive Maintenance system allows manufacturing companies to gain a strong competitive advantage based on reduced downtimes and optimized performance. The intense employment of IoT technology and cyber-physical systems across the industrial value chain leads to huge amounts of heterogeneous data comprising, for example, product model data from engineering, machine sensor data from manufacturing as well as telemetry data from product usage (Kemper et al., 2013). Extracting business insights and knowledge from these data is one of the major challenges in Industry 4.0 (Golzer et al., 2015; Groger et al., 2016; Groger, 2018). Apart from the IT technical aspects, managerial and organizational aspects have also to be taken into account for a successful implementation. Since sensor-driven information systems for maintenance operations in the context of Industry 4.0 have just started to emerge, managerial implications of their adoption are not widely investigated yet due to the lack or complexity of practical applications.

There is a large debate about whether Industry 4.0 is a revolution or an evolution with contradictory arguments: “The light bulb wasn’t invented by continuously improving the candle. It was about understanding what the job was and looking for solutions” and “Technological innovation is continuous and the concept of a "revolution" in technology innovation is based on a lack of knowledge of the details”. It is actually a clash of two worlds with two different cultures, as shown in Figure 11-1: the “world of business and manufacturing” with a long-term way of thinking meets the “world of IT and data analytics” with a continuous way of thinking. Therefore there is the need for building bridges between these two worlds. To this end, during the very last years, there is some consensus in the sense that Industry 4.0 is considered to be a revolution on a business level and an evolution on a technological level.

In order to reap the rewards of Proactive Maintenance in both the short and the long term, companies will have to create an organisational support structure aligned to the technological solution implemented. Successful implementation can take place and be sustained within organisations that are capable of change, fostering a digital culture and developing and attracting the right capabilities. In PwC's Global Industry 4.0 Survey 2016, respondents said their biggest implementation challenge isn't the right technology, but a lack of digital culture and digital skills in their organisations (PwC, 2016b). Therefore, apart from the right technologies, several people-related factors should be taken into account of predictive maintenance implementation. The most important ones are: a **maintenance implementation strategy,** a **digital culture**, **employee** enablement, **data analytics capabilities** and the **organizational structure**.

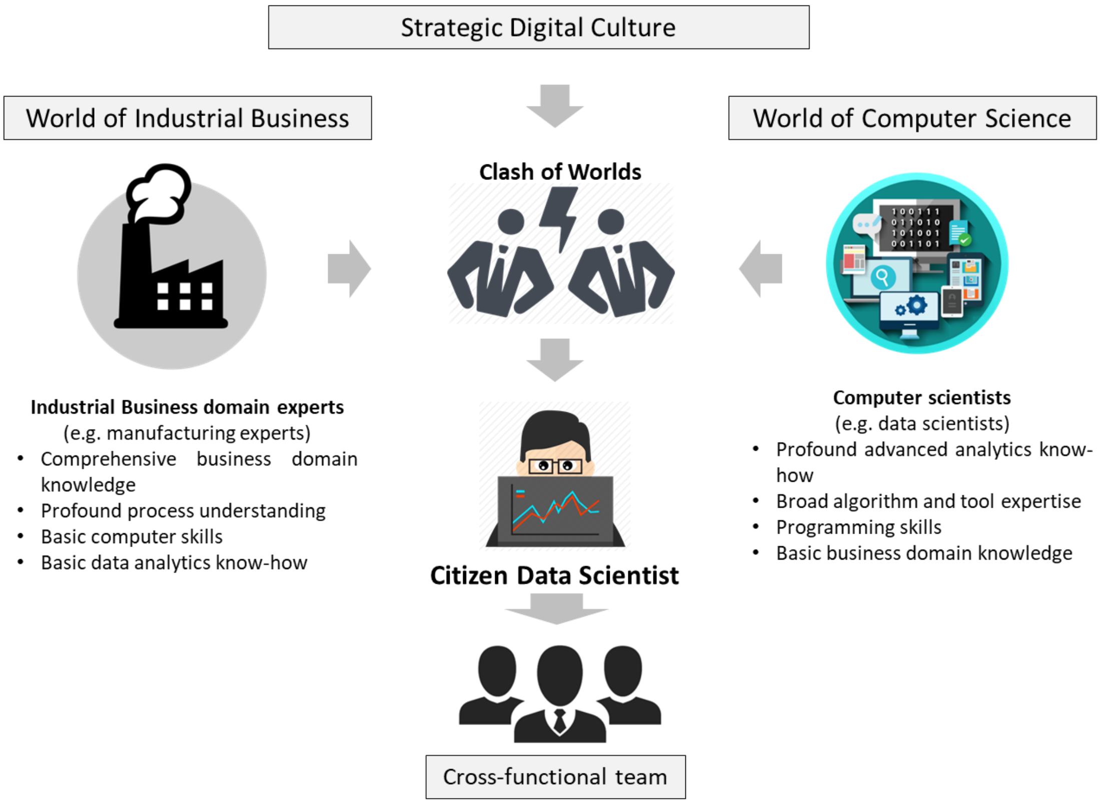

Figure 11-1: Approach for developing a strategic digital culture.

Defining a maintenance implementation strategy

The good functioning of Proactive Maintenance requires the use of data that have to be retrieved in real-time. Therefore, if one or more machines of a production system are unable to provide the required information, because they are technologically obsolete (i.e. the production environment is characterized by "smart" and "non-smart" machines), Proactive Maintenance can be implemented involving "smart" machines and can be further extended as soon as new technologically innovated equipment are introduced. This is a matter of corporate strategy since it deals, on the one hand, with the time and amount of investment for new equipment and, on the other hand, with the adoption of step-by-step logic according to several criteria (e.g. the most critical machines, etc.). For the deployment of modern complex solutions, manufacturing companies may need to stop the production processes and to fully book capacity on machines. This fact was a great challenge and such aspects are some of the key reasons why large enterprises are slowly adopting new technologies.

Similar aspects should be taken into account for the external environment of the manufacturing companies (e.g. suppliers, customers, etc.). To realize the aforementioned strategic goals, a clearly new orientation to change by the organization is needed. Far-reaching change is not always comfortable for the people who make it happen, so change management will also be critical. Thus, a strategic digital culture is of outmost importance. There are different kinds of corporate governments, which can lead subsidiary companies from headquarter (centrally) or leave the subsidiary company to be guided by the local management. Modern manufacturing companies are becoming more and more centrally guided. This fact provides capabilities but also may pose limitations with respect to the adoption of new technologies (e.g. for approving the investment of new sensing equipment).

Building a digital culture

Proactive Maintenance cannot be implemented in complete isolation within the maintenance organisation. It should be embedded into an overall digital manufacturing strategy that is owned and fully supported by top management. This is important not only because the implementation of Proactive Maintenance requires significant resources and capital investments, but also because it needs clear vision and a change management approach from company leaders who understand the power of new digital technologies. Involvement from the boardroom is also needed because the implementation of Proactive Maintenance can have wide-ranging effects within the organisation. It requires cross-functional expert teams with reliability engineers, operators, process technologists, data scientists and IT specialists who together develop new ways of working and communicating. Moreover, Proactive Maintenance is likely to shape new relationships with suppliers and customers with whom data could be exchanged.

These aspects of Proactive Maintenance implementation require a robust a digital culture. This means a culture that stimulates experimentation with new technologies and new ways of working; a culture that stimulates cross-functional cooperation and a culture that is comfortable with data-driven decision-making, even if this goes against human experiences. Such a digital-minded environment can only be cultivated with committed leadership from the top.

Cultivating employee engagement

A major managerial implication has been identified to be employee engagement, i.e. how to empower business domain specialists to get involved with advanced analytics. In the course of several advanced analytics projects, there is a clash of cultures between different groups of employees (Groger, 2018). These projects are typically organized according to the cross-industry standard process for data mining (CRISP-DM) (Han et al., 2012) and require interdisciplinary teams comprising, in particular, business domain specialists and computer scientists. An important goal of designing a Proactive Maintenance governance structure is to create an environment in which IT professionals (e.g. data scientists) and business domain experts (e.g. reliability engineers) can interact and complement each other.

A reliability engineer's insights in how and why assets fail should be paired with, challenged by and harmonised with the insights a data scientist extracts from the data, and vice versa. This type of cross-functional interaction is key to successfully applying data analytics in maintenance and asset management. Business domain specialists, e.g. manufacturing engineers, have comprehensive knowledge about their business domain, its processes and data sources. For instance, they may have detailed know-how on certain machines and manufacturing processes as well as initial ideas for promising data analytics use cases. Yet, they typically have only basic knowledge on data analytics tools and techniques especially regarding advanced analytics. Computer scientists, e.g. data scientists, have a profound know-how on advanced analytics. They typically have a thorough algorithm and tool expertise for the implementation of advanced analytics use cases, but only basic business domain knowledge.

These structural differences between business domain specialists and computer scientists frequently cause inefficiency and ineffectiveness in advanced analytics projects and increase the complexity of collaboration, e.g. having a different educational background, using different terminology (Groger, 2018). Moreover, missing advanced analytics knowledge of business domain specialists prevents data-driven decision making and slows down the development of a data-driven company culture (McAfee and Brynjolfsson, 2012, Groger, 2018).

Building data analytics capabilities

Success with Proactive Maintenance will depend on skills and knowledge. In the aforementioned 2016 Industry 4.0 report (PwC, 2016b), lack of data analytics skills or competencies in the company's workforce is the biggest challenge. Only 27% of survey's respondents currently employ reliability engineers in predictive maintenance, and even fewer (8%) employ data scientists. Consequently, companies' biggest obstacle is their ability to recruit the people needed to put Proactive Maintenance in place. Companies generally understand that it is critical to have in-house data analytics capabilities in order to successfully drive Industry 4.0 applications. Building these capabilities requires not only talented staff but also a holistic and consistent organisation and governance (Groger, 2018).

The system development process revealed a new understanding of the challenge in cooperating across business domains. Adopting key competency within both production process and computer science is critical to succeed in big data analytics. The involved costs need to be considered and approved. Hence, the need of organizational development in the field of business analytics is provided as input to the company's business development strategy. A good first step for companies considering how to best arrange their data analytics could be cross-functional expert teams. Companies may need to introduce new roles like that of data scientist, update existing job profiles to take into account new digital skills, or establish a digital council that oversees the development and further deployment of analytics capabilities throughout the organisation (PwC, 2016b).

Changing the organizational structure

The adoption of the Proactive Maintenance concept and technology requires new skills and competences. People needed for Proactive Maintenance will not want to stay if the company culture does not suit their talents. Hence, the challenge is to empower business domain specialists to get involved with advanced analytics. To tackle this challenge, a new role in the company's organizational structure, the citizen data scientist, is necessary. Citizen data scientists combine business domain knowledge with advanced analytics and computer skills in order to bridge the gap between the world of business and the world of data science (Burton, and Walker, 2015; Morgan, 2015; Davenport, and Harris, 2017; Thompson, and Rogers, 2017; Groger, 2018).

To do this, both technical and organizational aspects should be considered. Technical aspects refer to appropriate tools for citizen data scientists making advanced analytics techniques.Organizational aspects refer to concepts and methodologies to systematically identify and qualify business domain specialists as citizen data scientists as well as to define their organizational integration. Qualification of citizen data scientists particularly requires the development of interdisciplinary educational plans combining knowledge on databases, data engineering and statistics with knowledge on advanced analytics algorithms and suitable tools. Organizational integration comprises the definition of collaboration models between expert data scientists and citizen data scientists especially in large global enterprises.

However, the citizen data scientist role goes beyond just training the business domain expert with data analytics skills. There is the need for a multidisciplinary educational background, combining manufacturing engineering, industrial management and computer science. Strong project management skills are also needed to get Proactive Maintenance "up and running" (PwC, 2017). The citizen data scientist should also have leadership and human resources management skills in order to lead the aforementioned cross-functional teams. In this sense, they should be able to cope with the clash of the two worlds by incorporating the strategic digital culture during the evolution and control of a Proactive Maintenance project. In other words, the citizen data scientist is the link between the top-down strategic decisions and the bottom-up operational data, information, knowledge and experience.

12 Conclusions and Future Work

In this Chapter, the conclusions and the contribution of the thesis are summarized. Moreover, the limitations and the potential extensions of the research are highlighted. Finally, directions for future work are outlined.

