A web-based multilevel framework for condition monitoring of industrial equipment

WJ van Blerk



orcid.org/0000-0002-3427-1525

Dissertation submitted in fulfilment of the requirements for the degree *Master of Engineering in Computer and Electronic Engineering* at the North-West University

Supervisor: Dr JC Vosloo

Graduation ceremony: May 2019

Student number: 29892139

ABSTRACT

Title: A web-based multilevel framework for condition monitoring of industrial equipment

Author: WJ van Blerk

Supervisor: Dr Jan Vosloo

Keywords: Condition monitoring, industrial equipment, industrial machinery, web-based framework, online framework, configurable system, cognitive load reduction

Proactive maintenance strategies aim to maintain equipment before failure and, in doing so, avoid expensive repair costs. Condition monitoring provides relevant stakeholders with information on the mechanical health status of equipment. Condition monitoring is therefore a useful tool for assisting with proactive maintenance.

Condition monitoring consists of three key steps, namely, data acquisition, data processing, and information transfer and visualisation. Existing literature focuses heavily on data processing techniques, while literature on condition monitoring information transfer and visualisation is limited. The information transfer and visualisation aspect of existing condition monitoring systems focuses mostly on the detailed data of measurements and components.

Management level personnel in industrial organisations typically use this condition monitoring information to make resource allocation decisions. These personnel are, however, responsible for numerous operations and manual investigation of each component in these organisations, which creates cognitive overload.

This study aims to improve the information transfer and visualisation to relevant stakeholders by reducing the cognitive load and increasing the data accessibility. Thereafter, the study investigates if these improvements can improve proactive maintenance in an industrial organisation by reducing the reaction times to faulty equipment.

This study develops a generic, multilevel web-based framework to visualise the processed condition monitoring data of industrial organisations. The multilevel web-based system gives a range of stakeholders remote access to the condition monitoring data to encourage timeous maintenance decisions. The system is generic and usable in numerous industries.

The system was implemented across nine operations and five system groups in a mining organisation. The system significantly reduced the cognitive load at management levels, such

as the organisational and operational level, while maintaining acceptable levels of cognitive load at detail levels.

The system successfully reduced the maintenance reaction times relative to the number of maintenance reactions. As with most practical implementations, there were exceptions: one of the operations showed an increase in relative reaction times. Further investigation indicated that this increase was due to a significant drop in the number of faulty equipment, causing a decreasing denominator. The system is therefore considered a success.

ACKNOWLEDGEMENTS

Thanks to ETA Operations (Pty) Ltd, Enermanage, and its sister companies for the resources, time and financial assistance to complete this study.

Furthermore, I would like to thank the following people:

- My parents, for the opportunities they have given me in life and for their continued support.
- Dr JC Vosloo, Dr JN du Plessis and Dr S van Jaarsveld, for their guidance, insights and support.
- Prof. EH Mathews and Prof. M Kleingeld, for providing the opportunity to further my studies at CRCED Pretoria.

CONTENTS

Abstract ... **ii**
Acknowledgements ... **iv**
Contents ... **v**
List of Figures ... **vii**
List of Tables ... **xi**
List of Abbreviations ... **xii**
Nomenclature ... **xiii**
1 Introduction ... **1**
1.1 Preamble ... 1
1.2 Maintenance in the industry ... 1
1.3 Condition monitoring in the industry ... 5
1.4 Cognitive load at management level ... 7
1.5 Problem statement ... 8
1.6 Objectives ... 9
1.7 Methodology ... 10
1.8 Outline ... 11
2 Literature review ... **12**
2.1 Preamble ... 12
2.2 Existing solutions to industrial condition monitoring ... 12
2.3 Effective user interface design ... 24
2.4 Framework development in a software environment ... 33
2.5 Summary ... 34
3 Design overview ... **36**
3.1 Preamble ... 36
3.2 Requirements ... 36
3.3 Development context ... 38
3.4 Functional design ... 46
3.5 Summary ... 48
4 Detailed design ... **50**
4.1 Preamble ... 50
4.2 Front-end system ... 50
4.3 Database ... 60
4.4 Configuration tool ... 67
4.5 System logic ... 72
4.6 Verification ... 79
4.7 Summary ... 87
5 Results ... **89**
5.1 Preamble ... 89
5.2 Implementation ... 89
5.3 Background on case study ... 89
5.4 Validation ... 90
5.5 Summary ... 110

6 Conclusions and recommendations 111
6.1 Conclusions of the conducted work 111
6.2 Recommendations for future work 114
Reference List 117
Appendix A 123

LIST OF FIGURES

Figure 1: Illustration of the scale of data collection in industrial organisations 6
Figure 2: The three stages of information processing as proposed by Atkinson and Shiffrin (Adapted from [17]) 8
Figure 3: The study methodology 10
Figure 4: Left: A handheld thermal camera; Right: An infrared thermal image of a CNC machine 15
Figure 5: Top: Example of time-domain vibration data; Bottom: Example of corresponding frequency-domain vibration data [34] 22
Figure 6: Examples of component layout diagrams in condition monitoring systems 23
Figure 7: Example of an application of a column graph in a condition monitoring system [34] 23
Figure 8: The basic layout of the TurbinePhD™ fleet view[36] 24
Figure 9: Graphic presentation used to visualise instructional efficiency [35] 26
Figure 10: Introducing anomalies into a harmonious pattern versus a chaotic pattern 30
Figure 11: Demonstration of the effect of colour on depth perception 31
Figure 12: Demonstration of the effect of relative element size on user attention and focus 32
Figure 13: Demonstration of the effect of misalignment on user attention and focus 32
Figure 14: Existing condition monitoring functionality 38
Figure 15: Partial database ERD of structure-related tables 39
Figure 16: Visual representation of an example of a mining organisational structure 40
Figure 17: Partial database ERD of tree-structure-related tables 40
Figure 18: Visual representation of an example of a pumping tree structure 41
Figure 19: Partial database ERD of user and organisational tables 41
Figure 20: Partial database ERD of a user and user password table 42
Figure 21: Partial database ERD of a user and user right tables 42
Figure 22: Partial database ERD of a user and applicable structure tables 43
Figure 23: Propagated access in a pumping tree structure of a user with access to Mine 2 43
Figure 24: Various types of data tables 44
Figure 25: Partial database ERD of the tag-related tables 44
Figure 26: Partial database ERD of data-related tables 45
Figure 27: DBMS node builder tool 46
Figure 28: Simplified systems diagram of existing infrastructure 46
Figure 29: Integration of the system with existing infrastructure 47
Figure 30: Phases of condition monitoring 48
Figure 31: Illustration of system adaptability to data acquisition and data processing steps 48

Figure 32: Illustration of the condition monitoring parameters to measure in an organisation 50
Figure 33: Example of the system front-end user interface 51
Figure 34: Typical component status blocks 51
Figure 35: Status block legend 51
Figure 36: Status block with customised tooltip text 52
Figure 37: Status blocks with display values 52
Figure 38: Organisational overview grid 53
Figure 39: Operational system overview grid 54
Figure 40: Various operational systems inherit the system statuses of their lower levels 55
Figure 41: Tree structure node Level Up button 56
Figure 42: Navigation to organisational overview level using the node picker 56
Figure 43: Navigation to operational overview level using the node picker 56
Figure 44: Navigation to operational system overview level using the node picker 57
Figure 45: Grid block with graph linked 57
Figure 46: Graph with a single limit 58
Figure 47: Graph with multiple unique limits 58
Figure 48: Date range and date picker 59
Figure 49: Date range options 59
Figure 50: Graph displaying data for a seven-day period 59
Figure 51: Graph displaying data for a 14-day period 60
Figure 52: Illustration of the number of duplicate nodes in an organisational structure 61
Figure 53: Combination of two tree structures to form operational system node links 62
Figure 54: Combination of two tree structures to form component parameter node links 62
Figure 55: Condition monitoring business tree structure 63
Figure 56: Condition monitoring data tree structure 63
Figure 57: Partial database ERD of the node-link-related tables 64
Figure 58: Comparison between single tree structure and two-tree structure approach 64
Figure 59: Partial database ERD of the format tag-related tables 65
Figure 60: Partial database ERD of the graph and series related tables 66
Figure 61: Basic DBMS condition monitoring configuration tool user interface 67
Figure 62: Left: Business tree structure node tree; Right: Data tree structure node tree 68
Figure 63: Create Node Link button with breadcrumbs of the selected nodes 68
Figure 64: Delete Node Link button with breadcrumbs of the selected nodes 69
Figure 65: Node link formatting configuration 69
Figure 66: Tag search dialog 69
Figure 67: Format tag type drop-down menu 70

Figure 68: Node link graph configuration section 70
Figure 69: Node link graph only configuration 71
Figure 70: Graph ordering dialog 71
Figure 71: Node link series only configuration 71
Figure 72: Series limit dialog 72
Figure 73: Extended ASP.NET MVC architecture [16] 73
Figure 74: Process diagram of grid loading by node link selection 74
Figure 75: Process diagram of grid loading by node selection 75
Figure 76: Process diagram of grid construction 76
Figure 77: Process diagram of grid formatting 77
Figure 78: Process diagram of graph configuration for fusion charts 78
Figure 79: Graph presenting half-hourly average temperature values 80
Figure 80: Status blocks representing pump health statuses 80
Figure 81: Parameter detail level of a deep level gold mine pump's temperatures 81
Figure 82: User with full access. Left: Condition monitoring node tree, Right: User access control setup 82
Figure 83: User with limited access. Left: Condition monitoring node tree; Right: User access control setup 82
Figure 84: Left: New organisational condition monitoring business tree structure; Right: New organisational condition monitoring data tree structure 83
Figure 85: Top: Two nodes selected before creating a node link; Bottom: Two nodes selected after creating a node link 83
Figure 86: Node links on the front-end system 84
Figure 87: Node link with a deeper level of navigation 84
Figure 88: Top: Status information configuration; Bottom: Status on front-end system 85
Figure 89: Top: Customised tooltip text and display value configuration; Bottom: Customised tooltip text and display value on front-end system 85
Figure 90: Top: Graph and series configuration; Bottom: Graph and series on front-end system 86
Figure 91: Organisational overview level 92
Figure 92: Linear comparison of the number of parameters and the number of visual elements at the organisational overview level 92
Figure 93: Logarithmic comparison of the number of parameters and the number of visual elements at the organisational overview level 93
Figure 94: Pumping operational system overview level 93
Figure 95: Linear comparison of the average number of parameters and the average number of visual elements at operational system overview levels 94

Figure 96: Logarithmic comparison of the average number of parameters and the average number of visual elements at operational system overview levels 94
Figure 97: Pump temperature parameter detail level 95
Figure 98: Linear comparison of the average number of data points and the average number of visual elements at parameter detail levels 96
Figure 99: Logarithmic comparison of the average number of data points and the average number of visual elements at parameter detail levels 96
Figure 100: Maintenance reaction time measurement 99
Figure 101: Average reaction times for the organisation over 15 months 100
Figure 102: Average reaction times vs reaction count for the organisation over 15 months 100
Figure 103: Average reaction times relative to the reaction count for the organisation over 15 months 101
Figure 104: Average reaction times for Operation 4 over 11 months 102
Figure 105: Average reaction times for Operation 9 over 15 months 103
Figure 106: Average reaction times vs reaction count for Operation 4 over 11 months 103
Figure 107: Average reaction times vs reaction count for Operation 9 over 15 months 104
Figure 108: Average reaction times relative to the reaction count for Operation 4 over 11 months 104
Figure 109: Average reaction times relative to the reaction count for Operation 9 over 11 months 105
Figure 110: Average reaction times of the various operations 107
Figure 111: Average time to configure an average operation 109

LIST OF TABLES

Table 1: Summary of system requirements 37
Table 2: Descriptions of the various status block statuses 52
Table 3: Descriptions of the various format types 65
Table 4: Summary of applicable GridVM object properties 73
Table 5: Summary of applicable GridBlockVM object properties 73
Table 6: Summary of applicable GridBlockDisplayInfoVM object properties 74
Table 7: Summary of the amount of information that the system monitors 91
Table 8: Concepts applicable to the results of this subsection 99
Table 9: Raw data of minutes spent configuring grid blocks during Development Phase 1 123
Table 10: Raw data of minutes spent configuring format tags during Development Phase 1 123
Table 11: Raw data of minutes spent configuring graphs during Development Phase 2 124
Table 12: Raw data of minutes spent configuring data tags during Development Phase 3 124
Table 13: Configuration times for a partial operation during Development Phase 2 125
Table 14: Configuration times for a partial operation during Development Phase 3 125
Table 15: Extrapolated confirmation times of various development phases 126

LIST OF ABBREVIATIONS

ARM	Autoregressive modelling
BS	Business structure
CNC	Computer numerical control
DAL	Data access layer
DB	Database
DBMS	Database management system
DE	Drive end
DS	Data structure
DSM	Demand-side management
EMS	Energy management system
ERD	Entity relationship diagram
ESCo	Energy services company
FK	Foreign key
GUI	Graphical user interface
IT	Information technology
LDA	Linear discriminant analysis
NDE	Non-drive end
OPC	Open platform communications
SCADA	Supervisory control and data acquisition
SMS	Short message service
SVM	Support vector machine
VM	View model

NOMENCLATURE

ASP.NET MVC	A version of the Microsoft® ASP.NET web application framework that uses a model-view-controller pattern.
DasBox	GS&S's condition monitoring system for industrial plant maintenance.
Fault diagnosis	Identifying the root cause and severity of a mechanical fault in a piece of equipment.
Fault prognosis	Estimation of the remaining useful operational life of a faulty piece of equipment.
FusionCharts	A JavaScript library for displaying charts in web applications.
GS&S	Global Solutions & Services – an Italian engineering company.
Plug and play	An interface between two computer components that does not require reconfiguration or manual installation. In terms of software, it refers to two systems that can integrate without the need for additional development.
NRG Systems	A leading company in the renewable energy industry.
TurbinePhD™	NRG Systems' wind turbine condition monitoring system.

1 Introduction

1.1 Preamble

This chapter provides background to the identified problem. Section 1.2 gives an overview of the role of maintenance in industrial organisations. Section 1.3 describes condition monitoring as a tool to assist with the maintenance of industrial equipment and the associated challenges. Section 1.4 describes the problem of cognitive overload in maintenance management and condition monitoring.

Section 1.5 formulates the problem and justifies the need for the study. Section 1.6 describes the objectives, which aim to solve the problem defined in Section 1.5. Section 1.7 describes the methodology that will be followed to achieve and validate the objectives defined in Section 1.6.

Section 1.8 provides an overview of the document.

1.2 Maintenance in the industry

1.2.1 The need for maintenance

The main driver for industrial organisations is profit. Most industrial organisations operate in either the primary (extraction of raw materials) or secondary (manufacturing and construction) economical sector. A study done on the American manufacturing industry identified that quality, cost and product lead time are the main contributing factors to a company's competitiveness. These three factors are, however, not only applicable to the American manufacturing industry but also to the global primary and secondary sectors [1].

Production quality, cost and product lead time are all influenced by equipment reliability and maintenance. Well-maintained equipment can manufacture to finer tolerances, therefore producing higher quality products with greater consistency. This reduces the number of scrap pieces produced, which reduces costs. Well-maintained equipment increases product lead time through reduced downtime [1]. Extended downtime caused by poorly maintained equipment will not only hinder product lead time but can also have a detrimental effect on cost-saving initiatives and safety in the workplace.

Electrical cost savings projects, such as energy efficiency and demand-side management (DSM) projects, are largely dependent on the efficiency and availability of equipment [2], [3]. Multiple studies have found that optimising a maintenance strategy can lead to significant cost

savings by reducing energy usage, reducing maintenance costs and increasing product lead time [4]–[6].

Equipment downtime can potentially be a major safety concern. Ventilation systems are responsible for extracting potentially harmful gases. Refrigeration systems are used to ensure safe working temperatures. Dewatering pumps on mines are responsible for flood prevention on deep levels. Downtime of any of these systems puts the health and lives of workers at risk.

Operating and maintaining industrial equipment reliably for an extended period is no simple task. Maintenance of industrial equipment accounts for 25–40% of total equipment cost (procurement, installation and operation included) depending on the type of equipment and maintenance strategy used [3], [4]. Applying overly conservative maintenance (doing maintenance too often) can lead to unnecessary maintenance costs and equipment downtime. Not applying maintenance often enough will eventually lead to equipment failure, high replacement costs and extended equipment downtime. The various types of maintenance are described in Section 1.2.2.

1.2.2 Types of maintenance

In his PhD thesis, *A Performance-centered Maintenance Strategy for Industrial DSM Projects*, Groenewald divides maintenance into five types, namely: breakdown maintenance, corrective maintenance, preventative maintenance, reliability-centred maintenance and total productive maintenance. Preventative maintenance is further divided into time-based and condition-based maintenance [4]. Another maintenance type, not mentioned by Groenewald, is predictive maintenance [7]. Each of these maintenance types can be categorised as either reactive or proactive maintenance. A short description of each category and maintenance type is provided in Section 1.2.2.1 and Section 1.2.2.2.

1.2.2.1 Reactive maintenance

Reactive maintenance typically occurs after equipment damage or failure. The various types of reactive maintenance are described below.

- **Breakdown maintenance:** Maintenance performed after equipment failure has occurred. This typically entails replacing components [4].
- **Corrective maintenance:** Upgrades to equipment with the goal of improving component reliability or correcting a flawed designed [4].
- **Total productive maintenance:** Operators are not only responsible for production, but also for reporting on maintenance needs. The aim is to maximise production while

maintaining equipment reliability [4]. Operators tend to lack the skills necessary to make maintenance decisions before equipment damage occurs.

1.2.2.2 Proactive maintenance

Proactive maintenance is aimed at preventing equipment damage or failure. The various types of proactive maintenance are described below.