12.1 Conclusions

In the current thesis, a new maintenance strategy along with appropriate approaches, methods, models, algorithms and systems were presented in order to answer the research questions posed in Section 3. This maintenance strategy, i.e. Proactive Maintenance, is able to be implemented in the frame of Industry 4.0, in a sensor-driven big data-rich industrial environment. The current thesis also proposes new approaches, methods and models for tackling the challenges arisen in an IoT-based industrial environment along with novel technological solutions.

The current thesis was realized based on the research methodology presented in Section 1. From the literature review of the background concepts: Industry 4.0, Maintenance Management and Proactive Enterprise, the research area and focus was identified. Moreover, the literature review formed the basis for the development of a framework for Proactive Maintenance. A more focused state-of-the-art analysis for each phase of the aforementioned framework revealed the existing research gaps. Specifically, the Decide phase of the framework for Proactive Maintenance is still unexplored research field. To this end, the rest three research questions have to do with this phase and specifically, with proactive decision making, continuous improvement of proactive decision making and context-awareness in proactive decision making. In addition, in the context of the current thesis, an information system incorporating the aforementioned functionalities was implemented, deployed in a real industrial environment and evaluated.

The contribution of the thesis is located to two dimensions: First, it presents a framework for Proactive Maintenance in order to enable the implementation of a new maintenance strategy in a sensor-driven industrial environment. In this way, the thesis answers the first research question. Second, on the basis of the aforementioned framework, the Decide phase is addressed. More specifically, the thesis presented an approach for proactive decision making as well as appropriate proactive event-driven decision methods and models for maintenance and maintenance-driven manufacturing operations in Industry 4.0, which overcome issues of efficiency and scalability of previous approaches. To this end, it proposed two proactive event-driven decision methods for recommendations of maintenance actions, two for recommendations of joint maintenance and logistics actions and one for the recommendation of selection of maintenance spare parts' suppliers. In this way, the second research question was answered.

Then, the thesis presented the SEF approach which constitutes an adaptation mechanism in the sense that it gathers sensor and legacy data before and during actions implementation and processes them in order continuously to improve the generated proactive recommendations. The SEF approach answers the third research question of the thesis.

Finally, the thesis presented a context-awareness mechanism based on a machine learning approach capable of being applied in proactive decision making algorithms, since it can handle uncertainty. The context-awareness mechanism learns from the sensor-generated data during actions implementation and is reasoned through the SEF mechanism. In this way, the fourth research question of the current thesis is answered.

All the aforementioned approaches, methods and algorithms were embedded in an event-driven computational environment in order to ensure efficiency and scalability in a sensor-driven real-time industrial environment according to the Industry 4.0 principles. They take into account the proactive event processing principles and challenges and the most recent advancements in maintenance management as well as the challenges arisen from the increasing use of sensor and communication technologies.

12.2 Limitations and Potential Extensions

Dealing with the research questions posed in the current thesis required the use of a set of assumptions that led to respective limitations. These limitations allowed the research focus on specific methods or techniques as well as on the information system implementation. Potential extensions have to do with these limitations.

The current thesis focused on the Decide phase of the framework for Proactive Maintenance. In this sense, the other phases were not examined in terms of the development of new methods, algorithms and information systems. Since they are well-studied areas, existing methods, algorithms and systems were used for the evaluation of the framework for Proactive Maintenance. In this direction, potential extensions could deal with new algorithms addressing the rest of the phases, capable of being embedded in a real-time, event-driven computational environment and integrated with the proposed system for the Decide phase. Accuracy and reliability of such algorithms will be criteria of outmost importance in this direction.

The current thesis focused on proactive event-driven decision methods dealing with maintenance operations as well as with spare parts ordering and supplier selection driven by maintenance. More manufacturing operations depending on maintenance can be investigated. On this basis one holistic and configurable proactive decision model for maintenance-driven operational decision making could be developed. Finally, in this thesis, the gathering of legacy data from manufacturing companies for deriving the parameter values of the decision methods are not examined.

Moreover, the current thesis focused on continuous improvement of proactive decision making. This is addressed through an adaptation mechanism, the SEF mechanism. The SEF does not consider the challenges existing when integrating a new solution to the manufacturing company's sensors and legacy data systems. Moreover, additional methods and algorithms could be used in order to provide estimation of the proactive decision methods' input parameters. Continuous improvement of proactive decision making can further be investigated based on more business, system and technical requirements for the adoption of a proactive event-driven information system in manufacturing enterprises. The algorithms to be used also depend on the availability of sensors and their relationships with the proactive decision methods' input parameters. To this end, sensor fusion approaches may be required.

Finally, the context-awareness mechanism is incorporated to the Decide phase for enabling context-aware proactive recommendations and is reasoned through the SEF. A generic context-aware model being able to feed into in all the phases of the proactivity principle in order to overcome the challenges due to the difference of such models for the different phases would be one further direction. Finally, further evaluation in real industrial cases would be highly beneficial. Since the manufacturing domain requires long validation periods of new projects, further improvements will potentially be done from the new experience gained.

12.3 Future Work

In this Section, directions for further research based upon the current thesis are proposed. Future work could focus on the following directions:

- Maintenance decision making in the frame of Industry 4.0 can significantly benefit from proactive event-driven computing. Therefore, more proactive event-driven decision methods can be developed in order to tackle with decision making in various manufacturing operations. A more in-depth research on prescriptive analytics, which is considered to be the next business analytics frontier, will significantly contribute to this direction.
- In the current thesis, the proactive decision methods embedded in the information system require the configuration of their parameters by the user (e.g. business analyst). A valuable future direction will be gathering all the data and information necessary from various data sources (e.g. sensor, legacy data systems, etc.) so that the decision models are configured automatically. The use of rules representing the business logic could solve these issues.
- Continuous improvement of proactive decision making can further be investigated based on more business, system and technical requirements for the adoption of a proactive event-driven information system in manufacturing enterprises. To this end, feedback from FMECA analysis as well as legacy data analysis on a batch mode will be researched in order to take into account the most update information.

- A generic context-aware model being able to feed into in all the phases of the proactivity principle would be an interesting further direction. Moreover, additional approaches and algorithms can be researched for handling context-awareness under uncertainty.
- Finally, a generic mechanism for considering the context affecting the decisions in the form of rules and constraints will be useful. This mechanism could create constraints in the objective functions under optimization (e.g. maintenance expected loss functions) in order to take into account the company's policies, regulations, demand, etc.).

References

Aberer, K., Hauswirth, M., & Salehi, A. (2007, May). Infrastructure for data processing in large-scale interconnected sensor networks. In *Mobile Data Management, 2007 International Conference on* (pp. 198-205). IEEE.

Aboelmaged, M. G. S. (2015). E-maintenance research: a multifaceted perspective. *Journal of Manufacturing Technology Management*, *26*(5), 606-631.

Abramovich, Y. I., Besson, O., & Johnson, B. A. (2016). Conditional expected likelihood technique for compound Gaussian and Gaussian distributed noise mixtures. *IEEE Transactions on Signal Processing*, *64*(24), 6640-6649.

Adams, R. P., & MacKay, D. J. (2007). Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*.

Adams, R. P., & MacKay, D. J. (2007). Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*.

Adolphs, P., Bedenbender, H., Dirzus, D., Ehlich, M., Epple, U., Hankel, M., Heidel, R., Hoffmeister, M., Huhle, H., Kärcher, B., & Koziolek, H. (2015). Status Report-Reference Architecture Model Industrie 4.0 (RAMI4. 0). *VDI-Verein Deutscher Ingenieure eV and ZVEI-German Electrical and Electronic Manufacturers Association, Tech. Rep*.

Ahmad, R., & Kamaruddin, S. (2012). An overview of time-based and condition-based maintenance in industrial application. *Computers & Industrial Engineering*, *63*(1), 135-149.

Alaswad, S., & Xiang, Y. (2017). A review on condition-based maintenance optimization models for stochastically deteriorating system. *Reliability Engineering & System Safety*, *157*, 54-63.

Ali, J., & Ushaq, M. (2009). A consistent and robust Kalman filter design for in-motion alignment of inertial navigation system. *Measurement*, *42*(4), 577-582.

Al-Najjar, B., & Jacobsson, M. (2013). A computerised model to enhance the cost-effectiveness of production and maintenance dynamic decisions: A case study at Fiat. *Journal of Quality in Maintenance Engineering*, *19*(2), 114-127.

Amorim-Melo, P., Shehab, E., Kirkwood, L., & Baguley, P. (2014). Cost drivers of integrated maintenance in high-value systems. *Procedia CIRP*, *22*, 152-156.

Animah, I., & Shafiee, M. (2017). Condition assessment, remaining useful life prediction and life extension decision making for offshore oil and gas assets. *Journal of Loss Prevention in the Process Industries*.

Armstrong, M. J., & Atkins, D. R. (1996). Joint optimization of maintenance and inventory policies for a simple system. *IIE transactions*, *28*(5), 415-424.

Aronis, K. P., Magou, I., Dekker, R., & Tagaras, G. (2004). Inventory control of spare parts using a Bayesian approach: A case study. *European Journal of Operational Research*, *154*(3), 730-739.

Artikis, A., Baber, C., Bizarro, P., Canudas-de-Wit, C., Etzion, O., Fournier, F., Goulart, P., Howes, A., Lygeros, J., Paliouras, G., & Schuster, A. (2014). Scalable proactive event-driven decision making. *IEEE Technology and Society Magazine*, *33*(3), 35-41.

Artikis, A., Etzion, O., Feldman, Z., & Fournier, F. (2012, July). Event processing under uncertainty. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems* (pp. 32-43). ACM.

Arunraj, N. S., & Maiti, J. (2007). Risk-based maintenance—Techniques and applications. *Journal of hazardous materials*, *142*(3), 653-661.

Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer networks*, *54*(15), 2787-2805.

Azadeh, A., Asadzadeh, S. M., Salehi, N., & Firoozi, M. (2015). Condition-based maintenance effectiveness for series–parallel power generation system—A combined Markovian simulation model. *Reliability Engineering & System Safety*, *142*, 357-368.

Baheti, R., & Gill, H. (2011). Cyber-physical systems. *The impact of control technology*, *12*, 161-166.

Bangemann, T., Rebeuf, X., Reboul, D., Schulze, A., Szymanski, J., Thomesse, J. P., Thron, M., & Zerhouni, N. (2006). PROTEUS—Creating distributed maintenance systems through an integration platform. *Computers in Industry*, *57*(6), 539-551.

Barry, D., & Hartigan, J. A. (1992). Product partition models for change point problems. *The Annals of Statistics*, 260-279.

Basu, A. T. A. N. U. (2013). Five pillars of prescriptive analytics success. *Analytics magazine*, 8-12.

BCM Advanced Research. Industry 4.0 - The future of the Factory. Available online at: http://www.bcmcom.com/solutions_application_industry40.htm. Last accessed: 28/04/2018.

Belvedere, V., Grando, A., & Bielli, P. (2013). A quantitative investigation of the role of information and communication technologies in the implementation of a product-service system. *International Journal of Production Research*, *51*(2), 410-426.

Benedettini, O., Baines, T. S., Lightfoot, H. W., & Greenough, R. M. (2009). State-of-the-art in integrated vehicle health management. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *223*(2), 157-170.

Ben-Gal, I. (2007). Bayesian networks. *Encyclopedia of statistics in quality and reliability*.

Bernus, P., Noran, O., & Molina, A. (2015). Enterprise architecture: Twenty years of the GERAM framework. *Annual Reviews in Control*, *39*, 83-93.

Besnard, F., & Bertling, L. (2010). An approach for condition-based maintenance optimization applied to wind turbine blades. *IEEE Transactions on Sustainable Energy*, *1*(2), 77-83.

Bi, Z., Da Xu, L., & Wang, C. (2014). Internet of things for enterprise systems of modern manufacturing. *IEEE Transactions on industrial informatics*, *10*(2), 1537-1546.

Blanchard, B. S., Verma, D., & Peterson, E. L. (1995). *Maintainability: a key to effective serviceability and maintenance management* (Vol. 13). John Wiley & Sons.

Bohlin, M., & Wärja, M. (2015). Maintenance optimization with duration-dependent costs. *Annals of Operations Research*, *224*(1), 1-23.

Bumblauskas, D., Gemmill, D., Igou, A., & Anzengruber, J. (2017). Smart Maintenance Decision Support Systems (SMDSS) based on corporate big data analytics. *Expert Systems with Applications*, *90*, 303-317.

Cabinet Office. (2015). Report on The 5th Science and Technology Basic Plan. *Cabinet Office of Japan*.

Camarinha-Matos, L. M., Goes, J., Gomes, L., & Martins, J. (2013, April). Contributing to the Internet of Things. In *Doctoral Conference on Computing, Electrical and Industrial Systems*(pp. 3-12). Springer, Berlin, Heidelberg.

Campos, J. (2009). Development in the application of ICT in condition monitoring and maintenance. *Computers in Industry*, *60*(1), 1-20.

Candell, O., Karim, R., & Söderholm, P. (2009). eMaintenance—Information logistics for maintenance support. *Robotics and Computer-Integrated Manufacturing*, *25*(6), 937-944.

Cannata, A., Karnouskos, S., & Taisch, M. (2010). Dynamic e-maintenance in the era of SOA-ready device dominated industrial environments. In *Engineering Asset Lifecycle Management* (pp. 411-419). Springer, London.

Carlander, L., Kirkwood, L., Shehab, E., Baguley, P., & Durazo-Cardenas, I. (2016). Integration of cost-risk assessment of denial of service within an intelligent maintenance system. *Procedia CIRP*, *47*, 66-71.

Carnero, M. (2006). An evaluation system of the setting up of predictive maintenance programmes. *Reliability Engineering & System Safety*, *91*(8), 945-963.

Castro, I., Huynh, K. T., Barros, A., & Bérenguer, C. (2012, June). A predictive maintenance strategy based on mean residual life for systems subject to competing failures due to degradation and shocks. In *11th International Probabilistic Safety Assessment and Management Conference & the Annual European Safety and Reliability Conference-PSAM 11/ESREL 2012* (pp. 375-384). Curran Associates, Inc.

Chemweno, P., Pintelon, L., Muchiri, P. N., & Van Horenbeek, A. (2018). Risk assessment methodologies in maintenance decision making: A review of dependability modelling approaches. *Reliability Engineering & System Safety*, *173*, 64-77.

Chen, P. C., & Kezunovic, M. (2016). Fuzzy Logic Approach to Predictive Risk Analysis in Distribution Outage Management. *IEEE Transactions on Smart Grid*, *7*(6), 2827-2836.

Cheng, F. T., Tieng, H., Yang, H. C., Hung, M. H., Lin, Y. C., Wei, C. F., & Shieh, Z. Y. (2016). Industry 4.1 for wheel machining automation. *IEEE Robotics and Automation Letters*, *1*(1), 332-339.

Cheng, J., Greiner, R., Kelly, J., Bell, D., & Liu, W. (2002). Learning Bayesian networks from data: an information-theory based approach. *Artificial intelligence*, *137*(1-2), 43-90.

Cheng, T., Pandey, M. D., & van der Weide, J. A. (2012). The probability distribution of maintenance cost of a system affected by the gamma process of degradation: Finite time solution. *Reliability Engineering & System Safety*, *108*, 65-76.

Cinus, M., Confalonieri, M., Barni, A., & Valente, A. (2016). An ANN Based Decision Support System Fostering Production Plan Optimization Through Preventive Maintenance Management. In *Advances in Neural Networks* (pp. 447-455). Springer, Cham.

Compton, M., Barnaghi, P., Bermudez, L., GarcíA-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., & Huang, V. (2012). The SSN ontology of the W3C semantic sensor network incubator group. *Web semantics: science, services and agents on the World Wide Web*, *17*, 25-32.

Conseil national de l’industrie. (2013). The New Face of Industry in France. *French National Industry Council*.

Cortes, H., Daaboul, J., Le Duigou, J., & Eynard, B. (2016). Strategic Lean Management: integration of operational performance indicators for strategic Lean management. *IFAC-PapersOnLine*, *49*(12), 65-70.

da Rosa, J. H., Barbosa, J. L., & Ribeiro, G. D. (2016). ORACON: An adaptive model for context prediction. *Expert Systems with Applications*, *45*, 56-70.

Da Xu, L. (2011). Enterprise systems: state-of-the-art and future trends. *IEEE Transactions on Industrial Informatics*, *7*(4), 630-640.

Da Xu, L., He, W., & Li, S. (2014). Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, *10*(4), 2233-2243.

Dandotiya, R., & Lundberg, J. (2012). Economic model for maintenance decision: a case study for mill liners. *Journal of Quality in Maintenance Engineering*, *18*(1), 79-97.

Davenport, T., & Harris, J. (2017). *Competing on Analytics: Updated, with a New Introduction: The New Science of Winning*. Harvard Business Press.

de Almeida, A. T., Cavalcante, C. A. V., Alencar, M. H., Ferreira, R. J. P., de Almeida-Filho, A. T., & Garcez, T. V. (2015). Spare Parts Planning Decisions. In *Multicriteria and Multiobjective Models for Risk, Reliability and Maintenance Decision Analysis* (pp. 273-296). Springer International Publishing.

de Jonge, B., Dijkstra, A. S., & Romeijnders, W. (2015). Cost benefits of postponing time-based maintenance under lifetime distribution uncertainty. *Reliability Engineering & System Safety*, *140*, 15-21.

de Jonge, B., Teunter, R., & Tinga, T. (2017). The influence of practical factors on the benefits of condition-based maintenance over time-based maintenance. *Reliability engineering & system safety*, *158*, 21-30.

De Prado, A. G., Ortiz, G., & Boubeta-Puig, J. (2017). CARED-SOA: A context-aware event-driven service-oriented Architecture. *IEEE Access*, *5*, 4646-4663.

Delen, D., & Demirkan, H. (2013). Data, information and analytics as services.

den Hertog, D., & Postek, K. (2016). *Bridging the gap between predictive and prescriptive analytics-new optimization methodology needed*. Technical report, Tilburg University, Netherlands, 2016. Available at: http://www. optimization-online. org/DB_HTML/2016/12/5779. html.

Dhillon, B. S. (2017). *Engineering Systems Reliability, Safety, and Maintenance: An Integrated Approach*. CRC Press.

Diamond, S., & Boyd, S. (2016). CVXPY: A Python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research*, *17*(1), 2909-2913.

Ding, S. H., & Kamaruddin, S. (2015). Maintenance policy optimization—literature review and directions. *The International Journal of Advanced Manufacturing Technology*, *76*(5-8), 1263-1283.

Djurdjanovic, D., Lee, J., & Ni, J. (2003). Watchdog Agent—an infotronics-based prognostics approach for product performance degradation assessment and prediction. *Advanced Engineering Informatics*, *17*(3-4), 109-125.

Do, P., Voisin, A., Levrat, E., & Iung, B. (2015). A proactive condition-based maintenance strategy with both perfect and imperfect maintenance actions. *Reliability Engineering & System Safety*, *133*, 22-32.

Do, P., Voisin, A., Levrat, E., & Iung, B. (2015). A proactive condition-based maintenance strategy with both perfect and imperfect maintenance actions. *Reliability Engineering & System Safety*, *133*, 22-32.

Drath, R., & Horch, A. (2014). Industrie 4.0: Hit or hype?[industry forum]. *IEEE industrial electronics magazine*, *8*(2), 56-58.

Drumheller, Z., Smits, K. M., Illangasekare, T. H., Regnery, J., Lee, J., & Kitanidis, P. K. (2017). Optimal decision making algorithm for managed aquifer recharge and recovery operation using near real-time data: Benchtop scale laboratory demonstration. *Groundwater Monitoring & Remediation*, *37*(1), 27-41.

Duarte, J. C., Cunha, P. F., & Craveiro, J. T. (2013). Maintenance database. *Procedia CIRP*, *7*, 551-556.

Duffuaa, S. O., Ben-Daya, M., Al-Sultan, K. S., & Andijani, A. A. (2001). A generic conceptual simulation model for maintenance systems. *Journal of Quality in Maintenance Engineering*, *7*(3), 207-219.

Duffuaa, S. O., Raouf, A., & Campbell, J. D. (1999). Planning and control of maintenance systems. *John Willey and Son, New York*.

Dunsdon, J., & Harrington, M. (2008, March). The application of open system architecture for condition based maintenance to complete IVHM. In *Aerospace Conference, 2008 IEEE* (pp. 1-9). IEEE.

Eldar, Y. C., Ben-Tal, A., & Nemirovski, A. (2005). Robust mean-squared error estimation in the presence of model uncertainties. *IEEE Transactions on Signal Processing*, *53*(1), 168-181.

Eldar, Y. C., Ben-Tal, A., & Nemirovski, A. (2005). Robust mean-squared error estimation in the presence of model uncertainties. *IEEE Transactions on Signal Processing*, *53*(1), 168-181.

Elwany, A. H., & Gebraeel, N. Z. (2008). Sensor-driven prognostic models for equipment replacement and spare parts inventory. *IIE Transactions*, *40*(7), 629-639.

Elwany, A. H., & Gebraeel, N. Z. (2008). Sensor-driven prognostic models for equipment replacement and spare parts inventory. *IIE Transactions*, *40*(7), 629-639.

Emde, S., & Boysen, N. (2012). Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines. *International Journal of Production Economics*, *135*(1), 393-402.

Engel, Y., & Etzion, O. (2011, July). Towards proactive event-driven computing. In *Proceedings of the 5th ACM international conference on Distributed event-based system* (pp. 125-136). ACM.

Engel, Y., & Etzion, O. (2011, July). Towards proactive event-driven computing. In *Proceedings of the 5th ACM international conference on Distributed event-based system* (pp. 125-136). ACM.

Engel, Y., Etzion, O., & Feldman, Z. (2012, July). A basic model for proactive event-driven computing. In *Proceedings of the 6th ACM international conference on distributed event-based systems* (pp. 107-118). ACM.

Engel, Y., Etzion, O., & Feldman, Z. (2012, July). A basic model for proactive event-driven computing. In *Proceedings of the 6th ACM international conference on distributed event-based systems* (pp. 107-118). ACM.

Espíndola, D. B., Fumagalli, L., Garetti, M., Pereira, C. E., Botelho, S. S., & Henriques, R. V. (2013). A model-based approach for data integration to improve maintenance management by mixed reality. *Computers in Industry*, *64*(4), 376-391.

Etzion, O., Niblett, P., & Luckham, D. C. (2011). *Event processing in action*. Greenwich: Manning.

European Commission. (2016). Factories of the Future PPP: Towards Competitive EU Manufacturing. *Bruxelles: European Commission*.

Evans, P. C., & Annunziata, M. (2012). Industrial internet: Pushing the boundaries. *General Electric Reports*.

Faccio, M., Persona, A., Sgarbossa, F., & Zanin, G. (2014). Industrial maintenance policy development: A quantitative framework. *International Journal of Production Economics*, *147*, 85-93.

Feldman, Z., Fournier, F., Franklin, R., & Metzger, A. (2013, June). Proactive event processing in action: a case study on the proactive management of transport processes (industry article). In *Proceedings of the 7th ACM international conference on Distributed event-based systems* (pp. 97-106). ACM.

Feldman, Z., Fournier, F., Franklin, R., & Metzger, A. (2013, June). Proactive event processing in action: a case study on the proactive management of transport processes (industry article). In *Proceedings of the 7th ACM international conference on Distributed event-based systems* (pp. 97-106). ACM.

Feng, Y. H., Teng, T. H., & Tan, A. H. (2009). Modelling situation awareness for Context-aware Decision Support. *Expert Systems with Applications*, *36*(1), 455-463.

Fennel, H., Bunzel, S., Heinecke, H., Bielefeld, J., Fürst, S., Schnelle, K. P., Grote, W., Maldener, N., Weber, T., Wohlgemuth, F., & Ruh, J. (2006). *Achievements and exploitation of the AUTOSAR development partnership* (No. 2006-21-0019). SAE Technical Paper.

Fielding, R. T., & Taylor, R. N. (2000). *Architectural styles and the design of network-based software architectures* (Vol. 7). Doctoral dissertation: University of California, Irvine.