- **Preventative maintenance:** Maintenance performed with the goal of preventing equipment failure:
  - Time-based maintenance: Maintenance done on a predetermined schedule [4].
  - Condition-based maintenance: Maintenance done based on the current condition of equipment [4].
- **Reliability-centred maintenance:** Preventative maintenance performed only on parts that directly affect the overall system reliability [4].
- **Total productive maintenance:** Operators are not only responsible for production, but also for reporting on maintenance needs. The aim is maximising production while maintaining equipment reliability [4]. A properly trained and knowledgeable operator makes proactive maintenance decisions before equipment damage occurs.
- **Predictive maintenance:** The remaining useful life of equipment is predicted based on the current equipment health, historical data and manufacturer specifications [8]. Maintenance is done as close to the predicted date of failure as possible. The company can decide how much risk it is willing to take and how long before the predicted date of failure maintenance should be done.

Proactive maintenance tends to be more cost-effective than reactive maintenance. Proactive maintenance sacrifices equipment uptime in the short term for higher equipment reliability in the long term. The temporary loss in production rates is offset by the increase in equipment life. Proactive maintenance thus shields the company from expensive component replacement costs and extended equipment downtimes [7], [9].

When comparing the cost of general reactive maintenance with proactive maintenance types, such as preventative and predictive maintenance, the benefits of proactive maintenance become clear. A study on substation and service transformers found that reactive maintenance costs can increase the total equipment costs by up to 40%. Preventative maintenance can reduce these costs to approximately 15%. Predictive maintenance can reduce these costs further to 5–6% of the total equipment cost [7].

1.2.3 Maintenance practices in the industry

The industry consists of a wide range of organisations with different backgrounds and philosophies on maintenance. Many of these organisations use reactive maintenance strategies[1] [4], [7], [10]. Section 1.2.1 noted the safety concerns of a poor maintenance strategy while Section 1.2.2 identified the clear benefits of proactive maintenance over reactive maintenance. Organisations that implement reactive maintenance strategies will be forced to implement proactive maintenance at some point to remain competitive.

In many cases, industrial organisations aim to implement proactive maintenance strategies such as scheduled or time-based preventative maintenance, but in practice, maintenance is done in a reactive manner [11]. Companies tend to fall back on reactive maintenance due to various factors. It is difficult to predict equipment failure without the necessary information on equipment health. This combined with the ambiguity of the financial impact of performing maintenance before failure can very easily lead to a mindset of "if it isn't broke, don't fix it"[1] [6].

Additionally, short-term production rate goals are prioritised over temporary downtime for maintenance. In the South African mining industry, for instance, maintenance work orders are typically generated through daily inspections [10]. These inspections are done to identify equipment breakdowns or damage. Even though the mine may have a maintenance schedule in place for the equipment it is operating, it will often not adhere to the schedule as the inspections indicate that the equipment is still running.

The protocol for allocating maintenance resources, such as personnel, capital and equipment downtime, can vary from organisation to organisation; however, the basic outline is usually similar. In organisations with limited infrastructure, maintenance work orders are generated through equipment inspections, conducted on a predetermined schedule by trained personnel, or on a word-of-mouth basis where control room operators report to their supervisors if equipment behaves abnormally or breaks down [10]. Some organisations have implemented computerised maintenance management systems. These systems can be integrated with condition monitoring systems, as discussed in Section 1.3, to automate the generation of maintenance work orders and assign the applicable personnel.[2]

---

1 https://www.linkedin.com/pulse/key-advantages-proactive-vs-reactive-maintenance-ben-hailes/ [Accessed February 2018]

2 https://www.nrgsystems.com/assets/resources/Understanding-Wind-Turbine-Condition-Monitoring-Systems-Whitepaper.pdf [Accessed September 2018]

Maintenance work orders are escalated to the applicable organisational level depending on the severity, risk and resources required. Maintenance managers are then responsible for decision-making and resource allocation regarding maintenance.

1.3 Condition monitoring in the industry

1.3.1 The need for condition monitoring

Condition monitoring is the process of measuring and logging the operational parameters of a machine with the main purpose of observing changes that may be indicative of a potential fault before equipment failure occurs.[3, 4] Vibration and temperature are the most commonly used parameters for determining the mechanical health of equipment through condition monitoring as they are scalar values that are relatively simple to measure [12]. Other parameters and methods that may be used for condition monitoring purposes are oil debris presence, oil pressure, motor current, ultrasonic analysis and infrared thermography[5] [13].

Section 1.2.2 motivated the benefits of proactive maintenance over reactive maintenance strategies in industrial organisations. Condition monitoring is a useful tool for the implementation of preventative maintenance strategies and is crucial for the implementation of predictive maintenance strategies [4], [7], [13].

1.3.2 Challenges of industrial condition monitoring

One of the main challenges of implementing condition monitoring systems in industrial organisations is a lack of infrastructure. Large industrial organisations have been operating for decades before condition monitoring technology became available. This means that vast amounts of equipment and machinery may be operating without any form of measurement or data logging equipment installed. Implementing a condition monitoring system in such an organisation will require significant instrumentation installation resources [14].

Furthermore, these organisations may also face personnel who resist change. Uninformed, floor-level personnel may feel that their jobs are threatened by the automation of monitoring processes. Computer literacy can be very limited with floor-level and older personnel, which can lead to a reluctance to use a new system if employees feel intimidated by a complex system.

---

3 https://www.corrosionpedia.com/definition/314/condition-monitoring-cm [Accessed February 2018]

4 https://www.slideshare.net/ElenaMariaVaccher/plant-maintenance-condition-monitoring [Accessed February 2018]

5 http://www.plant-maintenance.com/articles/ConMon21stCentury.shtml [Accessed February 2018]

Large industrial organisations can typically consist of multiple operations spread across numerous geographical regions. An operation will typically operate various systems, each consisting of multiple pieces of equipment. For effective condition monitoring, parameters should be measured at multiple points on each piece of equipment [15]. Logging all these parameters across the organisation at high resolutions can generate extremely large volumes of data. Figure 1 illustrates the scale of data collection for a condition monitoring system in an industrial organisation.

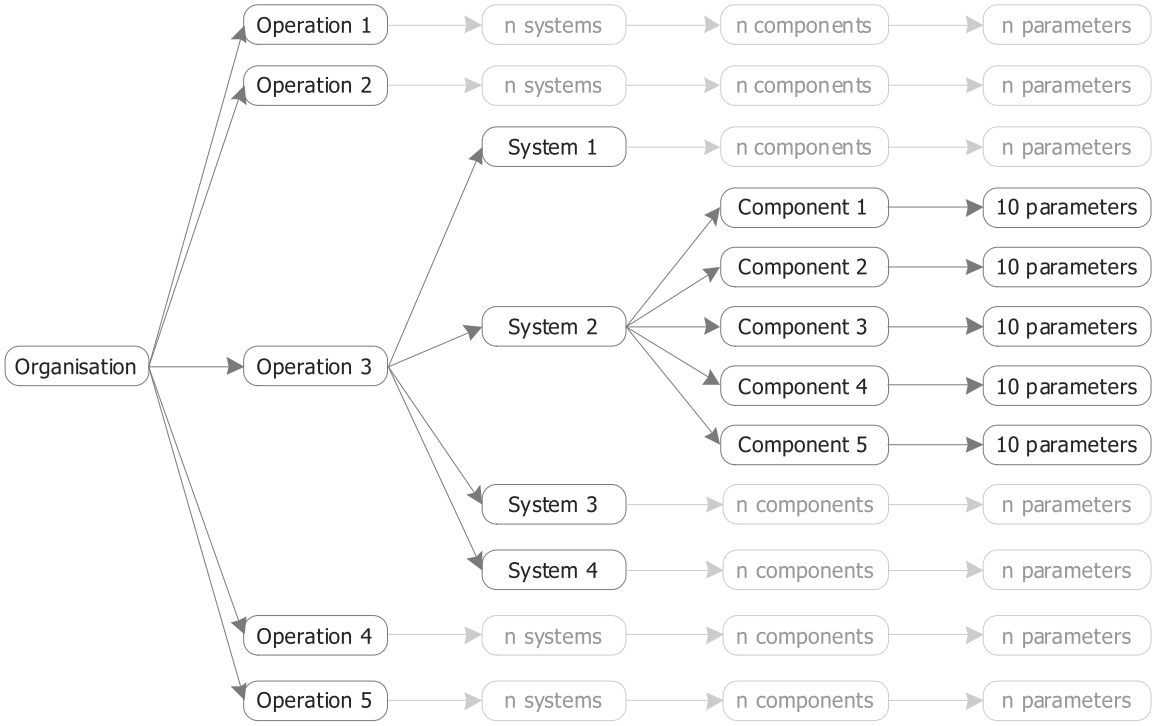


Figure 1: Illustration of the scale of data collection in industrial organisations

At control room level, simple time-domain graphs of each parameter may be sufficient to analyse and monitor equipment health. However, at operational or organisational level, this is not practical as a single organisation can measure more than 1500 parameters at a given time, as verified in Chapter 5, Section 5.4.1.2, Table 7. It is important that condition monitoring systems condense this data into useful information.

1.3.3 Requirements for industrial condition monitoring

The overarching purpose of condition monitoring systems is to provide information to the organisation and its employees. Decision makers can use this information for planning maintenance tasks, preventing potential safety risks and optimising equipment performance. It is therefore imperative that a condition monitoring system provides relevant information to the various stakeholders. For example, control room operators or vibration analysts typically need to access fine resolution data of a single piece of equipment or machinery. On the other

hand, a regional maintenance manager benefits more from an overview of the operations in his/her region.[6]

Information is pointless if it is not accessible to those it was intended for. Due to the potential widespread nature of industrial organisations, a network-based condition monitoring system is impractical. Web-based condition monitoring systems enable users to access information regarding equipment conditions from anywhere where an internet connection is available. Additionally, the software of web-based systems tends to be simpler to maintain and does not require on-site information technology (IT) technicians for installation[7] [16].

In modern times, industrial organisations often have existing IT networks, software platforms and data logging infrastructure in place. Developing and installing the necessary infrastructure for a condition monitoring system from scratch for every industrial organisation is not ideal as it is a resource-intensive process. It is therefore beneficial if a condition monitoring system can integrate seamlessly with existing infrastructure.

In terms of the scale of operations, amount of equipment/machinery and types of equipment/ machinery, there is a wide variety of industrial organisations. These organisations' needs for a condition monitoring system are equally varying. A condition monitoring system should therefore be adaptable for various applications. The system should reflect each organisation's needs individually in terms of information, evolution and expansion.

The rate of implementation is another key component of a condition monitoring system. If an organisation commits to using condition monitoring, the condition monitoring system should be implemented as quickly and effortlessly as possible without disrupting normal operations.

1.4 Cognitive load at management level

"Cognitive load refers to the total amount of mental effort being used in the working memory".[8] Humans have limited information processing ability and they have difficulty processing large volumes of information. A study on human memory suggests that humans process information in three stages. Figure 2 presents the three stages, which suggest that too much information will not only be hard to process and interpret, but will also be difficult to remember [17].

6 https://www.windpowerengineering.com/business-news-projects/featured/condition-monitoring-made-easy-to-read/ [Accessed September 2018]

7 http://www.differencebetween.net/technology/software-technology/difference-between-client-server-application-and-web-application/ [Accessed September 2018]

8 http://teotti.com/cognitive-overload-in-software-development/ [Accessed February 2018]

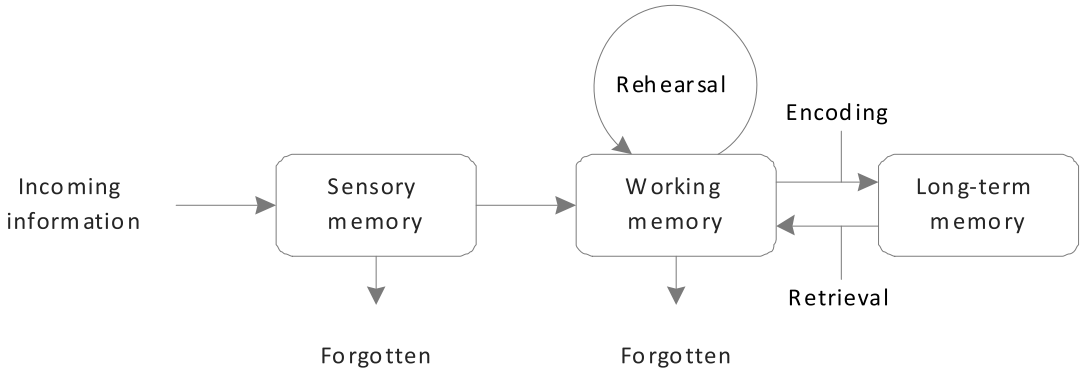


Figure 2: The three stages of information processing as proposed by Atkinson and Shiffrin (Adapted from [17]**)**

Employees operating at management level must typically deal with complex schedules, numerous projects and resource management. The cognitive load placed on these employees is inherently high. It is thus important to avoid cognitive overload when presenting managers with additional information.

Figure 1 and Chapter 5, Section 5.4.1, give insight into the amount of information that condition monitoring system can generate. Managers need to be able to interpret this information to make well-informed maintenance and resource allocation decisions. Information condensing and simplification are essential to avoid cognitive overload for these decision makers.

1.5 Problem statement

Industrial organisations can operate vast amounts of equipment that are critical to production rates and safety in the workplace. This equipment is typically spread across numerous operations and geographical regions. Maintenance is required for continual use of this equipment and cost-effective operation of the organisation. Among costs such as procurement, installation and operating costs, maintenance can add up to 40% of the overall equipment cost [4], [7]. This leaves significant room for the optimisation of maintenance strategies to minimise costs.

Many industrial organisations tend to use reactive maintenance strategies due to various reasons [4], [7], [10]. The constant drive for high production rates, the ambiguity of the financial impact of maintenance decisions, and the difficulty of predicting remaining equipment life before breakdown are some of the main reasons for the reactive trend.

Studies have proven that proactive maintenance strategies, such as preventative and predictive maintenance, have significant long-term financial benefits over reactive maintenance [4], [7]. Condition monitoring is a useful tool to assist with preventative and

predictive maintenance strategies. Through condition monitoring, the real-time mechanical condition and equipment health can be determined remotely. Maintenance managers can use this information strategically to apply maintenance before equipment failure occurs [4], [7], [13].

The need for condition monitoring varies among different industrial organisations. The size of the organisation, the organisation's equipment, and the organisation's financial state influence its needs for condition monitoring. A condition monitoring system therefore needs to be adaptable in the information it provides, its scalability, and its integration with existing infrastructure.

Various stakeholders typically use the system as part of the wide variety of needs that a condition monitoring system has to fulfil. These stakeholders have different job descriptions and operate at different levels within the organisation. A control room operator typically needs to access different information than an organisational-level maintenance manager.[9] These stakeholders typically operate from different locations. The system and data therefore need to be accessible remotely.

Condition monitoring at an organisational level can be challenging due to the vast amount of data that is generated. If the information from the condition monitoring system is difficult or impractical to interpret and analyse, users can experience cognitive overload [17]. This is a typical contributor to organisations reverting to reactive maintenance strategies. A condition monitoring system therefore needs to condense the available data into useful information for all relevant stakeholders.

1.6 Objectives

The objectives of the study are focused on solving the problems of condition monitoring of industrial equipment, identified in Section 1.5, by developing a web-based framework for condition monitoring. The objectives of the study are described below:

Literature review objectives

- Investigate existing solutions to industrial condition monitoring and potential shortcomings.
- Investigate solutions for cognitive overload and user interface design techniques that support cognitive load reduction.

[9] https://www.windpowerengineering.com/business-news-projects/featured/condition-monitoring-made-easy-to-read/ [Accessed September 2018]

- Investigate framework development techniques and design principles that can assist in the development of a customisable and configurable software system.

Main study objectives

- Develop a system that reduces the cognitive load on management level staff by condition monitoring data.
- Investigate if the system with remote accessibility and reduced cognitive load can reduce maintenance reaction times.
- Implement the system on a large multi-operational industrial organisation with as little configuration time as possible.

1.7 Methodology

Figure 3 presents the methodology with which the study aims to achieve the objectives defined in Section 1.6. A targeted literature review is conducted, which focuses on the literature review objectives defined in Section 1.6. From the literature, a set of system requirements is formulated. A detailed design of a system consisting of a front-end subsystem, a relational database and a configuration tool then commences. The design is verified regarding each of the defined requirements. Once verified, the system is implemented on a large multi-operational industrial organisation. The implementation is investigated regarding each of the main study objectives to validate if the objectives have been met.

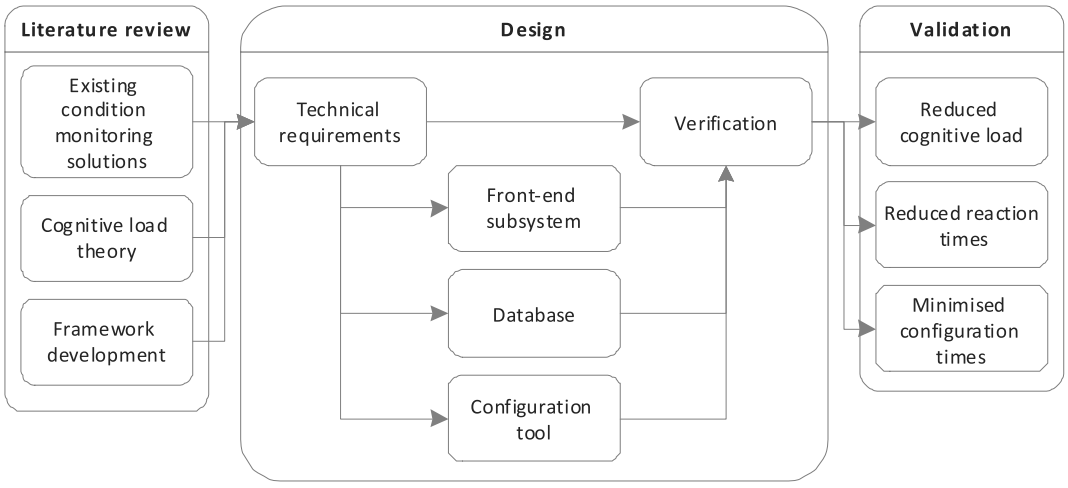


Figure 3: The study methodology

1.8 Outline

This section provides an overview of the rest of the document.

Chapter 2 – Literature review

Presents a comprehensive review conducted on literature regarding existing work done and potential solutions to the problems identified and objectives stated in this chapter. Section 2.2 investigates existing solutions for industrial condition monitoring. Section 2.3 investigates cognitive load theory and user interface design principles to minimise cognitive load and to manage user focus. Section 2.4 reviews the advantages of modular software design.

Chapter 3 – Design overview

Gives an overview of the system design, which is aimed at solving the identified problem. The design is based on the conclusions drawn from Chapter 2. Section 3.2 formulates requirements for the system. Section 3.3 gives background of the development context and resources. Section 3.4 gives an overview of the basic operating principles and the system's integration with existing infrastructure.

Chapter 4 – Detailed design

Describes the various elements of the system design, aimed at solving the identified problem in detail. Section 4.2 presents the front-end system functionality and features. Section 4.3 describes the relational database design. Section 4.4 describes the functionality and features of the configuration tool. Section 4.5 gives an overview of the logic flow of the system. Section 4.6 verifies that the system meets the requirements formulated in Section 3.2.

Chapter 5 – Results

Validates that the system solves the identified problem and that the system achieves the objectives of the study. Section 5.2 gives background on the implementation of the system. Section 5.3 provides background on the validation case study. Section 5.4 presents a case study where the system is implemented in a mining group. In this section, the system is evaluated in terms of each of the main study objectives.

Chapter 6 – Conclusions and recommendations

Chapter 6 concludes the study. Section 6.1 summarises the work done, discusses the case study results and concludes the success of the study. Section 6.2 identifies the shortcomings in the study and recommends future research and development work.

2 Literature review

2.1 Preamble

The chapter presents a literature review on concepts relevant to the problems and objectives mentioned in Chapter 1. The purpose of the literature review is to obtain knowledge of existing solutions to similar problems. These solutions can then be used to assist in the development of a solution to the problems addressed in this study.

Section 2.2 investigates existing condition monitoring solutions in industrial environments. Section 2.3 investigates cognitive load theory and user interface design principles to minimise cognitive load and to manage user focus. Section 2.4 reviews the advantages of framework development and modular software design.

Section 2.5 summarises the chapter and draws a conclusion from the literature review.

2.2 Existing solutions to industrial condition monitoring

2.2.1 Introduction to the aspects of condition monitoring

In his PhD thesis, Van Jaarsveld divided the condition monitoring process into three key steps. The first step is data acquisition and preparation, which is followed by operational condition assessment and, finally, information and exception reporting [18].

According to NRG Systems, a leading company in the renewable energy industry, the process used by all condition monitoring systems to convert a physical phenomenon into a recommendation to the applicable stakeholders is very similar. This process can be broken down into six basic steps, namely, data acquisition, data processing, fault detection, fault diagnosis, prognosis and, finally, the recommendation.[10]

Fault detection, fault diagnosis and prognosis are processes that analyse the historical data to determine the nature of equipment faults. These processes are therefore advanced versions of data processing. The recommendation step is part of information transfer from the condition monitoring system of applicable stakeholders. NRG System's condition monitoring philosophy can therefore be condensed into data acquisition, data processing and information transfer. This aligns with the three steps identified by Van Jaarsveld [18]. This study will therefore focus on the work done on these three aspects of condition monitoring.

[10] https://www.nrgsystems.com/assets/resources/Understanding-Wind-Turbine-Condition-Monitoring-Systems-Whitepaper.pdf [Accessed March 2018]

Data acquisition consists of the process and the infrastructure needed to convert a physical phenomenon, such as temperature, vibration or a crack, into measured digital values and to store these digital measurements. Measurements are typically done with analogue sensors (producing outputs such as voltage), which are then converted into more meaningful digital measurements (such as °C). Work done on data acquisition in condition monitoring is discussed in further detail in Section 2.2.2.

Data processing consists of all data analysis processes that convert large volumes of data into simpler values, for example, health indicators for a component or an estimate of a component's remaining life. With enough information, data processing can also entail identifying the cause of equipment malfunction and generating recommendations on fixing the equipment malfunction. Work done on data processing in condition monitoring is discussed in further detail in Section 2.2.3.

Information transfer or data visualisation is the way that processed data or information is presented to the applicable stakeholders. This can be achieved through automated SMSs,[11] emails, reports or software. The setup and configuration of such automated notification and visualisation systems can also be considered part of the data visualisation aspect of condition monitoring. Work done on data visualisation in condition monitoring is discussed in further detail in Section 2.2.4.

2.2.2 Data acquisition

This subsection gives an overview of the data acquisition process in a condition monitoring system. Section 2.2.2.1 discusses the typical measurements needed for condition monitoring purposes. Section 2.2.2.2 gives an overview of infrastructure needed to acquire this data and the process with which it is achieved.

2.2.2.1 Parameters and measurements

In order to determine the mechanical condition of industrial equipment or components, knowledge of the physical behaviour of the equipment is required. Various parameters can be measured to achieve this. The applicable parameters for determining the mechanical condition of industrial equipment may vary depending on industry, infrastructure, available resources and application.

The most commonly used parameters for determining the mechanical health of industrial equipment are temperature and vibration. Most pieces of industrial equipment have at least

[11] Short message service

one moving part during operation causing friction, heat build-up and vibration. These are also scalar measurements, which make analysis simpler [19].

Moving components, and especially rotating components, produce vibrations throughout a machine. Changes in the magnitude and frequency of these vibrations can be indicative of changes in component balance, component wear or a need for lubrication. Vibration analysis is thus a powerful tool for early fault detection in machinery. This statement is supported by the fact that numerous studies utilise vibration analysis for fault detection and diagnoses in various applications such as internal combustion engines, rolling element bearings and compressors [20]–[22].

Using vibration as an indicator of component mechanical health can be expensive. Vibration is typically measured at the bearings. Using an accelerometer is the most common method for measuring vibration [23]. These measurements are required at virtually all the bearings in the component to be meaningful. Furthermore, industrial equipment typically consists of a multitude of bearing elements. Sensor costs can quickly become expensive when multiple pieces of equipment or even operations are considered.

Temperature is an alternative to vibration as a component health identifier, which opens up cheaper measurements options to determine the mechanical health of a component [15], [24]. Temperature is a direct result of friction between moving parts. Changes in the friction forces between two moving parts will reflect in the temperature of the parts. Deviations in the operating temperatures of components can be an indication of a fault. Temperature is a flexible indicator of the mechanical component health in terms of the level of detail it provides as well as cost. For a detailed result set, sensors can be installed at all bearing elements. For a simpler result set at reduced costs, sensors can simply measure gearbox or oil temperatures [7], [24].

Temperature can be measured with a variety of methods. Thermistors, thermocouples and infrared thermography are some examples of such measurement sensors and techniques[12] [15], [25]. The best-suited measurement technique depends on its application and the budget. Thermistors are temperature-dependent resistors that have non-linear behaviour. Thermistors are therefore better suited for small temperature ranges.[13] Thermocouples generate a small voltage when subjected to higher temperatures. Thermocouples are typically used at higher temperatures and for larger ranges.[12] With infrared thermography, a thermal camera is pointed at the component during operation. The thermal camera produces a colour-coded thermal

[12] https://www.allaboutcircuits.com/technical-articles/introduction-temperature-sensors-thermistors-thermocouples-thermometer-ic/ [Accessed August 2018]

[13] https://www.maximintegrated.com/en/app-notes/index.mvp/id/817 [Accessed August 2018]

map of the component. This technique reduces the number of sensors while producing a detailed result set of the entire component's temperature.

Figure 4 illustrates a thermal camera and gives an example of a thermal map of a computer numerical control (CNC) milling machine. The drawback of using temperature as a health indicator is that it is not as immediate as vibration analysis. The monitored machine needs to operate stably for a while before the temperatures can indicate a developing fault.

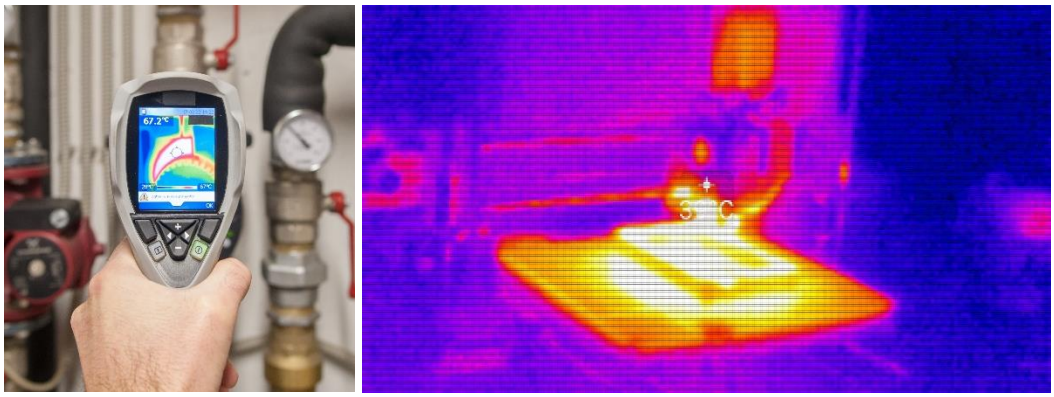


Figure 4: Left: A handheld thermal camera;[14] Right: An infrared thermal image of a CNC machine[15]

Vibration and temperature are not the only measurements that are useful for condition monitoring of industrial equipment. Oil debris content can also be a good indicator of grinding gears or mechanical parts [13]. However, depending on the type of machinery, practical real-time measurements may be challenging.

Motor current signal analysis is another method for detecting faults in rotating machinery that drive electric motors. Faults in machines, such as a defective bearing, cause deviations in motor torque and instantaneous angular speed [26].

2.2.2.2 Basic process and infrastructure

Section 2.2.2.1 mentioned numerous methods for measuring physical phenomena that can indicate the mechanical health of components. The sensors responsible for these measurements typically produce analogue outputs such as resistance or voltage.[16] These analogue values need to be converted to digital values using a digital converter. In some

14 © mazzamazza / Adobe Stock

15 © Science Photo / Adobe Stock

16 https://www.allaboutcircuits.com/technical-articles/introduction-temperature-sensors-thermistors-thermocouples-thermometer-ic/ [Accessed August 2018]

cases, such as accelerometers, sensor output voltages are too small for digital converters and require amplification first[17] [27].

Digital sensor data is transmitted to an on-site supervisory control and data acquisition (SCADA) system. SCADA systems are common in industrial organisations as they enable automated control and data logging of industrial equipment.[18] The SCADA system serves as a central data storage unit for a site. Data is stored on the SCADA system in the form of log files [18], [28]. SCADA systems are typically configured to display the real-time measurement data to control room operators. Even though SCADA systems are typically programmed to mitigate equipment issues through automated control, control room operators can intervene and take manual control [18].

Large industrial organisations typically consist of numerous operations, each with their own SCADA systems. Directly installing condition monitoring software on each of these SCADA systems makes software maintenance and optimal setup of the condition monitoring system impractical. To simplify data processing and software maintenance, data storage needs to be centralised. An open platform communications (OPC) server can access the SCADA data and transmit data via a mobile network. A centralised data storage server receives the data from the OPC servers and translates the data to a desirable format [18], [28].

Once data has been transferred to the centralised storage facility, the condition monitoring system can access the data for processing. Data processing for condition monitoring purposes is discussed in further detail in Section 2.2.3

2.2.3 Data processing

This subsection gives an overview of the work done on the processing of condition monitoring data. Section 2.2.3.1 discusses fault detection from condition monitoring data and the diagnosis of these detected faults. Section 2.2.3.2 discusses the work done on predicting the remaining useful life of equipment.

2.2.3.1 Fault detection and diagnoses

The first and most important feature of a condition monitoring system is fault detection. All other functions of a condition monitoring system are based on fault detection. Without fault detection, equipment will run until failure, which defeats the purpose of a condition monitoring system. The system cannot diagnose a fault that it had not detected and the remaining life of

[17] https://www.plantservices.com/assets/wp_downloads/MeasuringDisplacementUsingAccelerometers.pdf [Accessed August 2018]

[18] https://inductiveautomation.com/what-is-scada [Accessed August 2018]

the piece of equipment cannot be estimated accurately if all information regarding equipment faults is not available. It is therefore imperative that a condition monitoring system can detect equipment faults with a high degree of accuracy and reliability.

A study conducted on the maintenance of computer-integrated manufacturing proposes that the mean-time-between-failure of manufacturing machinery can theoretically be extended indefinitely. To achieve this, development of more accurate and reliable machine performance monitoring, fault detection and fault diagnostics is required. A machine performance or condition monitoring system that can detect faults with 100% accuracy and reliability will give maintenance personnel ample time to investigate issues, correct faults, and extend machine life [1].

Unfortunately, development of a system that can detect faults with 100% accuracy and reliability is only a theoretical concept. In practice, one can only thrive to achieve a fault detection accuracy as high as the available resources and that current technology and research allow. Consequently, numerous studies have investigated various fault detection methods in a search of improved fault detection accuracy.

In a conference on the *Advances in Condition Monitoring of Machinery in Non-stationary Operations*, a paper investigated the accuracy and repeatability of various supervised classification methods for vibration analysis of rolling element bearings. The study concluded that simple processing methods, such as direct frequency and time-domain analyses, can accurately and reliably detect faults and indicate mechanical health of rolling element bearings. However, these methods are not suitable for diagnosing and determining the root causes of failures. Advanced data processing methods, such as narrowband envelope analysis, modulation intensity distribution, and integrated modulation intensity distribution, are better suited for diagnostic purposes [20].

Another paper from this conference analysed the SCADA temperature data of a belt conveyer system used for copper ore transportation. Due to the significant variance in system temperature with time (due to mining schedules), immediate analysis of temperature data is inaccurate. Instead, an unsupervised learning method was developed to consider longer term data. System temperatures typically form a pattern over a one-week period. Therefore, anomalies in this pattern is an indication of a developing fault [24].

A study on fault detection in gears used stochastic resonance as an advance condition monitoring method for gears. The accuracy of existing fault detection methods was improved by reducing the number of false alarms without reducing the number of correctly identified

faults. The study reduced the complexity of raw signal data with two different strategies. The first strategy involved selective filtering of certain frequencies. The second strategy was to use a high pass filter at a strategic frequency [21].

An article published in the journal *Mechanical Systems and Signal Processing* investigated methods of classifying reciprocating compressors. The paper compared linear discriminant analysis, neural networks, support vector machines (SVMs) and extreme learning machines. It was found that non-linear classifiers (neural networks trained by Bayesian regularisation, non-linear SVMs, and extreme learning machines) performed better than their linear counter-parts and could solve more complex problems. Large volumes of historical vibration data were available for this analysis. The accuracy of machine learning techniques would reduce greatly if limited or no historical data is available [22].

Another study developed a method for high precision classification and diagnosis of bearing faults using vibration analysis. The method combined autoregressive modelling (ARM), linear discriminant analysis (LDA) and an SVM. This complex method used ARM to classify all the features in the raw vibration signal. The LDA filtered out features that were not applicable to faults. The results from the LDA were fed into the SVM to determine the state of the bearing [29].

2.2.3.2 Prognosis

Prognosis form part of advanced condition monitoring. Prognosis is a prediction of the future health of components until failure. The first requirement of prognosis is to diagnose faults accurately. If the condition monitoring system can identify a fault accurately and diagnose the severity of the fault, the system can attempt to realistically estimate the future mechanical health.[19] The operating conditions can also have a significant impact on the equipment wear and health deterioration. Historical data can give insight into the operating conditions of the equipment and is therefore a useful supplement for prognosis [8].

Prognosis is a powerful tool to assist with proactive maintenance. If a condition monitoring system can accurately predict the remaining life of equipment, maintenance can be done in a timeous manner without equipment failure occurring. Prognosis of industrial machinery is not a new concept and has been successful as far back as 2004 [8]. Yan, Koç and Lee used an autoregressive-moving-average model in conjunction with a maximum-likelihood technique to extrapolate historical data and determine the ideal window for maintenance [8].

[19] https://www.nrgsystems.com/assets/resources/Understanding-Wind-Turbine-Condition-Monitoring-Systems-Whitepaper.pdf [Accessed August 2018]

Accuracy is, a challenging area in prognosis, especially in complex machinery such as wind turbines. Prognosis in the wind turbine industry is predominantly done using vibration analysis with the occasional temperature data analysis. The study by Nie and Wang recommends that techniques such as noise inspection and oil quality monitoring are incorporated in prognosis [30].

Noise in condition monitoring signals is one of the main contributing factors to the accuracy of prognostic methods. With enough noise in vibration signals, remaining useful life prediction can become completely infeasible, especially when using the popular conventional Bayesian method. A robust method has been developed to mitigate this issue. This method uses a constrained Kalman filtering approach to try and predict the future vibration behaviour. This method has proven to be a significant improvement over the conventional Bayesian method when a significant amount of signal noise is present [31].

2.2.4 Information transfer and data visualisation

Very little academic work focuses on information transfer and the visualisation of condition monitoring data. These aspects mostly form part of the design process during the development of condition monitoring systems. This subsection therefore gives an overview of the information transfer philosophies of existing condition monitoring systems. Section 2.2.4.1 describes the platforms on which various condition monitoring systems operate. Section 2.2.4.2 analyses the way in which condition monitoring systems present data and information to stakeholders.

2.2.4.1 Platforms

Section 1.3 motivated the benefits of providing condition monitoring information to a wide range of stakeholders. These stakeholders operate from a range of physical locations. Most condition monitoring systems therefore incorporate some form of remote access to enable stakeholders to access this condition monitoring information. This is typically achieved through either client-server applications or web applications.