Fitouri, C., Fnaiech, N., Varnier, C., Fnaiech, F., & Zerhouni, N. (2016). A Decison-Making Approach for Job Shop Scheduling with Job Depending Degradation and Predictive Maintenance. *IFAC-PapersOnLine*, *49*(12), 1490-1495.

Foresight. (2013). The Future of Manufacturing: A New Era of Opportunity and Challenge for the UK. *UK Government Office for Science*.

Fouladirad, M., & Grall, A. (2014). On-line change detection and condition-based maintenance for systems with unknown deterioration parameters. *IMA Journal of Management Mathematics*, *25*(2), 139-158.

Fouladirad, M., & Grall, A. (2015). Monitoring and condition-based maintenance with abrupt change in a system's deterioration rate. *International Journal of Systems Science*, *46*(12), 2183-2194.

Fouladirad, M., Grall, A., & Dieulle, L. (2008). On the use of on-line detection for maintenance of gradually deteriorating systems. *Reliability Engineering & System Safety*, *93*(12), 1814-1820.

Fournier, F., Kofman, A., Skarbovsky, I., & Skarlatidis, A. (2015, March). Extending Event-Driven Architecture for Proactive Systems. In *EDBT/ICDT Workshops* (pp. 104-110).

Fu, S., & Xu, C. Z. (2010). Quantifying event correlations for proactive failure management in networked computing systems. *Journal of Parallel and Distributed Computing*, *70*(11), 1100-1109.

Fumagalli, L., & Macchi, M. (2015). Integrating maintenance within the production process through a flexible E-maintenance platform. *IFAC-PapersOnLine*, *48*(3), 1457-1462.

Galar, D., Thaduri, A., Catelani, M., & Ciani, L. (2015). Context awareness for maintenance decision making: A diagnosis and prognosis approach. *Measurement*, *67*, 137-150.

Gan, S., Zhang, Z., Zhou, Y., & Shi, J. (2015). Joint optimization of maintenance, buffer, and spare parts for a production system. *Applied Mathematical Modelling*, *39*(19), 6032-6042.

Garcia, M. C., Sanz-Bobi, M. A., & del Pico, J. (2006). SIMAP: Intelligent System for Predictive Maintenance: Application to the health condition monitoring of a windturbine gearbox. *Computers in Industry*, *57*(6), 552-568.

Garg, A., & Deshmukh, S. G. (2006). Maintenance management: literature review and directions. *Journal of quality in maintenance engineering*, *12*(3), 205-238.

Gartner (2014) Gartner's 2014 hype cycle for emerging technologies maps the journey to digital business. Available online: http://www.gartner.com/newsroom/id/2819918. Last accessed: 27/04/2018.

Gartner. (2016). 2017 Planning Guide for Data and Analytics. Available online: https://www.gartner.com/binaries/content/assets/events/keywords/catalyst/catus8/2017_planning_guide_for_data_analytics.pdf. Last accessed: 27/04/2018.

Gegenfurtner, K. R. (1992). PRAXIS: Brent's algorithm for function minimization. *Behavior Research Methods, Instruments, & Computers*, *24*(4), 560-564.

Gegenfurtner, K. R. (1992). PRAXIS: Brent's algorithm for function minimization. *Behavior Research Methods, Instruments, & Computers*, *24*(4), 560-564.

Ghosh, S., Ghosh, D., & Mohanta, D. K. (2017). Situational Awareness Enhancement of Smart Grids Using Intelligent Maintenance Scheduling of Phasor Measurement Sensors. *IEEE Sensors Journal*, *17*(23), 7685-7693.

Godarzi, A. A., Amiri, R. M., Talaei, A., & Jamasb, T. (2014). Predicting oil price movements: A dynamic Artificial Neural Network approach. *Energy Policy*, *68*, 371-382.

Gölzer, P., Cato, P., & Amberg, M. (2015, May). Data Processing Requirements of Industry 4.0-Use Cases for Big Data Applications. In *ECIS*.

Gopalakrishnan, M., Bokrantz, J., Ylipää, T., & Skoogh, A. (2015). Planning of maintenance activities–A current state mapping in industry. *Procedia CIRP*, *30*, 480-485.

Gröger, C. (2018). Building an Industry 4.0 Analytics Platform. *Datenbank-Spektrum*, 1-10.

Gröger, C., Kassner, L., Hoos, E., Königsberger, J., Kiefer, C., Silcher, S., & Mitschang, B. (2016, April). The Data-driven Factory. In *Proceedings of the 18th International Conference on Enterprise Information Systems* (pp. 40-52). SCITEPRESS-Science and Technology Publications, Lda.

Gu, C., He, Y., Han, X., & Chen, Z. (2017, July). Product quality oriented predictive maintenance strategy for manufacturing systems. In *Prognostics and System Health Management Conference (PHM-Harbin), 2017* (pp. 1-7). IEEE.

Guillén, A. J., Crespo, A., Gómez, J. F., & Sanz, M. D. (2016). A framework for effective management of condition based maintenance programs in the context of industrial development of E-maintenance strategies. *Computers in Industry*, *82*, 170-185.

Guillén, A. J., Crespo, A., Gómez, J. F., & Sanz, M. D. (2016). A framework for effective management of condition based maintenance programs in the context of industrial development of E-maintenance strategies. *Computers in Industry*, *82*, 170-185.

Gulledge, T., Hiroshige, S., & Iyer, R. (2010). Condition-based maintenance and the product improvement process. *Computers in Industry*, *61*(9), 813-832.

Guo, W., Jin, J. J., & Hu, S. J. (2013). Allocation of maintenance resources in mixed model assembly systems. *Journal of Manufacturing Systems*, *32*(3), 473-479.

Gupta, A., & Lawsirirat, C. (2006). Strategically optimum maintenance of monitoring-enabled multi-component systems using continuous-time jump deterioration models. *Journal of Quality in Maintenance Engineering*, *12*(3), 306-329.

Hamby, D. M. (1994). A review of techniques for parameter sensitivity analysis of environmental models. *Environmental monitoring and assessment*, *32*(2), 135-154.

Han, J., Pei, J., & Kamber, M. (2012). *Data mining: concepts and techniques*. Elsevier.

Han, T., & Yang, B. S. (2006). Development of an e-maintenance system integrating advanced techniques. *Computers in Industry*, *57*(6), 569-580.

Han, Y., & Song, Y. H. (2003). Condition monitoring techniques for electrical equipment-a literature survey. *IEEE Transactions on Power delivery, 18*(1), 4-13.

Handte, M., Schiele, G., Matjuntke, V., Becker, C., & Marrón, P. J. (2012). 3PC: System support for adaptive peer-to-peer pervasive computing. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, *7*(1), 10.

Hankel, M., & Rexroth, B. (2015). The reference architectural model industrie 4.0 (rami 4.0). *ZVEI, April*.

Haroun, A. E. (2015). Maintenance cost estimation: application of activity-based costing as a fair estimate method. *Journal of Quality in Maintenance Engineering*, *21*(3), 258-270.

Hashemian, H. M., & Bean, W. C. (2011). State-of-the-art predictive maintenance techniques. *IEEE Transactions on Instrumentation and measurement*, *60*(10), 3480-3492.

Hausladen, I., & Bechheim, C. (2004, June). E-maintenance platform as a basis for business process integration. In *Industrial Informatics, 2004. INDIN'04. 2004 2nd IEEE International Conference on* (pp. 46-51). IEEE.

Hazen, B. T., Boone, C. A., Ezell, J. D., & Jones-Farmer, L. A. (2014). Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications. *International Journal of Production Economics*, *154*, 72-80.

He, S., Cheng, B., Huang, Y., Duan, L., & Chen, J. (2017, June). Proactive Personalized Services in Large-Scale IoT-Based Healthcare Application. In *Web Services (ICWS), 2017 IEEE International Conference on* (pp. 808-813). IEEE.

He, Y., Gu, C., Chen, Z., & Han, X. (2017). Integrated predictive maintenance strategy for manufacturing systems by combining quality control and mission reliability analysis. *International Journal of Production Research*, *55*(19), 5841-5862.

Heckerman, D. (1998). A tutorial on learning with Bayesian networks. In *Learning in graphical models* (pp. 301-354). Springer, Dordrecht.

Hocheol, J., Taehwan, K., & Joongmin, C. (2010). Proactive Task Execution Using Data Sharing and Event Transition among Personal Devices. *KSII Transactions on Internet & Information Systems*, *4*(6).

Hong, H. P., Zhou, W., Zhang, S., & Ye, W. (2014). Optimal condition-based maintenance decisions for systems with dependent stochastic degradation of components. *Reliability Engineering & System Safety*, *121*, 276-288.

Hulett, D. (2016). *Integrated cost-schedule risk analysis*. Routledge.

Huynh, K. T., Barros, A., & Bérenguer, C. (2015). Multi-Level Decision-Making for The Predictive Maintenance of k-Out-of-n Deteriorating Systems. *IEEE transactions on Reliability*, *64*(1), 94-117.

Ibrahim, J. G., Chen, M. H., & Sinha, D. (2005). *Bayesian survival analysis*. John Wiley & Sons, Ltd.

Irigaray, A., Gilabert, E., Jantunen, E., & Adgar, A. (2009). Ubiquitous computing for dynamic condition-based maintenance. *Journal of Quality in Maintenance Engineering*, *15*(2), 151-166.

Islam, H. M., & Khan, M. A. (2010). Bayesian analysis of system availability with geometric failure law in life testing. *Journal of Quality in Maintenance Engineering*, *16*(2), 214-221.

ISO 13372. (2012a). Condition monitoring and diagnostics of machines—Vocabulary.

ISO 13379-1. (2012b). Condition monitoring and diagnosis of machines—Data interpretation and diagnosiss techniques—Part 1: General guidelines.

ISO 17359. (2011). Condition monitoring and diagnosis of machines- General guidelines.

Iung, B. (2003). From remote maintenance to MAS-based e-maintenance of an industrial process. *Journal of Intelligent Manufacturing*, *14*(1), 59-82.

Iung, B., Levrat, E., Marquez, A. C., & Erbe, H. (2009). Conceptual framework for e-Maintenance: Illustration by e-Maintenance technologies and platforms. *Annual Reviews in Control*, *33*(2), 220-229.

Jafari, L., & Makis, V. (2015). Joint optimal lot sizing and preventive maintenance policy for a production facility subject to condition monitoring. *International Journal of Production Economics*, *169*, 156-168.

Jardine, A. K., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, *20*(7), 1483-1510.

Jarke, M. (2014). Interview with Michael Feindt on "Prescriptive big data analytics". *Business & Information Systems Engineering*, *6*(5), 301-302.

Jiang, X., Duan, F., Tian, H., & Wei, X. (2015). Optimization of reliability centered predictive maintenance scheme for inertial navigation system. *Reliability Engineering & System Safety*, *140*, 208-217.

Jiang, Y., Chen, M., & Zhou, D. (2015). Joint optimization of preventive maintenance and inventory policies for multi-unit systems subject to deteriorating spare part inventory. *Journal of Manufacturing Systems*, *35*, 191-205.

Julier, S. J., & Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, *92*(3), 401-422.

Julier, S. J., & Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, *92*(3), 401-422.

Julka, N., Thirunavukkarasu, A., Lendermann, P., Gan, B. P., Schirrmann, A., Fromm, H., & Wong, E. (2011). Making use of prognostics health management information for aerospace spare components logistics network optimisation. *Computers in Industry*, *62*(6), 613-622.

Jwo, D. J., & Cho, T. S. (2007). A practical note on evaluating Kalman filter performance optimality and degradation. *Applied Mathematics and Computation*, *193*(2), 482-505.

Kagermann, H., Helbig, J., Hellinger, A., & Wahlster, W. (2013). *Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry; final report of the Industrie 4.0 Working Group*. Forschungsunion.

Kajko-Mattsson, M., Karim, R., & Mirijamdotter, A. (2011). Essential components of emaintenance. *International Journal of Pedagogy, Innovation and New Technologies*, *7*(6), 555-571.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, *82*(1), 35-45.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, *82*(1), 35-45.