Client-server applications are installed on on-site computers that directly connect to the site's networks. Condition monitoring systems that use this approach can only be accessed through the computer on which it is installed.[20] Condition monitoring systems that integrate directly with SCADA or control room systems are typical client-server systems [7], [24], [32], [33].

[20] http://www.differencebetween.net/technology/software-technology/difference-between-client-server-application-and-web-application/ [Accessed September 2018]

The advantages of client-server applications are mainly associated with performance. Data processing is streamlined since the data is available directly on the same computer. The system is also only responsible for processing a smallish data set (only one site's data). Furthermore, these systems tend to be more user-friendly due to the flexibility of window- and form-based applications.[21, 22]

The disadvantages of client-server applications are mainly associated with software maintenance and distribution. Software updates are used for fixing bugs, maintaining compatibility and delivering new features. Having software installed on numerous widespread networks and computers makes software updates extremely challenging. In these applications, software updates need to either be installed manually from a storage device or downloaded automatically from a remote server. Manual installation from a storage device can quickly become a logistical nightmare, while automatic downloads create other challenges such as server overload [16].

Web applications are installed on a remote server with no direct connection to any site's network. Condition monitoring systems that use this approach can only be accessed with an active internet connection through a web browser. Typical examples of such condition monitoring systems are NRG Systems' TurbinePhD™[23, 24] and GS&S DasBox condition monitoring system.[25, 26]

Web applications offer numerous advantages over client-server applications. The first major advantage is ease of software maintenance and distribution. A single version of the application is managed and maintained. Software updates to the server's application are available immediately to all users of the software. This ensures that all users are running the latest and most stable version of the application [16].

Ease of access is the second major advantage that web applications have over client-server applications. Web applications can be accessed from anywhere in the world if an internet connection is available. The software is not installed on the user's computer. This means that

21 https://www.cogz.com/articles/web-based-cmms-systems-compared [Accessed September 2018]

22 http://www.sigmaplot.co.uk/products/SIGMA_CERF/BrowserClient.pdf [Accessed September 2018]

23 https://www.windpowerengineering.com/business-news-projects/featured/condition-monitoring-made-easy-to-read/ [Accessed September 2018]

24 https://www.windpowerengineering.com/operations-maintenance/condition-monitoring-maintenance/how-is-condition-monitoring-changing-in-the-wind-industry/ [Accessed September 2018]

25 https://www.slideshare.net/ElenaMariaVaccher/plant-maintenance-condition-monitoring [Accessed September 2018]

26 https://www.slideshare.net/ElenaMariaVaccher/gss-services-for-industries-2016 [Accessed September 2018]

the software is available immediately to the user and the user's computer requires little computing power. This leads to web applications often being cheaper to operate.[27]

The disadvantages of web applications are mainly reduced performance and flexibility. A web-based condition monitoring system does not have direct access to data. Data needs to be transferred from the site to the server before data processing can commence. Furthermore, the server carries a much greater load as it must process data for all sites and the organisation[28] [18].

Web applications can only be accessed through a web browser and requires an internet connection. Some remote locations do not have an internet connection and can therefore not access web applications. Browser-based user interfaces also tend to be limited in flexibility compared with window- and form-based equivalents.[29]

2.2.4.2 Visualisation

Condition monitoring systems measure, log and process large volumes of data to produce large volumes of information. To use a condition monitoring system successfully, this information needs to be transferred effectively to all applicable stakeholders. Information transfer is mainly done through some form of data visualisation on the user interface.

The most common raw data in condition monitoring systems is in the time domain. Some systems that use vibration analysis have data in the frequency domain. Interpretation of both time-domain and frequency-domain data is easy when presented in the form of line graphs. Most condition monitoring systems therefore use line graphs to present raw data[30, 31] [7]. Figure 5 gives an example of line graphs used to present vibration data.

27 https://www.vstech.com/platforms/client-server-web-based/ [Accessed September 2018]

28 http://www.sigmaplot.co.uk/products/SIGMA_CERF/BrowserClient.pdf [Accessed September 2018]

29 https://www.cogz.com/articles/web-based-cmms-systems-compared [Accessed September 2018]

30 https://www.windpowerengineering.com/business-news-projects/featured/condition-monitoring-made-easy-to-read/ [Accessed September 2018]

31 https://www.slideshare.net/ElenaMariaVaccher/plant-maintenance-condition-monitoring [Accessed September 2018]

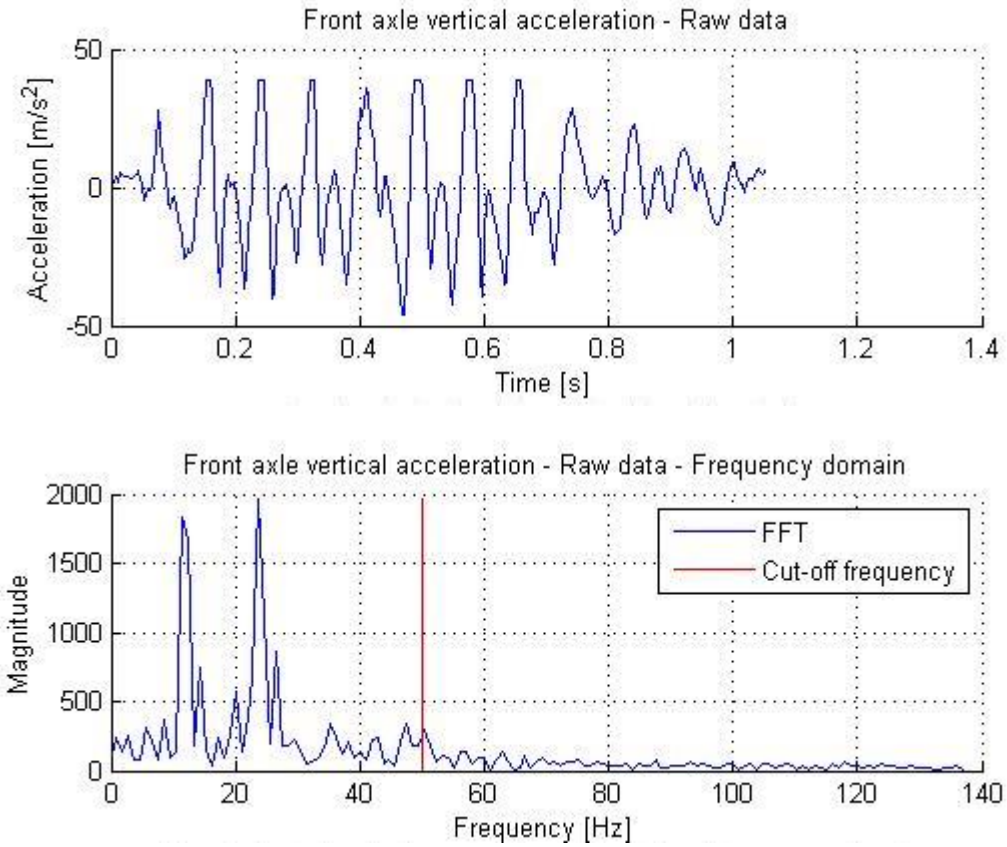


Figure 5: Top: Example of time-domain vibration data; Bottom: Example of corresponding frequency-domain vibration data [34]

Condition monitoring systems often incorporate component layout diagrams to display real-time measurements. This makes it easier for the user to interpret the meaning of a value by associating a value with a visual component. Component layout diagrams are common in control and SCADA software and, by extension, common in the condition monitoring system that integrates directly with these control and SCADA systems[32, 33] [18]. Figure 6 presents typical examples of such component layout diagrams.

[32] https://www.windpowerengineering.com/operations-maintenance/condition-monitoring-maintenance/how-is-condition-monitoring-changing-in-the-wind-industry/ [Accessed September 2018]

[33] https://www.slideshare.net/ElenaMariaVaccher/plant-maintenance-condition-monitoring [Accessed September 2018]

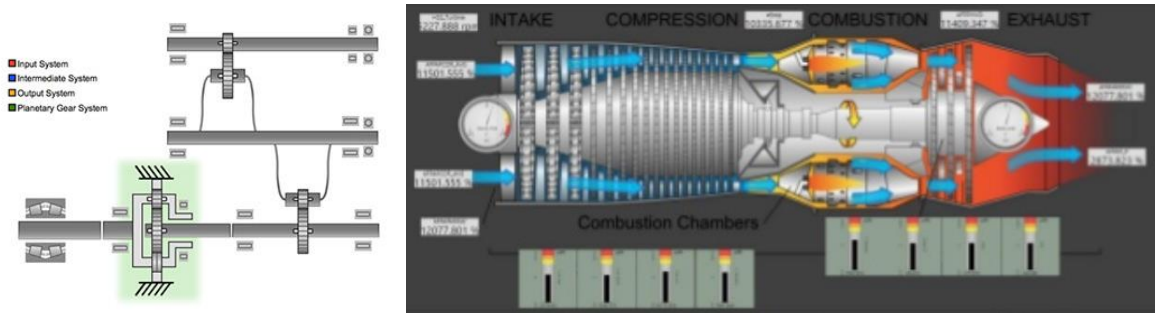


Figure 6: Examples of component layout diagrams in condition monitoring systems[34, 35]

Column graphs are useful for displaying categorised data. Typical examples of data categories in industrial condition monitoring are sites, components and parameters. The number of incidents or the average health index can then be plotted per category to isolate problem areas.[34] Figure 7 gives an example of a useful application for a column graph in a condition monitoring system.

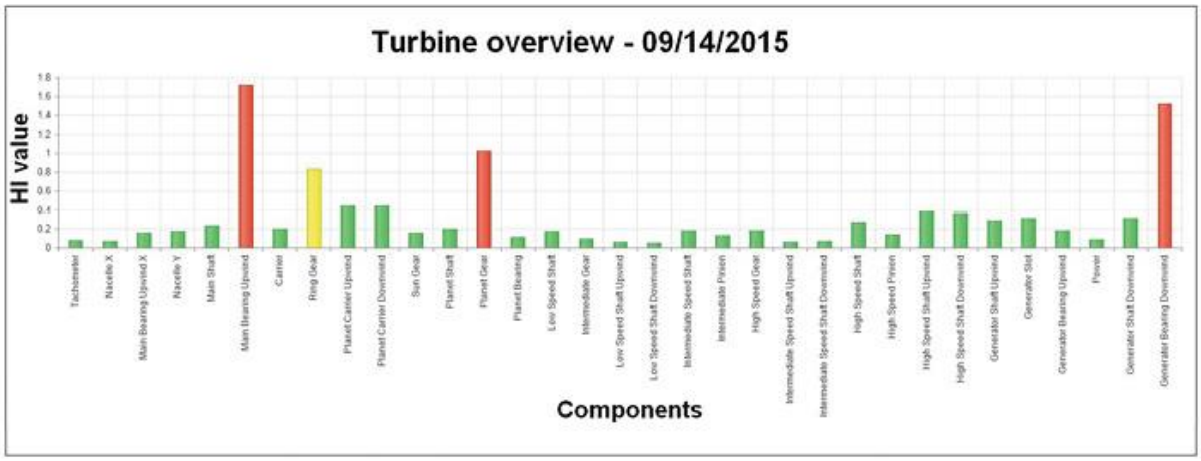


Figure 7: Example of an application of a column graph in a condition monitoring system [34]

Despite the value they offer, organisational and operational overview levels are rare in condition monitoring systems. TurbinePhD™ offers such functionality in the form of a fleet view. The fleet view summarises the overall health status of an organisation's wind farms in the form of bar graphs. Each wind farm is represented by a bar consisting of three colours. Green represents the number of wind turbines that operate according to without any issues. Yellow represents the number of wind turbines that have minor issues that may soon require

34 https://www.windpowerengineering.com/business-news-projects/featured/condition-monitoring-made-easy-to-read/ [Accessed September 2018]

35 https://www.slideshare.net/ElenaMariaVaccher/plant-maintenance-condition-monitoring [Accessed September 2018]

maintenance work. Red represents the number of wind turbines with critical issues that require immediate attention.[36] Figure 8 presents the basic layout of the TurbinePhD™ fleet view.

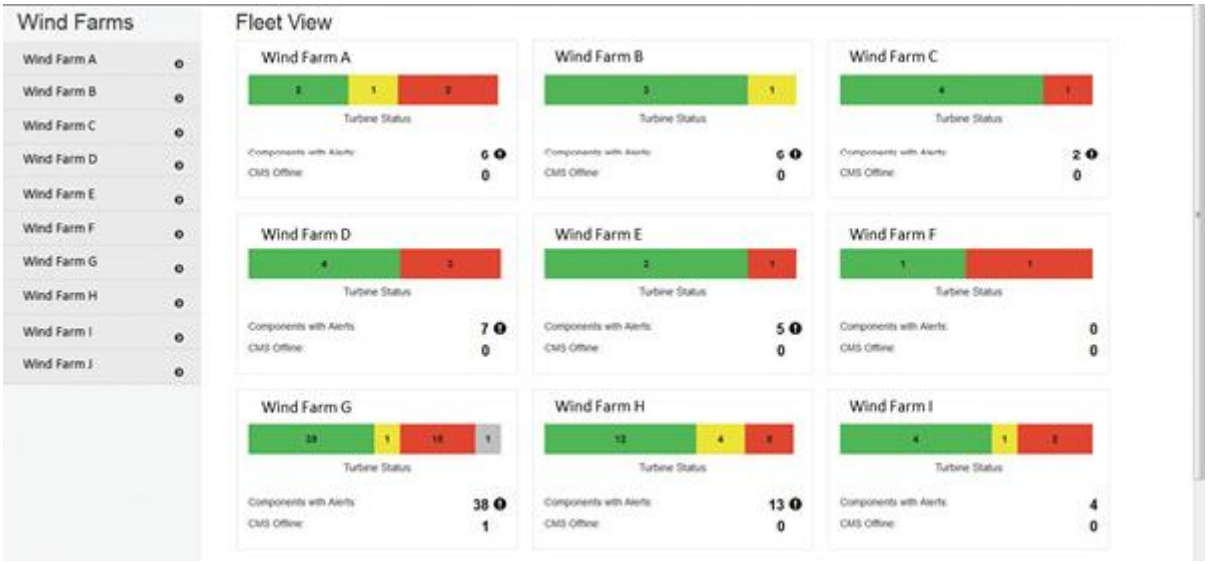


Figure 8: The basic layout of the TurbinePhD™ fleet view[36]

Some stakeholders have numerous responsibilities and are not necessarily able to constantly monitor condition monitoring systems. This means that reactions to critical issues can be delayed or even worse, that critical issues go unnoticed. To mitigate this risk, some condition monitoring systems incorporate automated notifications, such as warning SMSs and exception reports sent via email, to the applicable stakeholder. These notifications are typically only sent in the case of emergency situations to avoid spamming stakeholders and losing significance [7], [18].

2.3 Effective user interface design

2.3.1 Cognitive load theory

Chapter 1 defined cognitive load theory and described how it applies to a condition monitoring system. This section focuses on the work that has been done to determine the causes of cognitive overload and how to avoid or at least mitigate it.

Cognitive load consists of a causal and an assessment dimension. The causal dimension is concerned with the interaction between the task itself and the individual. The difficulty of the task and the individual's ability is part of this dimension. The assessment dimension is

[36] https://www.windpowerengineering.com/business-news-projects/featured/condition-monitoring-made-easy-to-read/ [Accessed September 2018]

concerned with the measurable concepts of cognitive load. These are the mental load, the mental effort and the performance with which the task is executed [35], [36].

Measurement of this multidimensional construct is proving to be difficult. Since cognitive load depends on numerous subjective factors, measurements are always somewhat subjective or based on subjective data. Researchers have followed various approaches to quantify cognitive load. Some of these approaches model cognitive load based on mental effort, mental load and performance measurements. Other approaches use analytical and empirical methods that estimate the mental load by collecting subjective data and combining it with expert opinion [35]–[38].

This study focuses on industrial environments and the cognitive load associated with maintaining the equipment in these environments. An individual's ability varies greatly depending on his/her position in the company. The individual's workload, and thus task difficulty and mental load, typically scales with his/her ability. The general maintenance managers or chief operating officers of an organisation typically deal with much larger volumes of data than floor operators do. This concept has been alluded to in Section 1.4.

The purpose of this study is not to challenge stakeholders into mastering a skill, but rather to improve performance. Instructional condition efficiency (*E)* is a means to present the relationship between the mental effort ($Z_{Mental\ Effort}$) and performance ($Z_{Performance}$) of an individual for a given instruction. Instructional condition efficiency is calculated using Equation 1. High performance with low mental effort requirements is considered high instructional efficiency, whereas low performance with high mental effort requirements is considered low instructional efficiency. Figure 9 demonstrates this concept. For this study, the aim is to achieve a high instructional efficiency so that personnel can employ as much of their mental efforts on other matters [35].

Equation 1 Instructional condition efficiency formula [35]

$$E = \frac{Z_{Performance} - Z_{Mental\ Effort}}{\sqrt{2}}$$

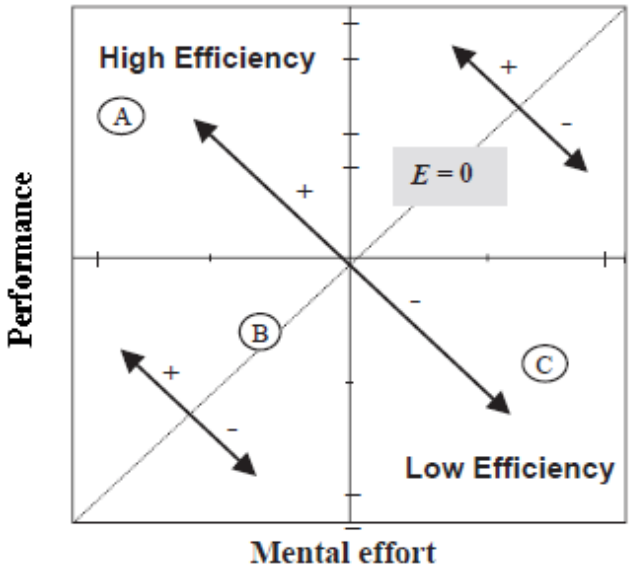


Figure 9: Graphic presentation used to visualise instructional efficiency [35]

Ideally, the cognitive load on the various personnel levels of operational experience needs to be quantified. This can be achieved by conducting numerous experiments and constructing analytical or empirical models. The numerous solutions can then be implemented to determine a solution that reduces cognitive load the most across all levels of personnel.

Unfortunately, gathering the data needed to effectively determine the cognitive load at various personnel levels is not practical. Stakeholders across multiple organisations would have to sacrifice precious time to participate in experiments or surveys. The subjective nature of cognitive load measurements can lead to significant inconsistencies across levels of personnel and organisations. The expertise needed to quantify cognitive load accurately and reliably places further emphasis on the impracticality of such an approach [35].

A simpler, more suitable solution is to employ as much cognitive load reduction techniques as possible to ensure high instructional efficiency and to prevent cognitive overload. Numerous methods have proved to reduce cognitive load in multimedia learning or education. The goals of a multimedia learning environment are similar to a condition monitoring information system in the sense that multimedia-based education tries to convey as much as possible information as efficiently as possible. Methods such as segmenting, pretraining, weeding, signalling, aligning words and pictures, and individualisation are all applicable to this study [39].

Methods such as off-loading, eliminating redundancy and synchronising are applicable to cognitive overload scenarios where both the visual and the auditory senses are involved. The main concern in terms of cognitive load management exists at management level in organisations. At this employment level, personnel typically operate in an office environment;

therefore, supplementing visual information with auditory information may be unsuitable. For this reason, off-loading, eliminating redundancy and synchronising will not be considered for this study [39].

Segmenting is a technique that can be used if large volumes of information are presented in succession. By breaking information down into smaller chunks, less information is processed at a given time. This technique extends the time taken to convey information, but the information is conveyed more effectively [39]. Pretraining is a similar technique that can be used when a great deal of new information needs to be conveyed. Pretraining involves giving a short overview of the basic concepts before introducing the main piece of information [39].

Weeding can be used when the presented information contains both essential and non-essential information to an extent that cognitive overload occurs. Weeding focuses on presenting concise information. Interesting additional information is discarded to emphasise essential information and to free up cognitive load [38]. Signalling can also be used if a great deal of additional information is presented. Signalling guides cognitive processes by emphasising essential information. Pointing bright coloured arrows at images, underlining or using bold fonts for words in text and exclaiming words in speech are examples of signalling methods to guide cognitive processes [39], [40].

Alignment of words and their corresponding visual elements is another simple technique for reducing cognitive load. Descriptive words or phrases are often used to supplement visual elements and improve learning performance. This integration of visual elements and descriptive text introduces additional processing to match visual elements with their corresponding descriptions. This additional non-essential processing can be minimised by aligning words and visual elements [39], [41].

Individuals' ability to process information vary greatly. Spatial ability refers to an individual's ability to form and hold mental images. Individuals with low spatial ability need to engage more mental effort when concepts are presented simultaneously. These individuals tend to benefit more from information that is presented sequentially. However, individuals with high spatial ability benefit more from information that is presented simultaneously as they can then make better connections between the presented concepts [39], [42].

The visual working memory of an individual also depends on his/her spatial ability. Individuals with high spatial abilities can process and remember details about more objects presented to them simultaneously than individuals with low spatial ability. A simple guideline to accommodate most individuals is to only present three to five objects to a person at a time [43].

Individualisation can be used to present information in different ways to different levels of personnel. Personnel who operate at an organisational level are typically involved in numerous projects, operations or systems. This work requires an understanding of how these projects, operations and systems integrate to form a successful organisation. This understanding of the "big picture" is associated with a high spatial ability. Floor operators are typically not concerned with numerous operations or systems. They typically operate on a script-like schedule basis with little understanding of the big picture required. This is typically associated with a low spatial ability.

2.3.2 User interface design principles

User interfaces form a crucial part of any computerised system. Poor user interface design can lead to poor efficiency when performing tasks as well as to user dissatisfaction. This can hamper adoption of the system. It is therefore important to consider the design of the user interface from the early phases of development. The environment where the user interface will be used is also important to consider. The user interface must integrate well with the existing routine of the user [44].

For an organisation and its personnel to adopt a system, the user interface must be easy to use. The ease-of-use or intuitiveness of a user interface stems largely from the familiarity of the various aspects of the user interface. Aspects such as layout colours, fonts and icons of user interfaces have standardised, to a degree, over the years. By using these standardised aspects, the user can feel familiar with a new user interface without any prior training [43].

It should be noted that the industry norms for user interface design are not always optimal. Deviation from the standardised designs is acceptable and even encouraged in some cases. Some studies have argued that novelty and creativity in the field of user interface design are often hindered by the need for familiarity. By continuously implementing standardised designs, the suboptimal status quo is reinforced. Functionality should therefore get preference to familiarity. In cases where familiarity is sacrificed for functionality, the interface should be easy to learn rather than easy to use from the beginning [43], [45], [46].

A method to support the ease-of-use of a user interface, and especially for web-based user interfaces, is to provide a site map. Site maps provide a layout of the possible navigation paths together with the current location. Site maps help the user to get an overview of the

functionality that the system has to offer through the user interface. Common examples of site maps are tree views and breadcrumbs[37] [43].

The concept of flow describes the psychological state in which an individual is immersed and loses self-consciousness. Flow in online web sites has been linked to customer and user retention due to the enjoyable experience that the site offers. This positive association with a user interface can be a powerful tool for system adoption. Flow in a computer interface can be achieved in four ways: user skill and control, level of challenge and arousal, interactivity and telepresence, and focus and attention [47]–[49].

Using user skill and control to achieve flow implies that either the user needs to be naturally adept at using computer systems or the user requires training. This is not suitable to this study as the system will be used by a multitude of users with different skill sets. Furthermore, user training requires additional resources.

Increasing the **level of challenge and arousal** of the user is also not a suitable method for achieving flow in the user interface of this study. The level of challenge and arousal is often a useful way to encourage skill mastery or learning [35]. However, the purpose of a condition monitoring information system is not to challenge the user to learn about the information it provides. The purpose of such a system is rather to get the available information across to the user as effectively as possible.

Interactivity can be beneficial in a condition monitoring system's user interface, but not telepresence. A user interface with good interactivity will enable the user to navigate through the user interface without putting much thought into the navigation process itself. Interactivity is closely linked to the intuitiveness of the user interface. **Telepresence** occurs when the user becomes immersed in the experience to the point where he/she has the sense of being elsewhere.[38] This is typically an area of focus for video conferencing technologies or online shopping sites[39, 40][50]. In these environments, the user is using the interface for extended periods. The purpose of the condition monitoring information system is to provide information to the user as quickly as possible. Once the user has obtained the critical information from the system, he/she leaves the user interface to either act on the information or to continue with daily activities.

[37] A navigation aid that presents the user's location relative to home or main menu.

[38] https://www.dictionary.com/browse/telepresence?s=t [Accessed August 2018]

[39] https://whatis.techtarget.com/definition/telepresence [Accessed August 2018]

[40] http://www.video-conferencing.com/definition/telepresence.html [Accessed August 2018]

Focus and attention is the most applicable method for achieving flow in a condition monitoring system's user interface. The user needs to be made aware of potential faults as swiftly as possible. This can be achieved by drawing his/her attention and focus to these faults. User attention and focus can be manipulated with colour, size and placement [43].

Human attention is typically drawn to anomalies in patterns. A user interface that utilises a strict layout or pattern can therefore easily draw the user's focus by introducing a visual anomaly. A harmonious user interface with a constant pattern is also important to achieve a balanced visual experience. A user interface with a wide spectrum of colours, element sizes and little alignment is chaotic and can become boring. It becomes extremely difficult to control user focus in a chaotic user interface [43], [51]. Figure 10 illustrates the effectiveness of introducing anomalies into a harmonious pattern compared with a chaotic pattern.

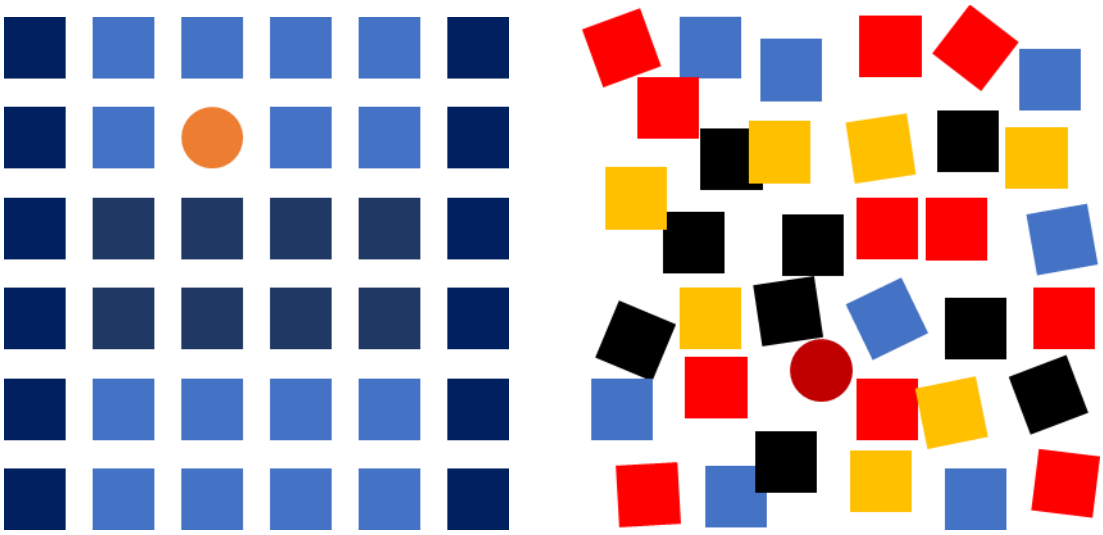


Figure 10: Introducing anomalies into a harmonious pattern versus a chaotic pattern

Colour forms a crucial part of user interface design. It can influence the mood of the user, it can guide a user's focus and if a business is well established, the business is often associated with the colour of its logo. A user interface that is launched as part of an existing company's offerings must incorporate the company's colour schemes. This emphasises the sense of familiarity with the company and the new system [51].

Colour can also be used to make an element stand out or become part of the background. Warm colours, such as red orange and yellow, tend to look closer to the human eye, while cool colours, such as blue and green, tend to recede.[41] Figure 11 demonstrates the effect that colour has on the perceived depth of elements. In the figure on the left, the small red squares

[41] https://www.thespruce.com/understanding-warm-and-cool-colors-1976480 [Accessed August 2018]

appear to be on top of the big blue square. In the figure on the right, the small blue squares appear to be holes in the big red square. Warm colours are also more energetic and vibrant. Red for instance, creates a sense of importance and a mood of alertness with the user. Cooler colours on the other hand are calm and reassuring. Green for instance, creates a sense of peace and stability with the user [51].

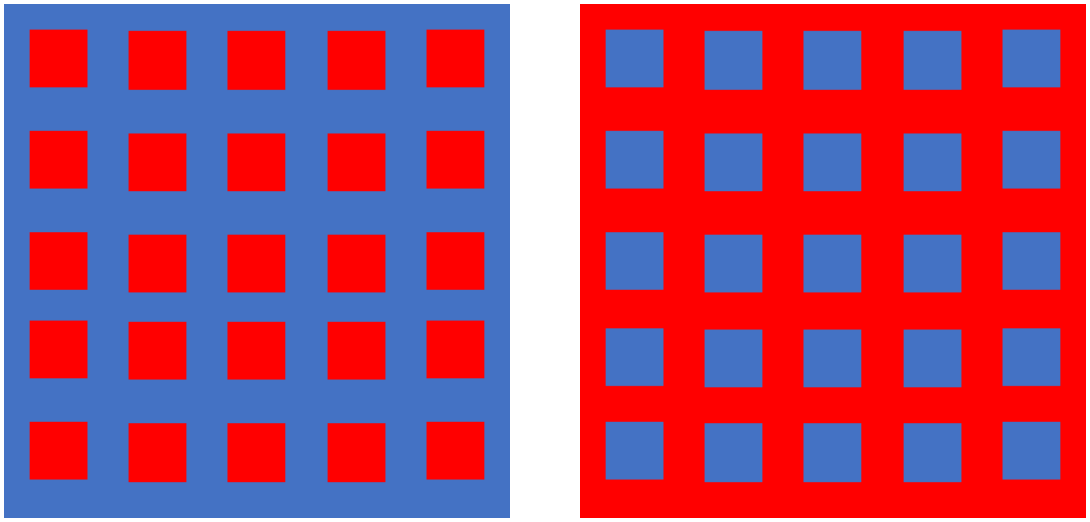


Figure 11: Demonstration of the effect of colour on depth perception

Element size is another way to manipulate the user's focus and attention. This applies to both text and shapes. The largest elements get the user's attention first. This only applies if most of the user interface is harmonious and consistent in terms of element size. Large elements are also most effective if kept simple. Complex shapes and paragraphs of text require more visual processing and are often glossed over [43]. Figure 12 demonstrates this by comparing a figure that has two larger elements with a figure that has six larger elements. The figure on the left does a much better job at drawing the user's attention and focus to the two large elements.

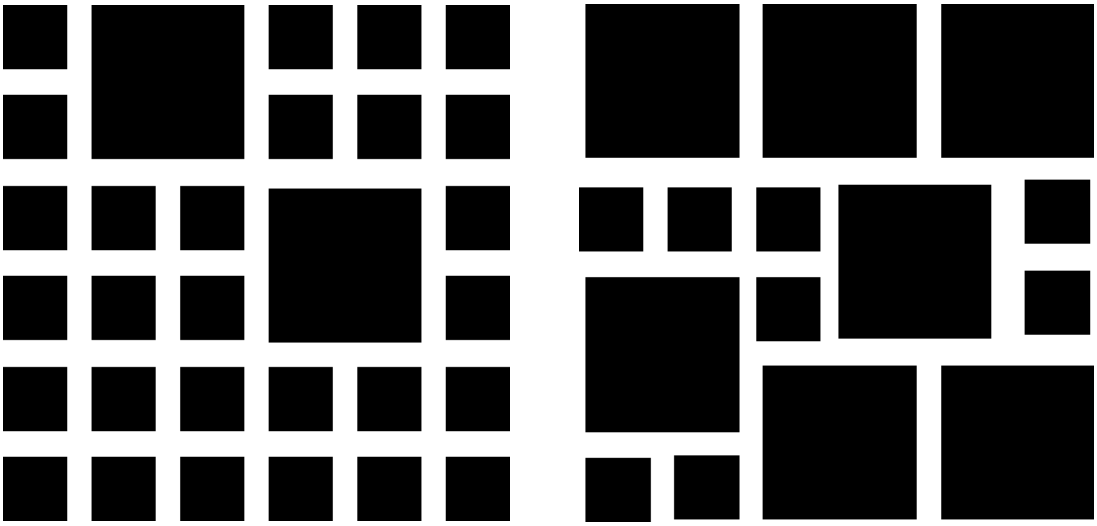


Figure 12: Demonstration of the effect of relative element size on user attention and focus

Element placement and alignment is the last method that will be discussed in this study to draw the user's attention. Most of the Western world read from left to right and top to bottom. This demographic will therefore typically attend to elements at the top-left first. If elements form a clear pattern, such as a line or row, and one element is slightly offset to break the pattern, it will draw the user's attention. Figure 13 demonstrates how a viewer's focus is drawn to the single misaligned element. An element that is out of pattern to draw the user's attention should still be within close proximity of the other elements that form the pattern. Stray elements can often go unnoticed [43].

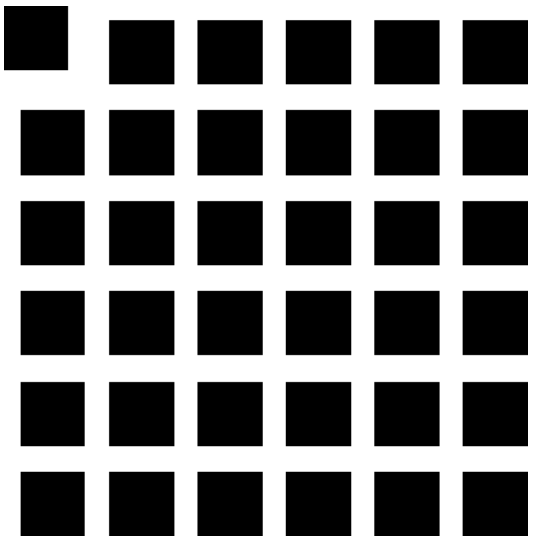


Figure 13: Demonstration of the effect of misalignment on user attention and focus

2.4 Framework development in a software environment

2.4.1 Software maintenance

Software maintenance is necessary to ensure that a system is compatible with new technologies, remains bug-free and, in doing so, continuously improves upon the existing system. Software maintenance forms a significant part of the life cycle of any software-based system. A lack of maintenance leads to a build-up of technical debt [16].

Technical debt is the term used to describe the code and resources required to improve poor quality software to acceptable or high-quality software. Technical debt is most often incurred when deadlines are prioritised over thorough design and testing. As a result, software bugs and poor designs are delivered as part of the final system, which requires development resources to fix at a later stage [16], [52]

Technical debt is also incurred when new temporary prototype functionality is added to an existing system. Prototype functionality often focuses on the visual possibilities for demonstration purposes. The code of such functionality is inherently underdesigned and often of poor quality. With continued emphasis on new and prototype functionality, maintenance on temporary prototype functionality is often neglected and technical debt becomes unrecoverable [16].