Kang, H. S., Lee, J. Y., Choi, S., Kim, H., Park, J. H., Son, J. Y., Kim, H., & Do Noh, S. (2016). Smart manufacturing: Past research, present findings, and future directions. *International Journal of Precision Engineering and Manufacturing-Green Technology*, *3*(1), 111-128.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., & Wu, A. Y. (2002). An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence*, *24*(7), 881-892.

Kapur, K. C., & Pecht, M. (2014). *Reliability engineering*. John Wiley & Sons.

Kapur, K. C., & Pecht, M. (2014). *Reliability engineering*. John Wiley & Sons.

Karim, R., Candell, O., & Söderholm, P. (2009). E-maintenance and information logistics: aspects of content format. *Journal of Quality in Maintenance Engineering*, *15*(3), 308-324.

Keizer, M. C. O., Teunter, R. H., & Veldman, J. (2016). Clustering condition-based maintenance for systems with redundancy and economic dependencies. *European Journal of Operational Research*, *251*(2), 531-540.

Kemper, H. G., Baars, H., & Lasi, H. (2013). An integrated business intelligence framework. In *Business Intelligence and Performance Management* (pp. 13-26). Springer, London.

Khaitan, S. K., & McCalley, J. D. (2015). Design techniques and applications of cyber-physical systems: A survey. *IEEE Systems Journal*, *9*(2), 350-365.

Korb, K. B., & Nicholson, A. E. (2010). *Bayesian artificial intelligence*. CRC press.

Kouedeu, A. F., Kenné, J. P., Dejax, P., Songmene, V., & Polotski, V. (2015). Production and maintenance planning for a failure-prone deteriorating manufacturing system: a hierarchical control approach. *The International Journal of Advanced Manufacturing Technology*, *76*(9-12), 1607-1619.

Kriegel, H. P., Schubert, E., & Zimek, A. (2017). The (black) art of runtime evaluation: Are we comparing algorithms or implementations?. *Knowledge and Information Systems*, *52*(2), 341-378.

Kroll, B., Schaffranek, D., Schriegel, S., & Niggemann, O. (2014, September). System modeling based on machine learning for anomaly detection and predictive maintenance in industrial plants. In *Emerging Technology and Factory Automation (ETFA), 2014 IEEE* (pp. 1-7). IEEE.

Kruchten, P. (2004). *The rational unified process: an introduction*. Addison-Wesley Professional.

Krumeich, J., Werth, D., & Loos, P. (2016). Prescriptive control of business processes. *Business & Information Systems Engineering*, *58*(4), 261-280.

Krumeich, J., Werth, D., & Loos, P. (2016). Prescriptive control of business processes. *Business & Information Systems Engineering*, *58*(4), 261-280.

Kumar, U., & Galar, D. (2018). Maintenance in the Era of Industry 4.0: Issues and Challenges. In *Quality, IT and Business Operations* (pp. 231-250). Springer, Singapore.

Kwon, O., Lee, N., & Shin, B. (2014). Data quality management, data usage experience and acquisition intention of big data analytics. *International Journal of Information Management*, *34*(3), 387-394.

Labib, A. W. (2004). A decision analysis model for maintenance policy selection using a CMMS. *Journal of Quality in Maintenance Engineering*, *10*(3), 191-202.

Le, M. D., & Tan, C. M. (2013). Optimal maintenance strategy of deteriorating system under imperfect maintenance and inspection using mixed inspection scheduling. *Reliability Engineering & System Safety*, *113*, 21-29.

Le, T., Luo, M., Zhou, J., & Chan, H. L. (2014, September). Predictive maintenance decision using statistical linear regression and kernel methods. In *Emerging Technology and Factory Automation (ETFA), 2014 IEEE* (pp. 1-6). IEEE.

Lebold, M., & Thurston, M. (2001, May). Open standards for condition-based maintenance and prognostic systems. In *Maintenance and Reliability Conference (MARCON)* (Vol. 200). May.

Lee, J., Abujamra, R., Jardine, A. K., Lin, D., & Banjevic, D. (2004). An integrated platform for diagnostics, prognostics and maintenance optimization. *Proceedings of the intelligent maintenance systems*, 15-27.

Lee, J., Bagheri, B., & Kao, H. A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, *3*, 18-23.

Lee, J., Lapira, E., Bagheri, B., & Kao, H. A. (2013). Recent advances and trends in predictive manufacturing systems in big data environment. *Manufacturing Letters*, *1*(1), 38-41.

Lee, J., Ni, J., Djurdjanovic, D., Qiu, H., & Liao, H. (2006). Intelligent prognostics tools and e-maintenance. *Computers in industry*, *57*(6), 476-489.

Lee, S., & Ni, J. (2013). Joint decision making for maintenance and production scheduling of production systems. *The International Journal of Advanced Manufacturing Technology*, *66*(5-8), 1135-1146.

Léger, J. B., & Morel, G. (2001). Integration of maintenance in the enterprise: towards an enterprise modelling-based framework compliant with proactive maintenance strategy. *Production planning & control*, *12*(2), 176-187.

Lehmann, E. L., & Casella, G. (2006). *Theory of point estimation*. Springer Science & Business Media.

Levrat, E., & Iung, B. (2007). TELMA: A full e-maintenance platform. *Proceedings of WCEAM/CM 2007*, 11-14.

Levrat, E., Iung, B., & Crespo Marquez, A. (2008). E-maintenance: review and conceptual framework. *Production Planning & Control*, *19*(4), 408-429.

Li, D., & Gao, J. (2010). Study and application of Reliability-centered Maintenance considering Radical Maintenance. *Journal of Loss Prevention in the Process Industries*, *23*(5), 622-629.

Li, H., Deloux, E., & Dieulle, L. (2016). A condition-based maintenance policy for multi-component systems with Lévy copulas dependence. *Reliability Engineering & System Safety*, *149*, 44-55.

Li, K. (2015). Made in China 2025. *Beijing: State Council of China*.

Li, S., & Lin, B. (2006). Accessing information sharing and information quality in supply chain management. *Decision support systems*, *42*(3), 1641-1656.

Li, S., Da Xu, L., & Zhao, S. (2015). The internet of things: a survey. *Information Systems Frontiers*, *17*(2), 243-259.

Li, Y., Tang, Q., Chang, Q., & Brundage, M. P. (2017). An event-based analysis of condition-based maintenance decision-making in multistage production systems. *International Journal of Production Research*, *55*(16), 4753-4764.

Liao, Y., Deschamps, F., Loures, E. D. F. R., & Ramos, L. F. P. (2017). Past, present and future of Industry 4.0-a systematic literature review and research agenda proposal. *International Journal of Production Research*, *55*(12), 3609-3629.

Lin, J., Pulido, J., & Asplund, M. (2015). Reliability analysis for preventive maintenance based on classical and Bayesian semi-parametric degradation approaches using locomotive wheel-sets as a case study. *Reliability Engineering & System Safety*, *134*, 143-156.

Lindström, J., Larsson, H., Jonsson, M., & Lejon, E. (2017). Towards intelligent and sustainable production: combining and integrating online predictive maintenance and continuous quality control. *Procedia CIRP*, *63*, 443-448.

Liu, H., & Gluch, D. P. (2004). Conceptual modeling with the object-process methodology in software architecture. *Journal of Computing Sciences in Colleges*, *19*(3), 10-21.

Liu, M., Gao, Y., Li, G., Guang, X., & Li, S. (2016). An improved alignment method for the Strapdown Inertial Navigation System (SINS). *Sensors*, *16*(5), 621.

Liu, X., Li, J., Al-Khalifa, K. N., Hamouda, A. S., Coit, D. W., & Elsayed, E. A. (2013). Condition-based maintenance for continuously monitored degrading systems with multiple failure modes. *IIE Transactions*, *45*(4), 422-435.

Löfsten, H. (1999). Management of industrial maintenance–economic evaluation of maintenance policies. *International Journal of Operations & Production Management*, *19*(7), 716-737.

Lorén, S., & de Maré, J. (2015). Maintenance for reliability—a case study. *Annals of Operations Research*, *224*(1), 111-119.

Louie, A. H. (2010). Robert Rosen's anticipatory systems. *Foresight*, *12*(3), 18-29.

Lourakis, M. I. (2005). A brief description of the Levenberg-Marquardt algorithm implemented by levmar. *Foundation of Research and Technology*, *4*(1), 1-6.

Luckham, D. (2002). *The power of events* (Vol. 204). Reading: Addison-Wesley.

Luckham, D. C. (2012). *Event processing for business: organizing the real-time enterprise*. John Wiley & Sons.

Lundberg, A. (2006). Leverage complex event processing to improve operational performance. *Business Intelligence Journal*, *11*(1), 55.

Macchi, M., Crespo Márquez, A., Holgado, M., Fumagalli, L., & Barberá Martínez, L. (2014). Value-driven engineering of E-maintenance platforms. *Journal of Manufacturing Technology Management*, *25*(4), 568-598.

Macchi, M., Martínez, L. B., Márquez, A. C., Fumagalli, L., & Granados, M. H. (2018). Value Assessment of e-Maintenance Platforms. In *Advanced Maintenance Modelling for Asset Management* (pp. 371-385). Springer, Cham.

Mahmood, Z. (2018). *Connected Environments for the Internet of Things: Challenges and Solutions*. Springer Publishing Company, Incorporated.

Maleki, S., Bingham, C., & Zhang, Y. (2016). Development and realization of change-point analysis for the detection of emerging faults on industrial systems. *IEEE Transactions on Industrial Informatics*, *12*(3), 1180-1187.

Markowitz, H. (1952). Portfolio selection. *The journal of finance*, *7*(1), 77-91.

Márquez, F. P. G., Tobias, A. M., Pérez, J. M. P., & Papaelias, M. (2012). Condition monitoring of wind turbines: Techniques and methods. *Renewable Energy*, *46*, 169-178.

Martorell, S., Villamizar, M., Carlos, S., & Sánchez, A. (2010). Maintenance modeling and optimization integrating human and material resources. *Reliability Engineering & System Safety*, *95*(12), 1293-1299.

Mayrhofer, R. (2005). Context prediction based on context histories: Expected benefits, issues and current state-of-the-art. *Cognitive science research paper-University of Sussex CSRP*, *577*, 31.

McAfee, A., Brynjolfsson, E., Davenport, T. H., Patil, D. J., & Barton, D. (2012). Big data: the management revolution. *Harvard business review*, *90*(10), 60-68.

Mendes, A. A., & Ribeiro, J. L. D. (2014). Establishment of a maintenance plan based on quantitative analysis in the context of RCM in a JIT production scenario. *Reliability Engineering & System Safety*, *127*, 21-29.

Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad hoc networks*, *10*(7), 1497-1516.

Moghaddass, R., & Zuo, M. J. (2014). An integrated framework for online diagnostic and prognostic health monitoring using a multistate deterioration process. *Reliability Engineering & System Safety*, *124*, 92-104.

Moré, J. J. (1978). The Levenberg-Marquardt algorithm: implementation and theory. In *Numerical analysis* (pp. 105-116). Springer, Berlin, Heidelberg.

Morgan, L. (2015). Citizen data scientists: 7 ways to harness talent. Available online at: https://www.informationweek.com/big-data/big-data-analytics/citizen-data-scientists-7-ways-to-harness-talent/d/d-id/1321389. *Last accessed: 28/04/2018.*

Muller, A., Marquez, A. C., & Iung, B. (2008b). On the concept of e-maintenance: Review and current research. *Reliability Engineering & System Safety*, *93*(8), 1165-1187.

Muller, A., Suhner, M. C., & Iung, B. (2007). Maintenance alternative integration to prognosis process engineering. *Journal of Quality in Maintenance Engineering*, *13*(2), 198-211.

Muller, A., Suhner, M. C., & Iung, B. (2008). Formalisation of a new prognosis model for supporting proactive maintenance implementation on industrial system. *Reliability Engineering & System Safety*, *93*(2), 234-253.

Muller, A., Suhner, M. C., & Iung, B. (2008a). Formalisation of a new prognosis model for supporting proactive maintenance implementation on industrial system. *Reliability Engineering & System Safety*, *93*(2), 234-253.

Nadin, M. (Ed.). (2016). *Anticipation across disciplines*. Springer.