Despite the drawbacks of technical debt, it is often unavoidable. The majority of stakeholders involved in decision-making in organisations are often not fully aware of the long-term set-backs that technical debt can cause. For these stakeholders, new functionality and short deadlines are typically more important than well-developed software as they believe that the software can add immediate value to the organisation's offerings. Furthermore, development resources are usually in short supply no matter the size of the organisation. These factors contribute to technical debt build-up [16], [53].

Effective development of a software system requires a compromise between the development rate and technical debt build-up. Modular frameworks have proved an effective method of minimising technical debt while maintaining an acceptable development rate. This approach is very useful if a system is expected to expand rapidly and needs to be scalable [16], [54].

2.4.2 Frameworks and modularity

A software framework is software that offers generic functionality that can be reused for numerous purposes. This generic functionality can be overridden selectively to be more

suitable for a specific application.[42] Software frameworks are useful for developing multiple systems with similar functionality. Each individual system can heavily leverage off existing functionality in the framework. Minimal additional development is needed to meet the requirements of each individual unique system [16], [54].

Modularity in software refers to the way that separate smaller modules are combined to form a piece of software.[43] The main purpose of modularity is to provide options, or additional tools, to the software development process. Options and, by extension, modularity, can therefore only add value or have no impact; it cannot have a negative impact [55], [56]. A modular software framework is therefore a piece of software with a generic base functionality. This base functionality consists of numerous smaller modules to form the building blocks of the framework. These modules or features can selectively be used and combined to solve unique software problems. These modules and features can further be overridden to be custom-tailored for a unique solution. This set of characteristics enables modular software frameworks to be highly evolvable [56].

2.5 Summary

Condition monitoring of industrial equipment and machinery consists of three key phases, namely, data acquisition, data processing, and information transfer and visualisation.

Data acquisition comprises the processes that equipment needs to convert a physical phenomenon into digital data. Data acquisition can take many forms in terms of the parameters measured, the logging process and data transfer.

Data processing involves analysing raw data to produce useful information. Various academic studies focus on this aspect of condition monitoring, and numerous solutions are available for condition monitoring data processing. Simple condition monitoring solutions only identify faults in equipment and machinery. More advanced condition monitoring solutions diagnose faults and identify their root causes in equipment and machinery. Some of these advanced condition monitoring solutions do not only diagnose faults but also estimate the remaining useful life of equipment. This is called prognosis.

The information transfer and visualisation aspect refers to the way that condition monitoring systems inform stakeholders of the condition of equipment. Academic work on this aspect of condition monitoring is very limited compared with data processing. Therefore, the review

[42] https://www.techopedia.com/definition/14384/software-framework [Accessed August 2018]
[43] https://www.techopedia.com/definition/24772/modularity [Accessed August 2018]

focused on the functionality of existing condition monitoring systems rather than literature. Condition monitoring systems typically present raw data with simple line graphs. Most of these systems focus on component-level information without showing operational or organisational overview levels. NRG Systems' TurbinePhD™ does provide such functionality in the form of a fleet view, but this system is specialised and specific to the wind turbine industry.

The review on cognitive load theory identified six methods to reduce cognitive load in condition monitoring user interfaces. These methods are segmenting (present information in succession), pretraining (train users before they use the system), weeding (weed out non-essential information), signalling (emphasise essential information), aligning words and pictures, and individualisation (custom-tailor system to the user's ability).

User interface design literature suggests that familiarity should be emphasised to achieve an intuitive user interface, which can assist with system adoption. Various methods can be used to manage a user's focus and attention. User interfaces should be harmonious in terms of element patterns, colour, colour tone, size and alignment. Introducing anomalies into any of these attributes will draw a user's attention and focus.

Modular framework development is a proven method when developing scalable systems. Modular software frameworks minimise technical debt build-up while enabling rapid development. This simplifies software maintenance. New implementations of a modular framework reuse existing functionality without additional development. This reduces implementation time and in turn improves scalability. Customised functionality is simple to add to modular frameworks without affecting other implementations, making the system more adaptable.

The literature review therefore identified a lack of generic condition monitoring information and visualisation systems that reduce cognitive load satisfactorily. For the system to improve maintenance decision-making effectively, the system must be remotely accessible and avoid cognitive overload. Using a modular framework development approach will ensure that the system is scalable in terms of large organisations and numerous industries.

3 Design overview

3.1 Preamble

This chapter presents an overview of the design of the system, which aims to solve the problem statement identified in Chapter 1. Section 3.2 describes the technical requirements that have been formulated for the system. Section 3.3 provides background on the resources available for the development of a new condition monitoring system. Section 3.4 presents the overview of the system functionality and the manner in which it integrates with existing resources.

3.2 Requirements

3.2.1 Accessibility

The system will be used by a wide range of stakeholders with varying job descriptions. Some of these stakeholders may operate in off-site offices or travel on a regular basis. System information should therefore be accessible remotely.

3.2.2 Integration

Implementing a new system can be resource-intensive. On-site data logging infrastructure, servers for running the system and user access control, to name but a few, are some of the challenges faced during the implementation of a new system. In order to streamline implementation, the system should utilise as many available resources and much of the infrastructure as possible.

3.2.3 Information

The purpose of the system is to provide information to the user. The system should therefore provide the user with access to the raw logged data as well as the processed data. Data and information should be provided in a condensed and logical way for simple interpretation.

3.2.4 Multilevel

Existing condition monitoring solutions and systems place a large emphasis on parameter-level detail. Existing condition monitoring systems lack organisational level information. The system should improve upon existing solutions by providing multiple levels of detail including an organisational overview and raw parameter data detail.

3.2.5 Access control

The various stakeholders who will use the system have different positions in the organisation. The information to which stakeholders have access to is determined by their positions in the

organisation. Thus, the system should limit users to only the information applicable to their positions.

3.2.6 Configurability

The system is intended for implementation in multiple organisations and on numerous operations. Business structures and the types of equipment used vary between organisations. The system should be developed in such a way that it can be implemented in various organisations without additional development time required during rollout. The system should be configurable through database changes only. These database changes should be made through a customised user interface, which improves database safety and enables personnel without the necessary development skills to configure the system.

3.2.7 Adaptability

The system will serve as a framework for visualising condition monitoring data in various industries. The equipment used in these industries and, by extension, the techniques to determine their mechanical health, varies. The techniques used for determining equipment health may vary in terms of data acquisition and data processing. To ensure that the system can be used across a wide range of industries, it should thus be adaptable in terms of data acquisition and data processing.

Data acquisition consists of measuring a physical phenomenon and logging the data up to the point of data storage. The system should thus accommodate data from various phenomenon, unit and measurement types.

Data processing consists mainly of analysing raw data. The analysis can be performed with the intention of identifying a potential fault, diagnosing existing faults or even predicting remaining equipment life. The purpose of data processing and the algorithms used may vary between industries and even between organisations. Therefore, the system should be adaptable to a variety of data processing algorithms.

3.2.8 Requirements compilation

Table 1 summarises the requirements for the system that have been defined in this chapter.

Table 1: Summary of system requirements

Requirement	Description
Accessibility	Must have remote access.
Integration	Must utilise and integrate with existing infrastructure.
Information	Must provide raw detail data as well as condensed information.
Multilevel	Must have parameter detail level as well as overview levels.

Requirement	Description
Access control	Must have the ability to manage the information available to various users.
Configurability	Must be configurable without changes to code.
Adaptability	Must be adaptable to various data acquisition and data processing steps.

3.3 Development context

3.3.1 Condition monitoring data acquisition and processing

The condition monitoring system was developed as an information transfer and visualisation system for existing data acquisition and processing functionality. The data acquisition and processing formed part of Dr S van Jaarsveld's PhD. The data acquisition process was designed and implemented in the PhD. The data processing involved fault detection based on the previous month's data and the equipment design limits. The equipment health status was stored in the relational database as a tag value. In addition to data acquisition and data processing, the existing functionality included automated notification SMSs and reporting [18]. Figure 14 illustrates the existing condition monitoring functionality with which the newly developed system integrates.

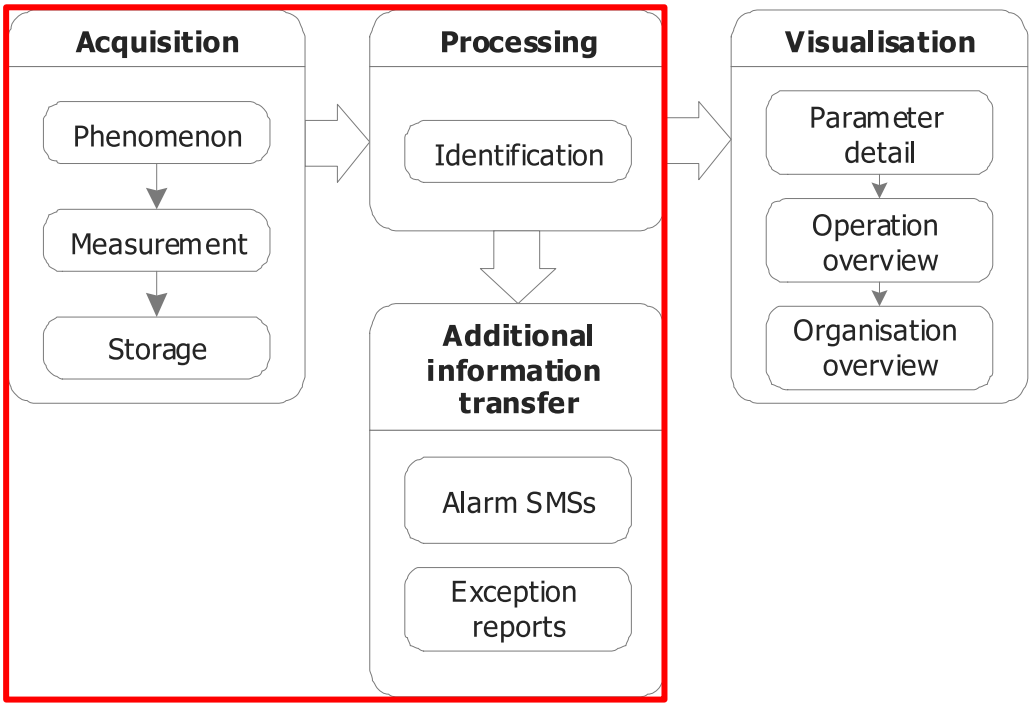


Figure 14: Existing condition monitoring functionality

3.3.2 Energy management systems

Energy management systems (EMSs) provide various tools that assist in improving industrial organisations operationally. These tools include, but are not limited to, energy and resource consumption data visualisation, DSM project performance tracking, environmental data management and asset management. EMSs use relational databases for user access control, organisational structure management, configuration of various software tools, and data access. Section 3.3.3 discusses the relational databases of EMSs and their existing features in further detail.

EMSs also serve as platforms to develop new software tools. An existing web-based EMS is therefore the ideal platform for developing the condition monitoring system presented in this study. The learning curve for system users is minimal as they are already familiar with the EMS. The database can be used to configure the system, give access to the condition monitoring data, and control user rights.

3.3.3 Relational database

An EMS uses a relational database for controlling user access, configuring various software tools depending on the client, accessing organisational structures, and accessing data. This section describes the typical database features utilised by the condition monitoring system presented in this study. General management of such a database is done using a database management system (DBMS).

3.3.3.1 Structures and tree structures

Structures and tree structures represent the business structures and operational layouts of organisations. Structures are typically used to represent entire organisations and consist of nodes and node connections. Nodes represent categories in an organisation from the organisation itself to an operation or a component. Node connections define the relationship between various nodes. Figure 15 presents an entity relationship diagram (ERD) showing the typical structure, node and node connection tables, and their relationships. Figure 16 presents an example of a simple organisational structure.

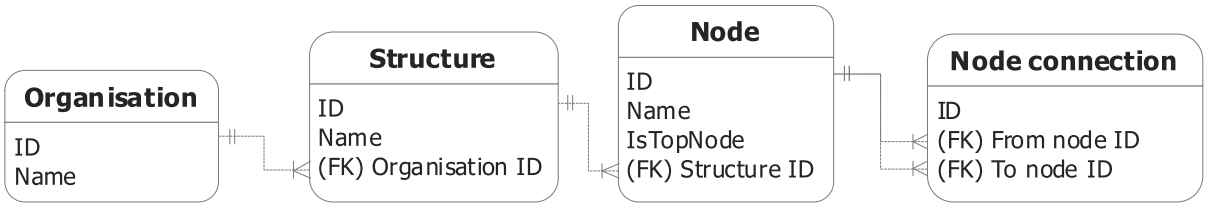


Figure 15: Partial database ERD of structure-related tables

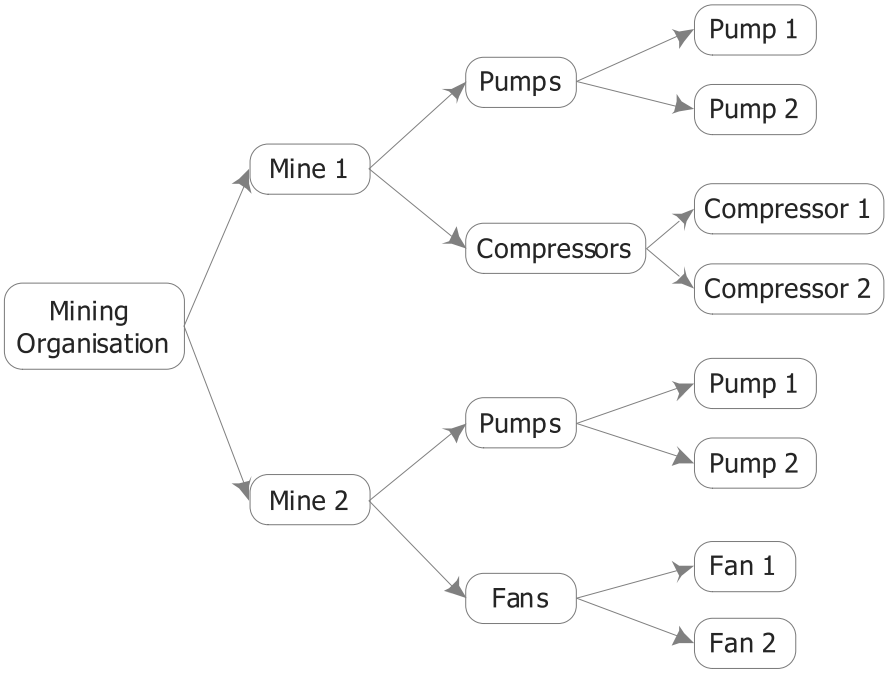


Figure 16: Visual representation of an example of a mining organisational structure

Tree structures are used to represent substructures of the organisation such as electricity, projects, or in the case of this study, condition monitoring substructures. Tree structures leverage off organisational structures. Tree structures overlay organisational structures and only include the applicable nodes and node connections. Figure 17 presents the tree structure and tree structure link node connection tables and their relationships with structure and node connection tables. Figure 18 presents an example of a simple tree structure.

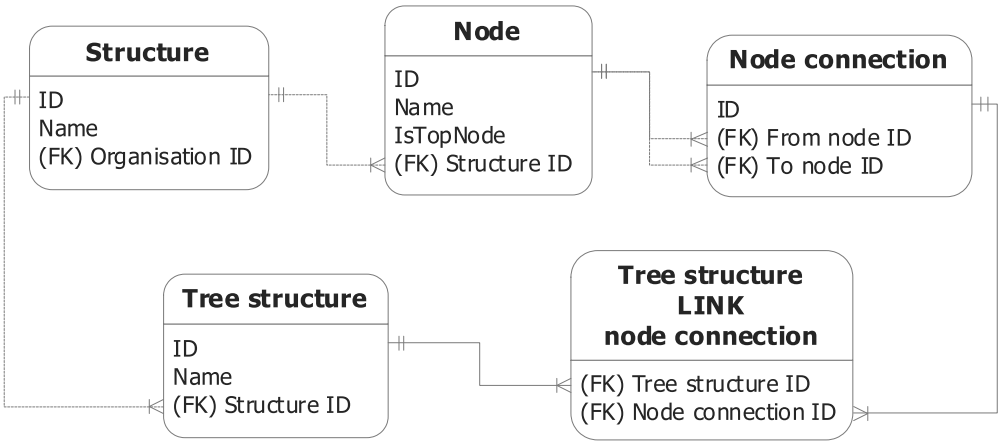


Figure 17: Partial database ERD of tree-structure-related tables

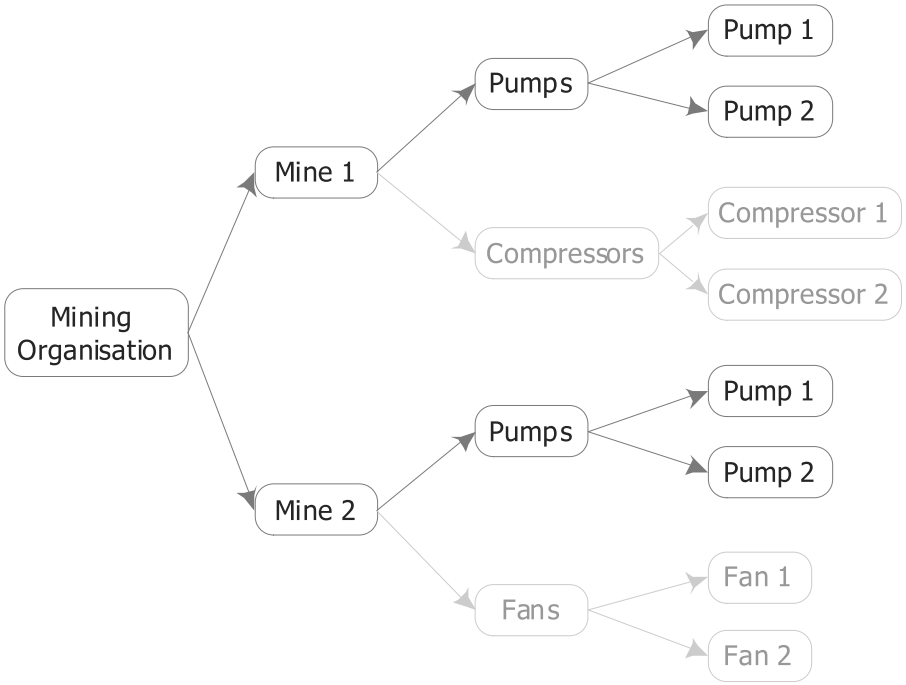


Figure 18: Visual representation of an example of a pumping tree structure

3.3.3.2 User access control

Basic information regarding users is stored in the user table. Typical information stored in this table include first and last names, the user's organisation, and the user's email address. Figure 19 presents a user table with its applicable fields and the link to an organisational table.

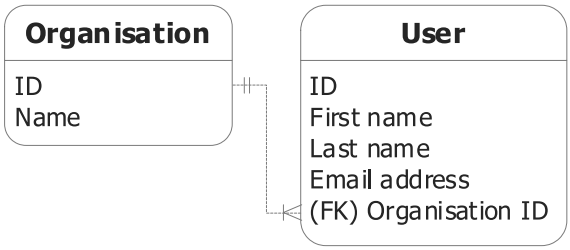


Figure 19: Partial database ERD of user and organisational tables

User authentication is handled with a user password table. A user password table contains the users' encrypted passwords and their encryption keys. Due to the sensitive nature of the table's contents, only the most trusted and experienced administration personnel have access to this content. The table contains additional information such as the expiration date and status of the user's account. Figure 20 presents a user password table and its link to the user table.

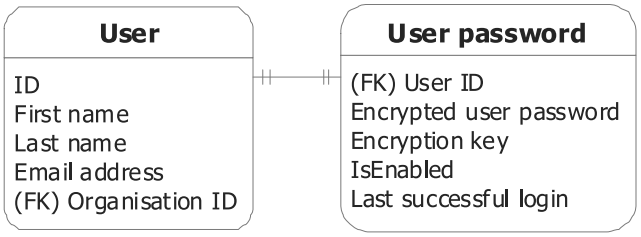


Figure 20: Partial database ERD of a user and user password table

User rights are handled through a user right table and a user link right table. User rights grant users read, write or administration access to various tools and menus within the EMS. The user link right table is used to link a single user to any number of these rights. Figure 21 presents the user right, user link right tables and their relationships.

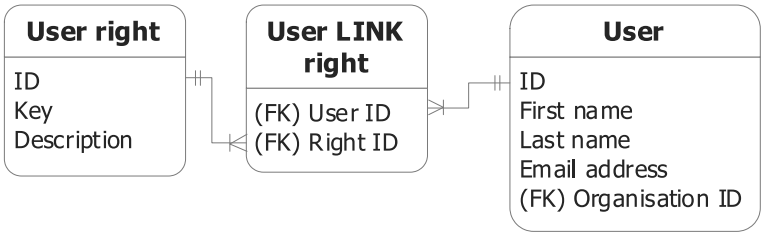


Figure 21: Partial database ERD of a user and user right tables

A user link node table is used to limit a user's access in a specific tree structure. The user is linked to the highest tree structure node he/she should have access to. The user is then able to access all child and grandchild nodes, but not parent or sibling nodes. Figure 22 presents the relationship between the user, the user link node and the node, the tree structure and the structure tables. Figure 23 presents the practical implication of this form of user access control.

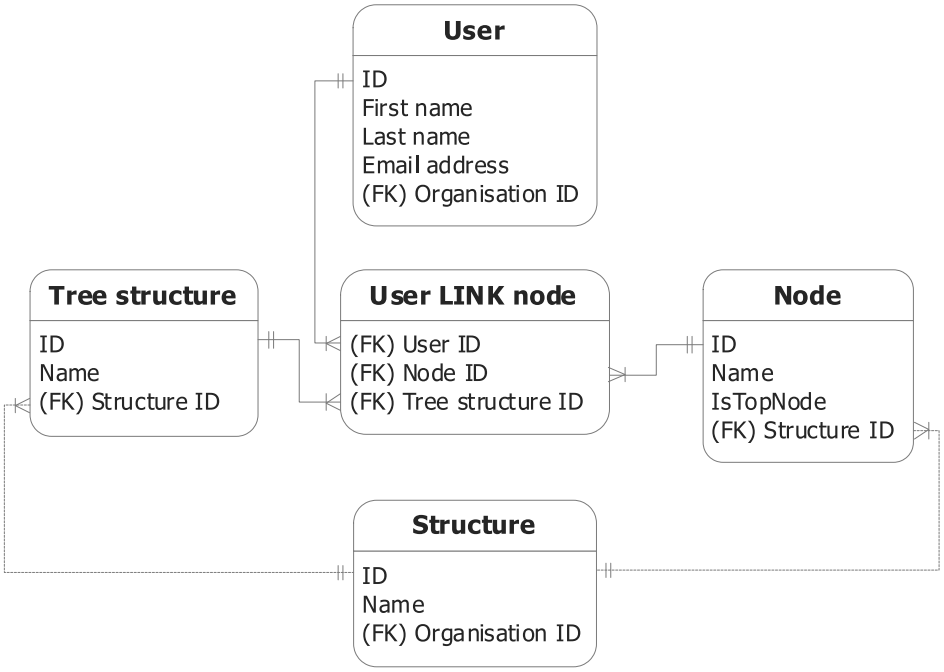


Figure 22: Partial database ERD of a user and applicable structure tables

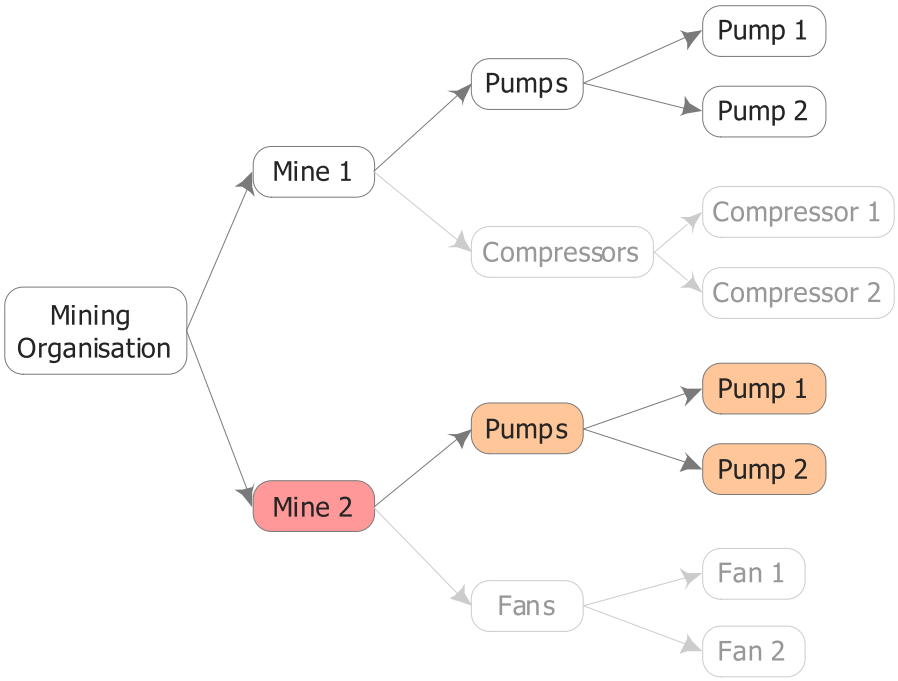


Figure 23: Propagated access in a pumping tree structure of a user with access to Mine 2

3.3.3.3 Tags and data

In addition to system management, the database is used to centralise data storage. Unfortunately, relational databases are not optimal for use with large volumes of data. Data is therefore typically preprocessed and aggregated to the finest resolution of half-hour intervals. The finer resolution raw data is stored in a big-data database, as discussed in Section 3.3.4.

Data can originate from various sources and can take on numerous value types. Data sources include automated logging systems, automated processing systems or manual user inputs through a DBMS, discussed in Section 3.3.5. Value types include monthly, daily, hourly or half-hourly data, which can take on numeric, string or Boolean values. Data values are sorted into various tables according to the value type. Figure 24 presents the various data tables with their respective fields.

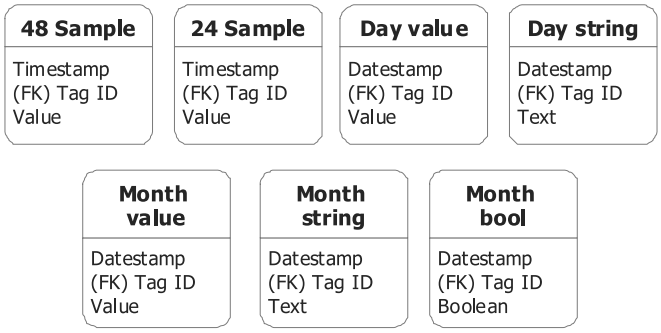


Figure 24: Various types of data tables

Data is identified using tags. Tags contain information such as the value type, the unit of the values, the data source and a description of the tag. Figure 25 presents a tag table together with its linked tables and their respective relationships.

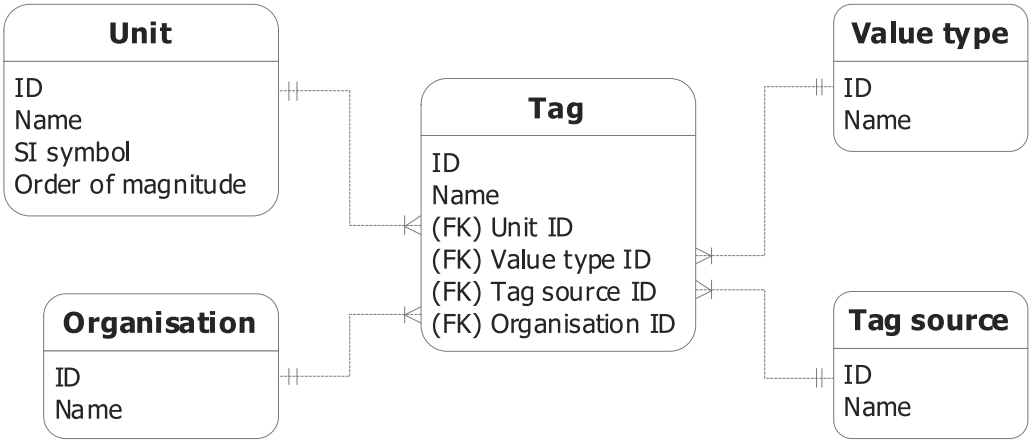


Figure 25: Partial database ERD of the tag-related tables

Each data value is linked to a tag to form data sets. This, combined with the value type table, allows the EMS to access data sets for visualisation and analysis. Figure 26 presents the data tables' relationships with the tag and value type tables.

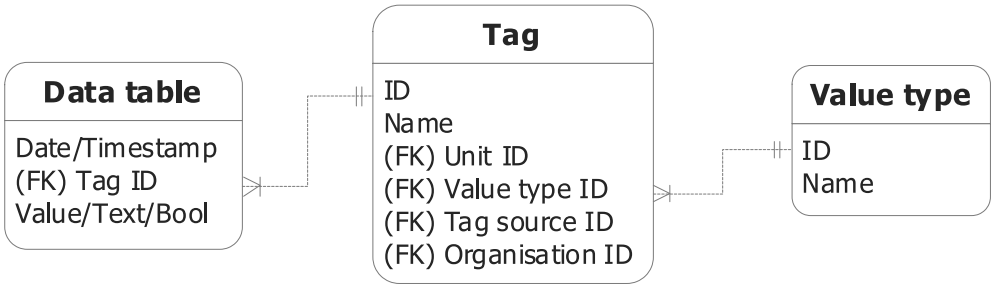


Figure 26: Partial database ERD of data-related tables

3.3.4 Big-data database

Big-data databases are used for storing raw data. This raw data typically originates in the form of log files from SCADA systems and remote servers installed at various operations. Data is transmitted from these operations to a centralised server via automated emails. The log files are then translated and the data is transferred to the big-data database.

Not all EMSs have direct access to big-data databases. Where a big-data database is available, data needs to be processed and transferred to the relational database first. This can be achieved with the tools available in a DBMS discussed in Section 3.3.5.

3.3.5 Database management systems

General maintenance of both the relational databases and the big-data databases is done through DBMSs. There are multiple advantages to using a DBMS. These include:

- **Control** – It allows control over personnel's access to the database. A system based on a graphical user interface (GUI) is a much safer, more stable and controllable method of reading and writing to and from the database than doing so directly through database queries.
- **Skill required** – A GUI enables a wider range of personnel to access database information as users need less skill to access data through a GUI than what is needed to develop customised queries to gain access to information.
- **Customised functionality** – It can be customised to simplify configuration of EMS software tools and user management, and to visualise event logs. A web-based DBMS can be accessed from anywhere with an internet connection.
- **Node builder tool** – A DBMS typically includes useful built-in functionality such as a node builder. A node builder simplifies creating and editing structures, tree structures and nodes. Figure 27 presents an example of a node builder tool.

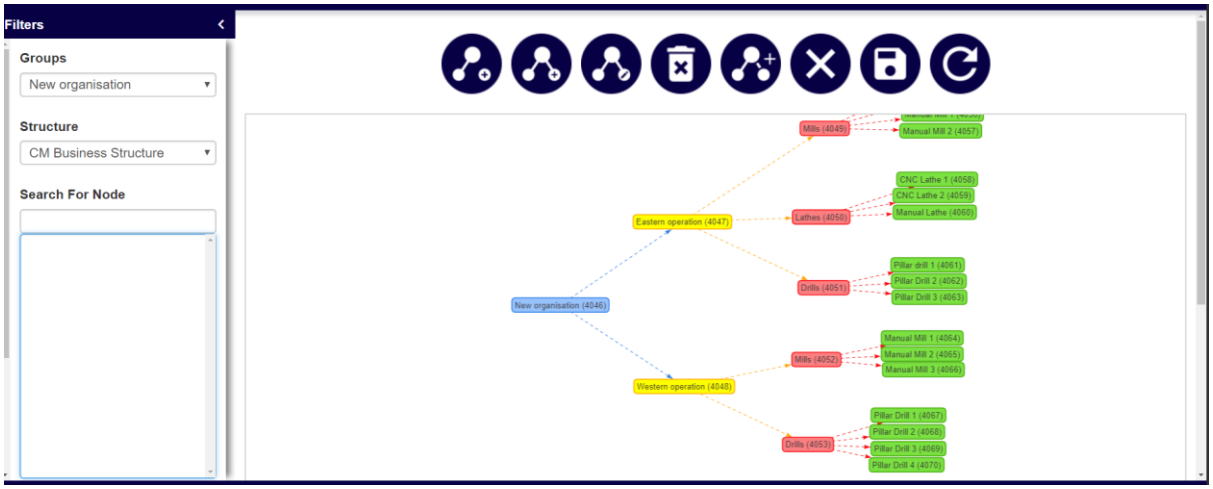


Figure 27: DBMS node builder tool

3.4 Functional design

3.4.1 Integration

The developed system utilises existing infrastructure and resources to reduce implementation time and cost. The existing infrastructure and resources available for the condition monitoring system development have been described in Section 3.3. Figure 28 presents the basic communication and data flow between existing systems. Figure 29 presents the integration of the developed system with this existing infrastructure.