Nadj, M., Jegadeesan, H., Maedche, A., Hoffmann, D., & Erdmann, P. (2016, June). A Situation Awareness Driven Design for predictive Maintenance Systems: the Case of Oil and gas Pipeline Operations. In *ECIS*.

National Research Foundation. (2016). Research, Innovation and Enterprise (RIE) 2015 Plan. *Singapore: Prime Minister's Office of Singapore*.

Nguyen, K. A., Do, P., & Grall, A. (2015). Multi-level predictive maintenance for multi-component systems. *Reliability Engineering & System Safety*, *144*, 83-94.

Nguyen, K. A., Do, P., & Grall, A. (2017). Joint predictive maintenance and inventory strategy for multi-component systems using Birnbaum's structural importance. *Reliability Engineering & System Safety*, *168*, 249-261.

Njike, A. N., Pellerin, R., & Kenne, J. P. (2012). Simultaneous control of maintenance and production rates of a manufacturing system with defective products. *Journal of Intelligent Manufacturing*, *23*(2), 323-332.

Nodem, F. I. D., Kenne, J. P., & Gharbi, A. (2011). Production planning and repair/replacement switching policy for deteriorating manufacturing systems. *The International Journal of Advanced Manufacturing Technology*, *57*(5-8), 827.

Nordgård, D. E., & Sand, K. (2010). Application of Bayesian networks for risk analysis of MV air insulated switch operation. *Reliability Engineering & System Safety*, *95*(12), 1358-1366.

Ollila, A., & Malmipuro, M. (1999). Maintenance has a role in quality. *The TQM magazine*, *11*(1), 17-21.

OSIsoft. Increased Rig performance through holistic information view. Available online at: https://cdn.osisoft.com/corp/en/media/presentations/2013/EMEA2013/PDF/

EMEA13_AkerSolutions_Bjerke_IncreasedRigperformanceThroughHolisticInformationView.pdf. Last accessed 28/04/2018.

Papazoglou, M. (2008). *Web services: principles and technology*. Pearson Education.

Park, C., Moon, D., Do, N., & Bae, S. M. (2016). A predictive maintenance approach based on real-time internal parameter monitoring. *The International Journal of Advanced Manufacturing Technology*, *85*(1-4), 623-632.

Paruolo, P., Saisana, M., & Saltelli, A. (2013). Ratings and rankings: voodoo or science?. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *176*(3), 609-634.

Payne, T. (2010). Offshore operations and maintenance: a growing market. *Petroleum economist*, *77*(1).

Pelleg, D., & Moore, A. W. (2000, June). X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML* (Vol. 1, pp. 727-734).

Pelliccione, P., Knauss, E., Heldal, R., Ågren, S. M., Mallozzi, P., Alminger, A., & Borgentun, D. (2017). Automotive architecture framework: the experience of volvo cars. *Journal of Systems Architecture*, *77*, 83-100.

Peng, Y., Dong, M., & Zuo, M. J. (2010). Current status of machine prognostics in condition-based maintenance: a review. *The International Journal of Advanced Manufacturing Technology*, *50*(1-4), 297-313.

Peng, Y., Dong, M., & Zuo, M. J. (2010). Current status of machine prognostics in condition-based maintenance: a review. *The International Journal of Advanced Manufacturing Technology*, *50*(1-4), 297-313.

Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Context aware computing for the internet of things: A survey. *IEEE communications surveys & tutorials*, *16*(1), 414-454.

Petersen, S. A., van der Kooij, R., & Puhar, P. (2016, October). Connecting Business Processes and Sensor Data in Proactive Manufacturing Enterprises. In *Working Conference on Virtual Enterprises* (pp. 101-109). Springer, Cham.

Pinjala, S. K., Pintelon, L., & Vereecke, A. (2006). An empirical investigation on the relationship between business and maintenance strategies. *International journal of production economics*, *104*(1), 214-229.

Pistofidis, P., Emmanouilidis, C., Koulamas, C., Karampatzakis, D., & Papathanassiou, N. (2012, March). A layered e-maintenance architecture powered by smart wireless monitoring components. In *Industrial Technology (ICIT), 2012 IEEE International Conference on* (pp. 390-395). IEEE.

Platform Industrie 4.0. (2015). Platform Industrie 4.0: *Umsetzunsstrategie Industrie 4.0. Berlin*.

Portioli-Staudacher, A., & Tantardini, M. (2012). Integrated maintenance and production planning: a model to include rescheduling costs. *Journal of Quality in Maintenance Engineering*, *18*(1), 42-59.

Potocnik, M., & Juric, M. B. (2014). Towards complex event aware services as part of SOA. *IEEE Transactions on Services Computing*, *7*(3), 486-500.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

PwC. (2016a). Proactive Organization. Available online at: http://www.chairdigitaleconomy.com.au/wp-content/uploads/2016/03/RPT Proac-tiveOrganisation FIN.pdf. Last accessed: 28/04/2018.

PwC. (2016b). Industry 4.0: Building the digital enterprise. Available online at: https://www.pwc.com/gx/en/industries/industries-4.0/landing-page/industry-4.0-building-your-digital-enterprise-april-2016.pdf. Last accessed: 28/04/2018.

PwC. (2017a). Predictive Maintenance 4.0: Predict the unpredictable. Available online at: https://www.pwc.nl/en/publicaties/predictive-maintenance-40-predict-the-unpredictable.html . Last accessed: 28/04/2018.

PwC. (2017b). Industry 4.0: Global Digital Operations Study 2018. Available online at: http://www.pwc.com/gx/en/industries/industry-4.0.html. Last accessed: 28/04/2018.

Rafael, R., Shirley, A. J., & Liveris, A. (2014). Report To The President Accelerating US Advanced Manufacturing. *Report, Washington DC, US*.

Rajan, N. M., & Rajalakshmy, P. (2014). Estimation of Sensor Temperature Drift using Kalman Filter. *International Journal of Engineering Research and Technology, 3*(2). ESRSA Publications.

Rausch, M., & Liao, H. (2010). Joint production and spare part inventory control strategy driven by condition based maintenance. *IEEE Transactions on Reliability*, *59*(3), 507-516.

Ridgway, K., Clegg, C. W., & Williams, D. J. (2013). The Factory of the Future. Future of Manufacturing Project: Evidence Paper 29. *Foresight, Government Office for Science, Sheffield*.

Romero, D., & Vernadat, F. (2016). Future perspectives on next generation enterprise information systems. *Computers in Industry*, (79), 1-2.

Rosen, R. (2012). Anticipatory systems. In *Anticipatory systems* (pp. 313-370). Springer, New York, NY.

RTInsights. (2016). Proactive Computing: Changing the future. Available online at: https://www.rtinsights.com/proactive-computing-prescriptive-analytics-special-report/. Last accessed: 28/04/2018.

Ruschel, E., Santos, E. A. P., & Loures, E. D. F. R. (2017). Industrial maintenance decision-making: A systematic literature review. *Journal of Manufacturing Systems*, *45*, 180-194.

Ruschel, E., Santos, E. A. P., & Loures, E. D. F. R. (2017). Industrial maintenance decision-making: A systematic literature review. *Journal of Manufacturing Systems*, *45*, 180-194.

Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,.

Rüßmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P., & Harnisch, M. (2015). Industry 4.0: The future of productivity and growth in manufacturing industries. *Boston Consulting Group*, *9*.

Said, A. B., Shahzad, M. K., Zamai, E., Hubac, S., & Tollenaere, M. (2016). Towards proactive maintenance actions scheduling in the Semiconductor Industry (SI) using Bayesian approach. *IFAC-PapersOnLine*, *49*(12), 544-549.

Salovaara, A., & Oulasvirta, A. (2004, October). Six modes of proactive resource management: a user-centric typology for proactive behaviors. In *Proceedings of the third Nordic conference on Human-computer interaction* (pp. 57-60). ACM.

Sankavaram, C., Kodali, A., & Pattipati, K. (2013, January). An integrated health management process for automotive cyber-physical systems. In *Computing, Networking and Communications (ICNC), 2013 International Conference on*(pp. 82-86). IEEE.

Santucci, G., Martinez, C., & Vlad-Câlcic, D. (2012). The sensing enterprise. In *FInES workshop at FIA 2012* (pp. 1-14).

Schmidt, B., & Wang, L. (2015). Predictive maintenance: literature review and future trends. In *The International Conference on Flexible Automation and Intelligent Manufacturing (FAIM), 23-26 June 2015, University of Wolverhampton, UK* (Vol. 1, pp. 232-239). The Choir Press.

Schmidt, B., Galar, D., & Wang, L. (2016). Context awareness in predictive maintenance. In *Current trends in reliability, availability, maintainability and safety* (pp. 197-211). Springer, Cham.

Sejdovic, S., & Kleiner, N. (2016, July). Proactive and dynamic event-driven disruption management in the manufacturing domain. In *Industrial Informatics (INDIN), 2016 IEEE 14th International Conference on* (pp. 1320-1325). IEEE.

Sepehri, M. (2013). Strategic selection and empowerment of supplier portfolios case: oil and gas industries in Iran. *Procedia-Social and Behavioral Sciences*, *74*, 51-60.

Sharma, A., Yadava, G. S., & Deshmukh, S. G. (2011). A literature review and future perspectives on maintenance optimization. *Journal of Quality in Maintenance Engineering*, *17*(1), 5-25.

Sharma, R. K., & Sharma, P. (2010). System failure behavior and maintenance decision making using, RCA, FMEA and FM. *Journal of Quality in Maintenance Engineering*, *16*(1), 64-88.

Sharma, R. K., & Sharma, P. (2012). Integrated framework to optimize RAM and cost decisions in a process plant. *Journal of Loss Prevention in the Process Industries*, *25*(6), 883-904.

Shawash, J., & Selviah, D. R. (2013). Real-time nonlinear parameter estimation using the Levenberg–Marquardt algorithm on field programmable gate arrays. *IEEE Transactions on Industrial Electronics*, *60*(1), 170-176.

Shawash, J., & Selviah, D. R. (2013). Real-time nonlinear parameter estimation using the Levenberg–Marquardt algorithm on field programmable gate arrays. *IEEE Transactions on Industrial Electronics*, *60*(1), 170-176.

Sheu, S. H., Chang, C. C., Chen, Y. L., & Zhang, Z. G. (2015). Optimal preventive maintenance and repair policies for multi-state systems. *Reliability Engineering & System Safety*, *140*, 78-87.

Sheu, S. H., Griffith, W. S., & Nakagawa, T. (1995). Extended optimal replacement model with random minimal repair costs. *European journal of operational research*, *85*(3), 636-649.

Shrouf, F., Ordieres, J., & Miragliotta, G. (2014, December). Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm. In *Industrial Engineering and Engineering Management (IEEM), 2014 IEEE International Conference on* (pp. 697-701). IEEE.

Si, X. S., Zhang, Z. X., & Hu, C. H. (2017). A Real-Time Variable Cost-Based Maintenance Model. In *Data-Driven Remaining Useful Life Prognosis Techniques* (pp. 393-404). Springer, Berlin, Heidelberg.

Siemieniuch, C. E., & Sinclair, M. A. (2015). Global drivers, sustainable manufacturing and systems ergonomics. *Applied ergonomics*, *51*, 104-119.

Šikšnys, L., & Pedersen, T. B. (2018). Prescriptive Analytics. *Encyclopedia of Database Systems*. Springer.

Sinkkonen, T., Marttonen, S., Tynninen, L., & Kärri, T. (2013). Modelling costs in maintenance networks. *Journal of Quality in Maintenance Engineering*, *19*(3), 330-344.

Slack, N., Chambers, S., & Johnston, R. (2010). *Operations management*. Pearson education.

Soltanpoor, R., & Sellis, T. (2016, September). Prescriptive analytics for big data. In *Australasian Database Conference*(pp. 245-256). Springer, Cham.

Song, S., Coit, D. W., & Feng, Q. (2014). Reliability for systems of degrading components with distinct component shock sets. *Reliability Engineering & System Safety*, *132*, 115-124.

Su, L. H., & Tsai, H. L. (2010). Flexible preventive maintenance planning for two parallel machines problem to minimize makespan. *Journal of Quality in Maintenance Engineering*, *16*(3), 288-302.

Sun, Y., Ma, L., & Mathew, J. (2007). Prediction of system reliability for single component repair. *Journal of Quality in Maintenance Engineering*, *13*(2), 111-124.

Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., & Beghi, A. (2015). Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics*, *11*(3), 812-820.

Susto, G. A., Wan, J., Pampuri, S., Zanon, M., Johnston, A. B., O'Hara, P. G., & McLoone, S. (2014, August). An adaptive machine learning decision system for flexible predictive maintenance. In *Automation Science and Engineering (CASE), 2014 IEEE International Conference on* (pp. 806-811). IEEE.

Syed, B., Pal, A., Srinivasarengan, K., & Balamuralidhar, P. (2012, December). A smart transport application of cyber-physical systems: Road surface monitoring with mobile devices. In *Sensing Technology (ICST), 2012 Sixth International Conference on* (pp. 8-12). IEEE.

Szymanski, J., Bangemann, T., Thron, M., Thomesse, J. P., Reboeuf, X., Lang, C., & Garcia, E. (2003, September). PROTEUS-a European initiative for e-maintenance platform development. In *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA'03. IEEE Conference*(Vol. 2, pp. 415-420). IEEE.

Takata, S., Kirnura, F., van Houten, F. J., Westkamper, E., Shpitalni, M., Ceglarek, D., & Lee, J. (2004). Maintenance: changing role in life cycle management. *CIRP Annals-Manufacturing Technology*, *53*(2), 643-655.

Tam, A. S. B., Chan, W. M., & Price, J. W. H. (2006). Optimal maintenance intervals for a multi-component system. *Production Planning and Control*, *17*(8), 769-779.

Tang, D., Makis, V., Jafari, L., & Yu, J. (2015). Optimal maintenance policy and residual life estimation for a slowly degrading system subject to condition monitoring. *Reliability Engineering & System Safety*, *134*, 198-207.

Tang, D., Yu, J., Chen, X., & Makis, V. (2015). An optimal condition-based maintenance policy for a degrading system subject to the competing risks of soft and hard failure. *Computers & Industrial Engineering*, *83*, 100-110.

Tao, B., Ding, H., & Xiong, Y. L. (2003, October). IP sensor and its distributed networking application in e-maintenance. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on* (Vol. 4, pp. 3858-3863). IEEE.

Tao, F., Zuo, Y., Da Xu, L., & Zhang, L. (2014). IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing. *IEEE Transactions on Industrial Informatics*, *10*(2), 1547-1557.

Telford, S., Mazhar, M. I., & Howard, I. (2011, January). Condition based maintenance (CBM) in the oil and gas industry: An overview of methods and techniques. In *Proceedings of the 2011 International Conference on Industrial Engineering and Operations Management, Kuala Lumpur, Malaysia*.

Tennenhouse, D. (2000). Proactive computing. *Communications of the ACM*, *43*(5), 43-50.

Terkaj, W., Tolio, T., & Urgo, M. (2015). A virtual factory approach for in situ simulation to support production and maintenance planning. *CIRP Annals*, *64*(1), 451-454.

Thaduri, A., Kumar, U., & Verma, A. K. (2017). Computational intelligence framework for context-aware decision making. *International Journal of System Assurance Engineering and Management*, *8*(4), 2146-2157.

Thompson, J., & Rogers, S. (2017). *Analytics: How to Win with Intelligence*. Technics Publications.

Thullner, R., Rozsnyai, S., Schiefer, J., Obweger, H., & Suntinger, M. (2011, August). Proactive business process compliance monitoring with event-based systems. In *Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2011 15th IEEE International* (pp. 429-437). IEEE.

Tian, Z., Lin, D., & Wu, B. (2012). Condition based maintenance optimization considering multiple objectives. *Journal of Intelligent Manufacturing*, *23*(2), 333-340.

Trappey, A. J., Trappey, C. V., Govindarajan, U. H., Chuang, A. C., & Sun, J. J. (2017). A review of essential standards and patent landscapes for the Internet of Things: A key enabler for Industry 4.0. *Advanced Engineering Informatics*, *33*, 208-229.

Tsang, A. H. (1995). Condition-based maintenance: tools and decision making. *Journal of Quality in Maintenance Engineering*, *1*(3), 3-17.

Ucar, M., & Qiu, R. G. (2005). E-maintenance in support of e-automated manufacturing systems. *Journal of the Chinese institute of industrial engineers*, *22*(1), 1-10.

Usher, J. S., Kamal, A. H., & Syed, W. H. (1998). Cost optimal preventive maintenance and replacement scheduling. *IIE transactions*, *30*(12), 1121-1128.

Vainio, A. M., Valtonen, M., & Vanhala, J. (2008). Proactive fuzzy control and adaptation methods for smart homes. *IEEE Intelligent Systems*, *23*(2).

van der Weide, J. A., & Pandey, M. D. (2011). Stochastic analysis of shock process and modeling of condition-based maintenance. *Reliability Engineering & System Safety*, *96*(6), 619-626.

Van der Weide, J. A., Pandey, M. D., & van Noortwijk, J. M. (2010). Discounted cost model for condition-based maintenance optimization. *Reliability Engineering & System Safety*, *95*(3), 236-246.

Van Horenbeek, A., & Pintelon, L. (2013). A dynamic predictive maintenance policy for complex multi-component systems. *Reliability Engineering & System Safety*, *120*, 39-50.

Van Horenbeek, A., & Pintelon, L. (2015). A joint predictive maintenance and inventory policy. In *Engineering Asset Management-Systems, Professional Practices and Certification* (pp. 387-399). Springer, Cham.

Van Horenbeek, A., & Pintelon, L. (2015). A joint predictive maintenance and inventory policy. In *Engineering Asset Management-Systems, Professional Practices and Certification* (pp. 387-399). Springer, Cham.

Van Horenbeek, A., Buré, J., Cattrysse, D., Pintelon, L., & Vansteenwegen, P. (2013). Joint maintenance and inventory optimization systems: A review. *International Journal of Production Economics*, *143*(2), 499-508.

Van Horenbeek, A., Buré, J., Cattrysse, D., Pintelon, L., & Vansteenwegen, P. (2013). Joint maintenance and inventory optimization systems: A review. *International Journal of Production Economics*, *143*(2), 499-508.

Vansyckel, S., Schafer, D., Schiele, G., & Becker, C. (2013, September). Configuration management for proactive adaptation in pervasive environments. In *Self-Adaptive and Self-Organizing Systems (SASO), 2013 IEEE 7th International Conference on* (pp. 131-140). IEEE.

Varnier, C., & Zerhouni, N. (2012, May). Scheduling predictive maintenance in flow-shop. In *Prognostics and System Health Management (PHM), 2012 IEEE Conference on* (pp. 1-6). IEEE.

Vasili, M., Hong, T. S., Ismail, N., & Vasili, M. (2011). Maintenance optimization models: a review and analysis. *optimization*, *1*(2).

Vaughan, T. S. (2005). Failure replacement and preventive maintenance spare parts ordering policy. *European Journal of Operational Research*, *161*(1), 183-190.

Vaurio, J. K. (2011). Importance measures in risk-informed decision making: Ranking, optimisation and configuration control. *Reliability Engineering & System Safety*, *96*(11), 1426-1436.

Veldman, J., Wortmann, H., & Klingenberg, W. (2011). Typology of condition based maintenance. *Journal of Quality in Maintenance Engineering*, *17*(2), 183-202.

Veldman, J., Wortmann, H., & Klingenberg, W. (2011). Typology of condition based maintenance. *Journal of Quality in Maintenance Engineering*, *17*(2), 183-202.

Venkatesan, M. (1984). Production-inventory with equipment replacement—PIER. *Operations Research*, *32*(6), 1286-1295.

Vogl, G. W., Weiss, B. A., & Helu, M. (2016). A review of diagnostic and prognostic capabilities and best practices for manufacturing. *Journal of Intelligent Manufacturing*, 1-17.

Vohnout, K. D. (2003). Curve fitting and evaluation. *Mathematical Modeling for System Analysis in Agricultural Research*, 140-178.

Voisin, A., Levrat, E., Cocheteux, P., & Iung, B. (2010). Generic prognosis model for proactive maintenance decision support: application to pre-industrial e-maintenance test bed. *Journal of Intelligent Manufacturing*, *21*(2), 177-193.

Wan, E. A., & Van Der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000* (pp. 153-158). IEEE.

Wan, E. A., & Van Der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000* (pp. 153-158). Ieee.

Wan, J., Zhang, D., Zhao, S., Yang, L., & Lloret, J. (2014). Context-aware vehicular cyber-physical systems with cloud support: architecture, challenges, and solutions. *IEEE Communications Magazine*, *52*(8), 106-113.

Wan, S., Gao, J., Li, D., Tong, Y., & He, F. (2015). Web-based process planning for machine tool maintenance and services. *Procedia CIRP*, *38*, 165-170.

Wang, K. (2016). Intelligent predictive maintenance (IPdM) system–Industry 4.0 scenario. *WIT Transactions on Engineering Sciences*, *113*, 259-268.

Wang, S., Wan, J., Zhang, D., Li, D., & Zhang, C. (2016). Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. *Computer Networks*, *101*, 158-168.

Wang, W. (2011). A joint spare part and maintenance inspection optimisation model using the delay-time concept. *Reliability Engineering & System Safety*, *96*(11), 1535-1541.

Wang, W. (2012). A stochastic model for joint spare parts inventory and planned maintenance optimisation. *European Journal of Operational Research*, *216*(1), 127-139.

Wang, W. Z., Guo, Y. W., Huang, B. Y., Zhao, G. R., Liu, B. Q., & Wang, L. (2011, November). Analysis of filtering methods for 3D acceleration signals in body sensor network. In *Bioelectronics and Bioinformatics (ISBB), 2011 International Symposium on* (pp. 263-266). IEEE.

Wang, Y., & Cao, K. (2014). A proactive complex event processing method for large-scale transportation internet of things. *International Journal of Distributed Sensor Networks*, *10*(3), 159052.

Wang, Z., Hu, C., Wang, W., Kong, X., & Zhang, W. (2015). A prognostics-based spare part ordering and system replacement policy for a deteriorating system subjected to a random lead time. *International Journal of Production Research*, *53*(15), 4511-4527.

Want, R., Pering, T., & Tennenhouse, D. (2003). Comparing autonomic and proactive computing. *IBM Systems journal*, *42*(1), 129-135.

Weyrich, M., & Ebert, C. (2016). Reference architectures for the internet of things. *IEEE Software*, *33*(1), 112-116.

Widodo, A., & Yang, B. S. (2011). Application of relevance vector machine and survival probability to machine degradation assessment. *Expert Systems with Applications*, *38*(3), 2592-2599.

Will, N. C., & Cardoso, R. (2012, November). Comparative analysis between FFT and Kalman filter approaches for harmonic components detection. In *Industry Applications (INDUSCON), 2012 10th IEEE/IAS International Conference on* (pp. 1-7). IEEE.

Wu, F., Niknam, S. A., & Kobza, J. E. (2015). A cost effective degradation-based maintenance strategy under imperfect repair. *Reliability Engineering & System Safety*, *144*, 234-243.

Wu, S. J., Gebraeel, N., Lawley, M. A., & Yih, Y. (2007). A neural network integrated decision support system for condition-based optimal predictive maintenance policy. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *37*(2), 226-236.

Wu, S. J., Gebraeel, N., Lawley, M. A., & Yih, Y. (2007). A neural network integrated decision support system for condition-based optimal predictive maintenance policy. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *37*(2), 226-236.

Xia, T., Xi, L., Zhou, X., & Lee, J. (2013). Condition-based maintenance for intelligent monitored series system with independent machine failure modes. *International Journal of Production Research*, *51*(15), 4585-4596.

Xu, H., & Hu, W. (2008). Availability optimisation of repairable system with preventive maintenance policy. *International Journal of Systems Science*, *39*(6), 655-664.

Xu, T., Tang, T., Wang, H., & Yuan, T. (2013). Risk-based predictive maintenance for safety-critical systems by using probabilistic inference. *Mathematical Problems in Engineering*, *2013*.

Xu, Y., Zhang, Y., & Zhang, S. (2015, October). Uncertain generalized remaining useful life prediction-driven predictive maintenance decision. In *Prognostics and System Health Management Conference (PHM), 2015* (pp. 1-6). IEEE.

Yam, R. C. M., Tse, P. W., Li, L., & Tu, P. (2001). Intelligent predictive decision support system for condition-based maintenance. *The International Journal of Advanced Manufacturing Technology*, *17*(5), 383-391.

Yildirim, M., Sun, X. A., & Gebraeel, N. Z. (2016). Sensor-driven condition-based generator maintenance scheduling—Part I: Maintenance problem. *IEEE Transactions on Power Systems*, *31*(6), 4253-4262.

Yildirim, M., Sun, X. A., & Gebraeel, N. Z. (2016). Sensor-driven condition-based generator maintenance scheduling—part II: incorporating operations. *IEEE Transactions on Power Systems*, *31*(6), 4263-4271.

Yu, R., Iung, B., & Panetto, H. (2003). A multi-agents based E-maintenance system with case-based reasoning decision support. *Engineering applications of artificial intelligence*, *16*(4), 321-333.

Zagorecki, A., & Druzdzel, M. J. (2013). Knowledge engineering for Bayesian networks: How common are noisy-MAX distributions in practice?. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *43*(1), 186-195.

Zan, X., Zhang, S. X., Zhang, Y., Gao, H., & Han, C. S. (2018). A Decision-Making Model of Condition-Based Maintenance About Functionally Significant Instrument. In *Engineering Asset Management 2016* (pp. 329-338). Springer, Cham.

Zaplata, S., Meiners, M., & Lamersdorf, W. (2013). Designing future-context-aware dynamic applications with structured context prediction. *Software: Practice and Experience*, *43*(10), 1185-1204.

Zezulka, F., Marcon, P., Vesely, I., & Sajdl, O. (2016). Industry 4.0–An Introduction in the phenomenon. *IFAC-PapersOnLine*, *49*(25), 8-12.

Zheng, P., Sang, Z., Zhong, R. Y., Liu, Y., Liu, C., Mubarok, K., ... & Xu, X. (2018). Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives. *Frontiers of Mechanical Engineering*, 1-14.

Zhou, J., Djurdjanovic, D., Ivy, J., & Ni, J. (2007). Integrated reconfiguration and age-based preventive maintenance decision making. *IIE transactions*, *39*(12), 1085-1102.

Zhou, Y., Zhang, Z., Lin, T. R., & Ma, L. (2013). Maintenance optimisation of a multi-state series–parallel system considering economic dependence and state-dependent inspection intervals. *Reliability Engineering & System Safety*, *111*, 248-259.

Zhu, H., Liu, F., Shao, X., Liu, Q., & Deng, Y. (2011). A cost-based selective maintenance decision-making method for machining line. *Quality and Reliability Engineering International*, *27*(2), 191-201.

Zhu, Y., Elsayed, E. A., Liao, H., & Chan, L. Y. (2010). Availability optimization of systems subject to competing risk. *European Journal of Operational Research*, *202*(3), 781-788.

Appendix A: Questionnaire for Evaluation of PANDDA

Usefulness

Question ID	Question
PPUs1	I find it useful to get condition-based maintenance action recommendations
PPUs2	I find it useful to use PANDDA to monitor the actual cost of maintenance-related action implementation on the basis of sensor-enabled feedback
PPUs3	I find it useful to use PANDDA to update my initial estimation of maintenance-related action cost function on the basis of sensor-enabled feedback
PP9	I think that PANDDA is capable of improving my overall working experience
PP4	Do you have any comments or suggestions

Usability

Question ID	Question
PP1	My level of expertise related to maintenance decision support applications is high
PP2	I find it useful to use a graphical user interface to configure decision methods
PP3	The meaning of decision method instance in the PANDDA system is understandable to me
PP4	Do you have any comments or suggestions?
PP5	I can easily understand how to create a decision method instance
PP6	I can easily locate information about decision method instances (e.g. actions, costs)
PP7	I find the time needed to create a decision method instance reasonable
PP8	The PANDDA system provides me with an easy-to-use interface for creating decision method instances
PP9	I think that PANDDA is capable of improving my overall working experience
PP10	The PANDDA GUI behaves as the users expect
PP11	The use of the PANDDA GUI is easy and intuitive

PP12	I can easily understand the messages displayed by the system (e.g. error messages)
PP13	Navigation in the system's functions is simple and easy
PP14	The appearance of the system (design, aesthetics, colours) is attractive to the user
PP15	The response time of the system is acceptable
PP16	The system requires a lot of memory

List of Publications

Alexandros Bousdekis has 19 publications in scientific journals and proceedings of international scientific conferences, which have taken 71 citations. The h-index of Alexandros Bousdekis is 4 and has been calculated with the application Publish or Perish (25/06/2018).

Journal Publications

[j1] Bousdekis, A., Magoutas, B., Apostolou, D., & Mentzas, G. (2015). A proactive decision making framework for condition-based maintenance. *Industrial Management & Data Systems, 115*(7), 1225-1250. (24/05/2018 - Impact Factor: 2.205, 5-Year Impact Factor: 2.343)

[j2] Bousdekis, A., Magoutas, B., Apostolou, D., & Mentzas, G. (2015). Review, analysis and synthesis of prognostic-based decision support methods for condition based maintenance. *Journal of Intelligent Manufacturing*, 1-14. (24/05/2018 - Impact Factor: 3.035)

- **NTUA Thomaideio Award 2015***

[j3] Bousdekis, A., Papageorgiou N., Magoutas, B., Apostolou, D., & Mentzas, G. (2018). Information Processing for Generating Recommendations ahead of Time in an IoT-based Environment. *International Journal of Monitoring and Surveillance Technologies Research, 5*(4), 38-62.

[J4] Bousdekis, A., Papageorgiou N., Magoutas, B., Apostolou, D., & Mentzas, G. (2018). Enabling Condition-Based Maintenance Decisions with Proactive Event-driven Computing. *Computers in Industry*, *100*, 173-183 (24/05/2018 - Impact Factor: 2.691, 5-Year Impact Factor: 2.731)

Conference Publications - Proceedings

[c1] Magoutas, B., Stojanovic, N., Bousdekis, A., Apostolou, D., Mentzas, G., & Stojanovic, L. (2014). Anticipation-driven Architecture for Proactive Enterprise Decision Making. In CAiSE (Forum/Doctoral Consortium) (pp. 121-128).

[c2] Bousdekis, A., Magoutas, B., Apostolou, D., & Mentzas, G. (2015, April). Supporting the Selection of Prognostic-based Decision Support Methods in Manufacturing. In ICEIS (pp. 487-494).

* NTUA Thomaideio Award: Yearly awards of the National Technical University of Athens (NTUA) to PhD students for scientific journal publications included in high-impact databases (e.g. Scopus, Science Citation Index Expanded) and for scientific conference proceedings with full text after peer-review.

[c3] Bousdekis, A., & Mentzas, G. (2015). A Proactive Decision Support System for Maintenance Cost Minimisation in Manufacturing Enterprises. In 4th Student Conference of Hellenic Operational Research Society (HELORS) (pp. 61-66).

- **Best Paper Award**

[c4] Bousdekis, A., & Mentzas, G. (2016). A Multiple Criteria Approach Using ELECTRE for the Selection of Maintenance Strategy in Manufacturing Companies. In Proceedings of 5th International Symposium and 27th National Conference on Operational Research (pp. 117-121).

[c5] Bousdekis, A., Papageorgiou, N., Magoutas, B., Apostolou, D., & Mentzas, G. (2015). A real-time architecture for proactive decision making in manufacturing enterprises. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (pp. 137-146). Springer, Cham.

[c6] Bousdekis, A., Papageorgiou, N., Magoutas, B., Apostolou, D., & Mentzas, G. (2016). Continuous Improvement of Proactive Event-driven Decision Making through Sensor-Enabled Feedback (SEF). In ICEIS (pp. 166-173).

[c7] Bousdekis, A., Papageorgiou, N., Magoutas, B., Apostolou, D., & Mentzas, G. (2016). A probabilistic model for context-aware proactive decision making. In Information, Intelligence, Systems & Applications (IISA), 2016 7th International Conference on (pp. 1-6). IEEE.

[c8] Bousdekis, A., Papageorgiou, N., Magoutas, B., Apostolou, D., & Mentzas, G. (2017). A proactive event-driven decision model for joint equipment predictive maintenance and spare parts inventory optimization. Procedia CIRP, 59, 184-189.

- **Best Paper Award** in the 5th International Conference on Through-life Engineering Services (TESConf 2016), Cranfield, UK
- **NTUA Thomaideio Award 2016**

[c9] Bousdekis, A., & Mentzas, G. (2017). Condition-Based Predictive Maintenance in the Frame of Industry 4.0. In IFIP International Conference on Advances in Production Management Systems (pp. 399-406). Springer, Cham.

[c10] Bousdekis, A., Papageorgiou, N., Magoutas, B., Apostolou, D., & Mentzas, G. (2017). A Framework for Integrated Proactive Maintenance Decision Making and Supplier Selection. In IFIP International Conference on Advances in Production Management Systems (pp. 416-424). Springer, Cham.

[c11] Bousdekis, A., Papageorgiou, N., Magoutas, B., Apostolou, D., & Mentzas, G. (2017). An Information System for Deciding and Acting ahead of Time in Sensing Enterprises. In Information, Intelligence, Systems & Applications (IISA), 2017 8th International Conference on (pp. 1-6). IEEE.

[c12] Bousdekis, A., & Mentzas, G. (2018). A proactive model for joint maintenance and logistics optimization in the frame of Industrial Internet of Things. In Operational Research in Business and Economics. Springer International Publishing. (*In Press*)

[c13] Bousdekis, A., Mentzas, G., Hribernik, K., Lewandowski, M., von Stietencron, M., & Thoben, K. D. (2018). A Unified Architecture for Proactive Maintenance in Manufacturing Enterprises. In Enterprise Interoperability VIII: Proceedings of the I-ESA Conference 2018. Springer International Publishing. (*In Press*)

[c14] Bousdekis, A., Magoutas, B., Apostolou, D., Mentzas, G., & Puhar, P. (2018). The ProaSense Platform for Predictive Maintenance in the Automotive Lighting Equipment Industry. In I-ESA ’18 Workshop Proceedings. ISTE-Wiley. (*In Press*)

[c15] Lepenioti, K., Bousdekis, A., Apostolou, D., Mentzas, G. (2018). Prescriptive Analytics: A Survey of Approaches and Methods. In International Conference on Business Information Systems (BIS). Springer, Cham. (*In Press*)

Resume

Mr. Alexandros Bousdekis graduated from the School of Production and Management Engineering of the Technical University of Crete (TUC), Greece, in 2011. He also pursued an MSc degree in Manufacturing Systems Engineering of the Warwick Manufacturing Group (WMG) at the University of Warwick, UK, in 2012. He is Member of the Technical Chamber of Greece (TEE) and owner of professional permits in Mechanical and Electrical Engineering from 2011. He holds the Certificate of Proficiency in English provided by the University of Michigan (excellent knowledge of english), the Diplome Approfondi de Langue Francaise (DALF) provided by the Ministere del' Education Nationale, de l' Enseignment Superieur et de la Recherche of France (excellent knowledge of french) and the State Certificate of the Italian Language Knowledge provided by Ministry of Education of Greece (very good knowledge of Italian).

Since 2014, Mr. Bousdekis has been working as a researcher at the Information Management Unit (IMU) of the Institute of Communication and Computer Systems (ICCS) of the National Technical University of Athens (NTUA), where he has participated in two (2) research and innovation projects. Since 2014, he has been a teaching assistant at the National Technical University of Athens (NTUA) for the courses: "Project Management" and "Management of the Digital Enterprise". Since 2014, he has also been working as a lecturer at the Mediterranean College where he has been teaching "Operations Management", "Logistics and Distribution Management" and "Information Systems and Technologies for Logistics" at a BSc and MSc level as well as at the Professional Executive Diplomas.

Since 2014, Mr. Bousdekis has been a PhD candidate at the IMU of the School of Electrical and Computer Engineering of NTUA. His PhD thesis gets involved with the exploitation of proactive computing for optimizing maintenance operations in the context of Industry 4.0. His research interests include: proactive computing, operations management, predictive maintenance, predictive and prescriptive analytics, decision making algorithms and technologies.