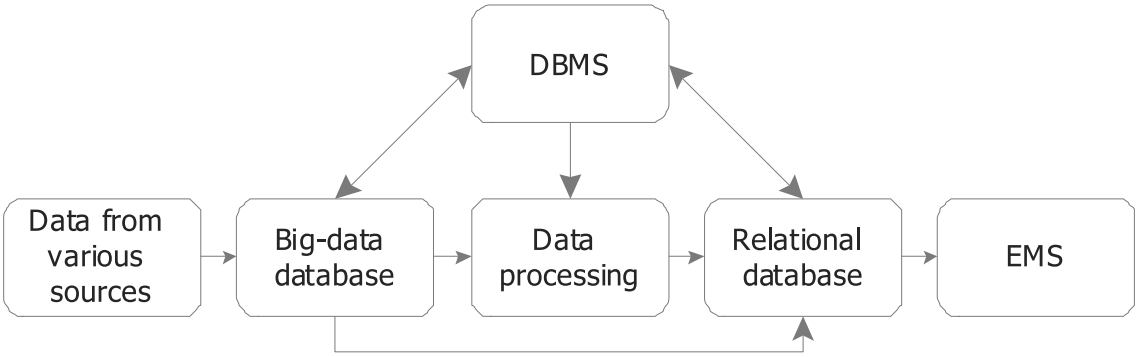


Figure 28: Simplified systems diagram of existing infrastructure

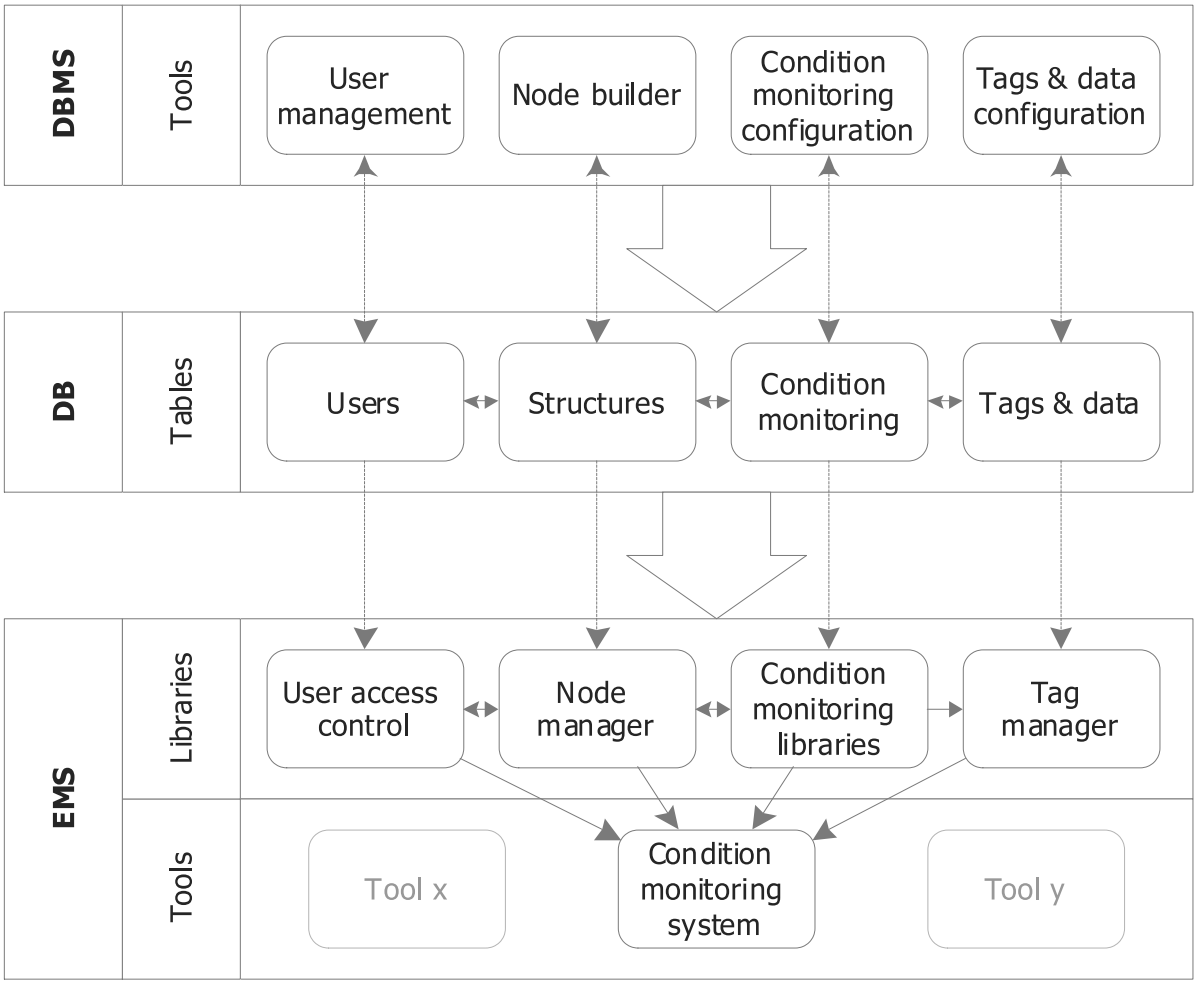


Figure 29: Integration of the system with existing infrastructure

3.4.2 Data flow

The developed system functions as a plug-and-play condition monitoring visualisation system to the rest of the condition monitoring processes. Figure 30 presents the three phases of condition monitoring and indicates the role of the developed system. The data acquisition and data processing phases can be adapted to suit the needs of each individual implementation. The only data constraint is that it must be transferred to the EMS's relational database. Figure 31 illustrates the adaptability of the system to various data acquisition and data processing steps.

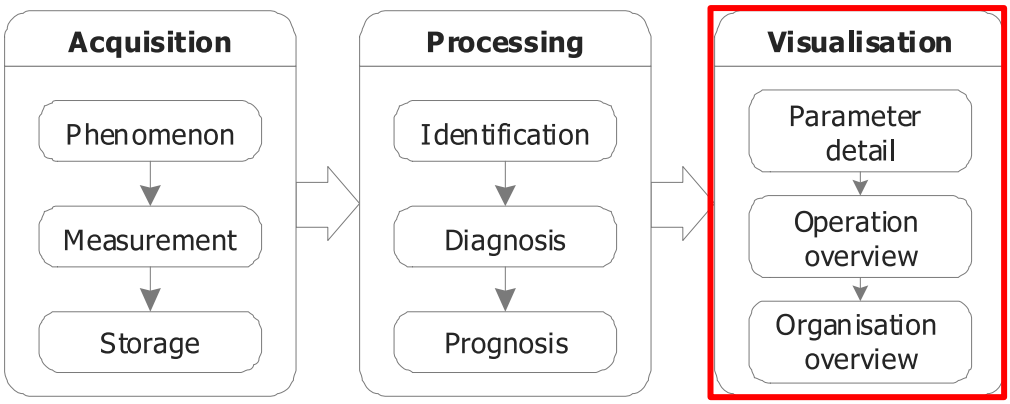


Figure 30: Phases of condition monitoring

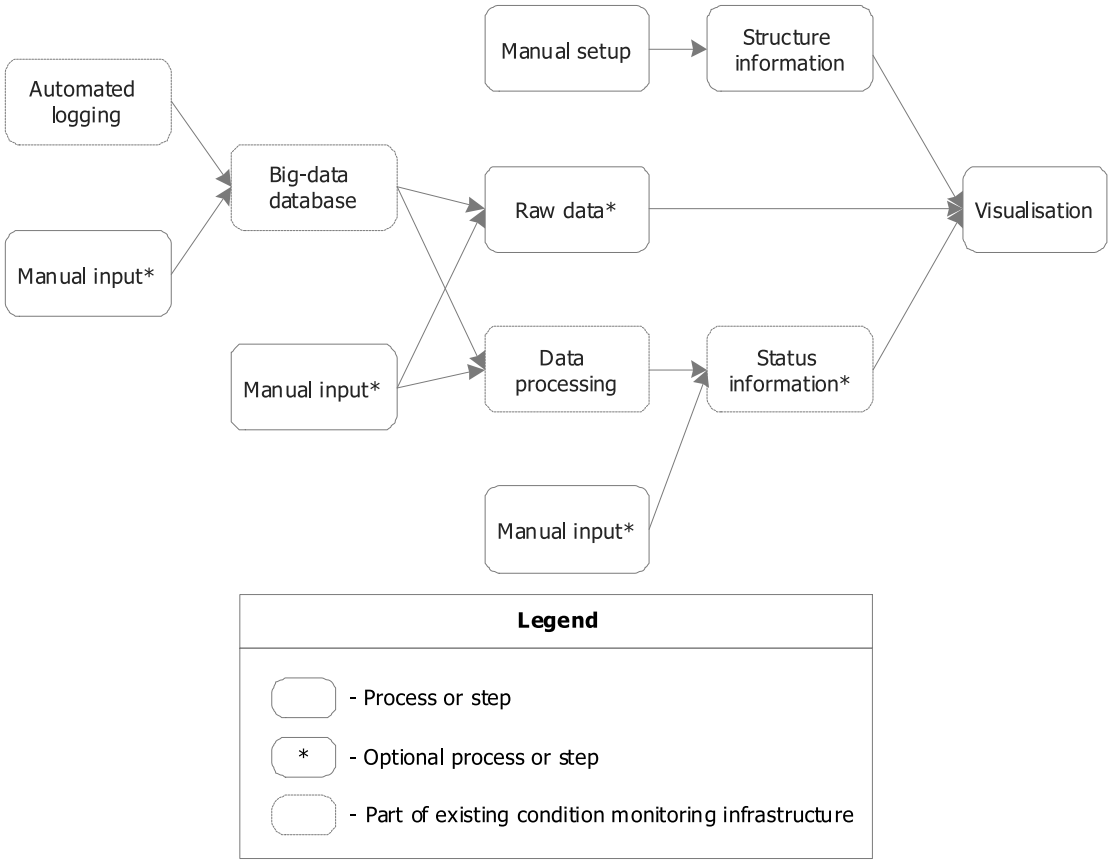


Figure 31: Illustration of system adaptability to data acquisition and data processing steps

3.5 Summary

The defined system requirements were based on the conclusions from the literature review. The requirements concern accessibility, integration, information, multilevel framework, access control, configurability and adaptability. These requirements aim to assist the system in achieving the study objectives of reducing maintenance reaction times through cognitive load reduction and data accessibility, while minimising system configuration time.

The system integrates with existing web-based EMSs. This enables the system to leverage off existing infrastructure and functionality. This functionality includes user authentication and access control, tree structures, databases and database management tools. Additionally, integration with existing EMSs allows users to use the system in a familiar environment.

The condition monitoring system is available as a tool in an EMS. A DBMS serves as a platform for the configuration of the condition monitoring system. DBMSs typically include node builder tools, which are used to easily configure tree structures for the condition monitoring system.

4 Detailed design

4.1 Preamble

This chapter provides a detailed description of the design of the system, which aims to solve the problem identified in Chapter 1. Section 4.2 presents the front end of the developed system. The system front end is used primarily for visualising the condition monitoring data. Section 4.3 describes the database design of the developed system and the manner in which it integrates with the existing database resources. Section 4.4 presents the developed system configuration tool. Section 4.5 describes the logic and algorithms which the system uses for data visualisation and cognitive load reduction. Section 4.6 verifies that the system meets the technical requirements, defined in Section 3.2.

4.2 Front-end system

4.2.1 Overview

The front-end system is the part of the condition monitoring system that the end user sees. It presents the user with all the data logged and processed by the condition monitoring system. Simply presenting the user with time series graphs of the numerous logged parameters is not a practical solution and will produce cognitive overload. Figure 32 presents a simple example of cognitive overload due to the manual monitoring of large numbers of parameters.

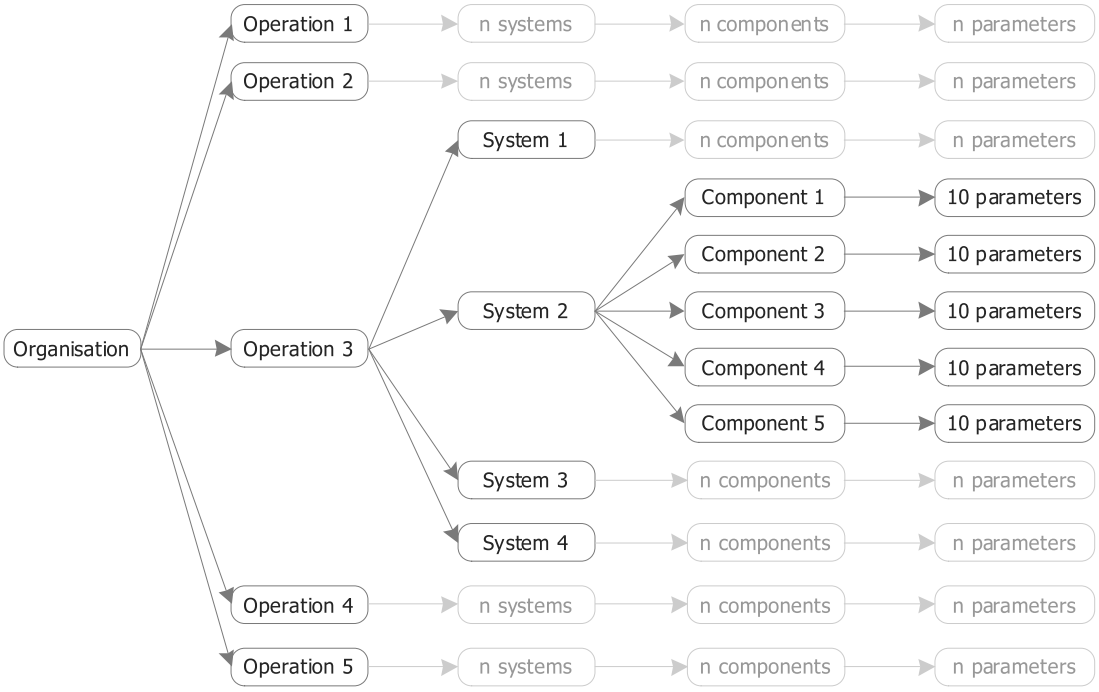


Figure 32: Illustration of the condition monitoring parameters to measure in an organisation

The front-end solution is implemented on a web-based EMS. Figure 33 presents a basic example of the system front-end user interface. The sections that follow describe the various components of the user interface.

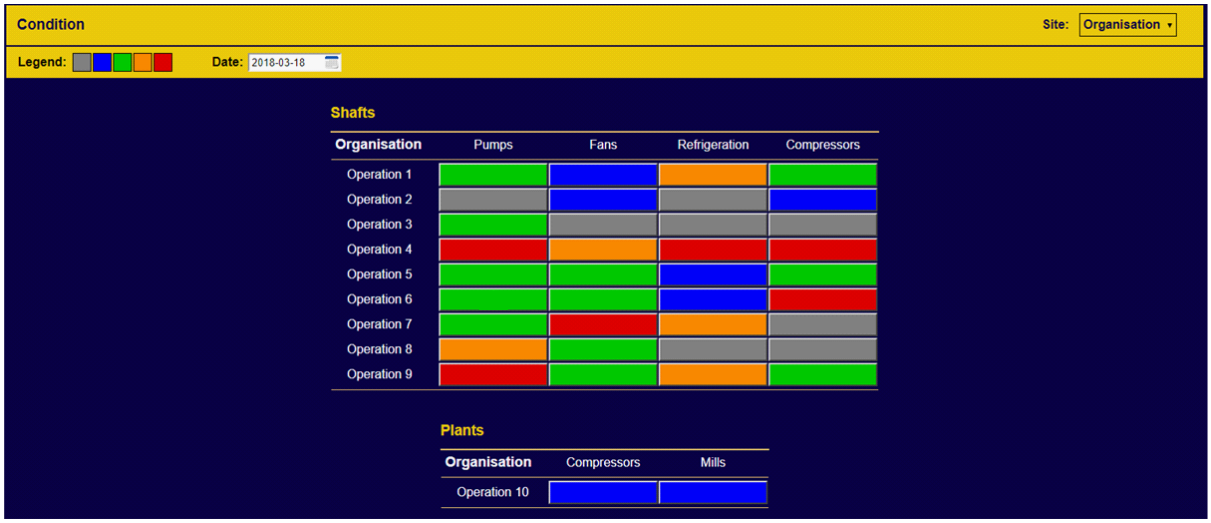


Figure 33: Example of the system front-end user interface

4.2.2 Status blocks

Status bocks have been developed to present the operational condition of components. Status blocks use various colours to visualise the status of component parameters. Figure 34 presents an example of the status blocks of a component.

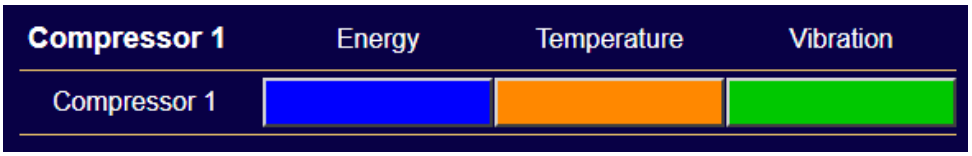


Figure 34: Typical component status blocks

A legend is provided at the top-left of the user interface to describe the various colours. Figure 35 presents the status block legend. Note that descriptions of the various legend items are provided in mouse-over tooltip text (not shown). Table 2 provides the full list of legend items with their respective descriptions.

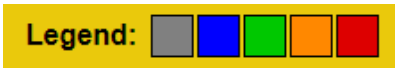


Figure 35: Status block legend

Table 2: Descriptions of the various status block statuses

Colour	Priority level	Description (as per user interface)	Description (detailed)
	0	No data	The status block does not have any lower levels (see Section 4.2.3) or any logged data to display
	0.5	No status	The status block does not have any status information available
	1	Good	The component parameter represented by the status block is operating according to design and at safe conditions
	2	Warning	The component parameter represented by the status block is operating close to its design limits
	3	Critical	The component parameter represented by the status block is operating outside of its design limits and requires immediate attention

The tooltip text of status blocks can be configured to display customised messages. These messages display if the block has either a warning or a critical status. The purpose of the customised tooltip text is to provide more detailed information regarding the nature of the warning/critical component without the user having to investigate further. If, for example, the status block of compressor temperature is critical, the tooltip text could be used to indicate that it is the motor drive end bearing temperature that is critical. Figure 36 presents an example of a status block with customised tooltip text.

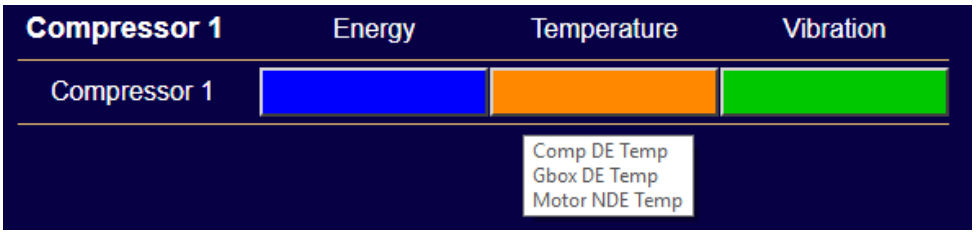


Figure 36: Status block with customised tooltip text

Status blocks can also be configured to display values. Display values are typically used to present the user with scalar values, such as a component's current operating efficiency. Figure 37 presents an example of status blocks with display values.

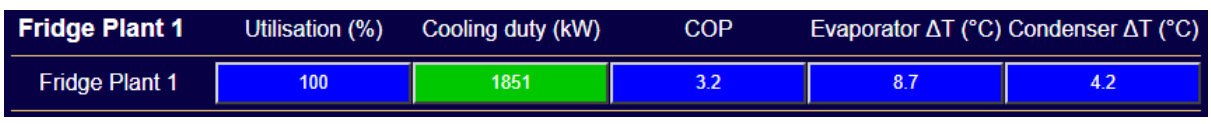


Figure 37: Status blocks with display values

4.2.3 Grids and multilevel navigation

To reduce cognitive load, the front-end solution uses multiple levels of overview grids. The number of levels and the contents of each level are fully configurable to suit the needs of the user. Typical configurations consist of an organisational overview level and an operational overview level. Each level's overview grid consists of operational-type categories on the vertical axis and parameter-type categories on the horizontal axis. The contents of each grid consist of a series of status blocks, as discussed in Section 4.2.2. Figure 38 presents an example of an organisational overview grid.

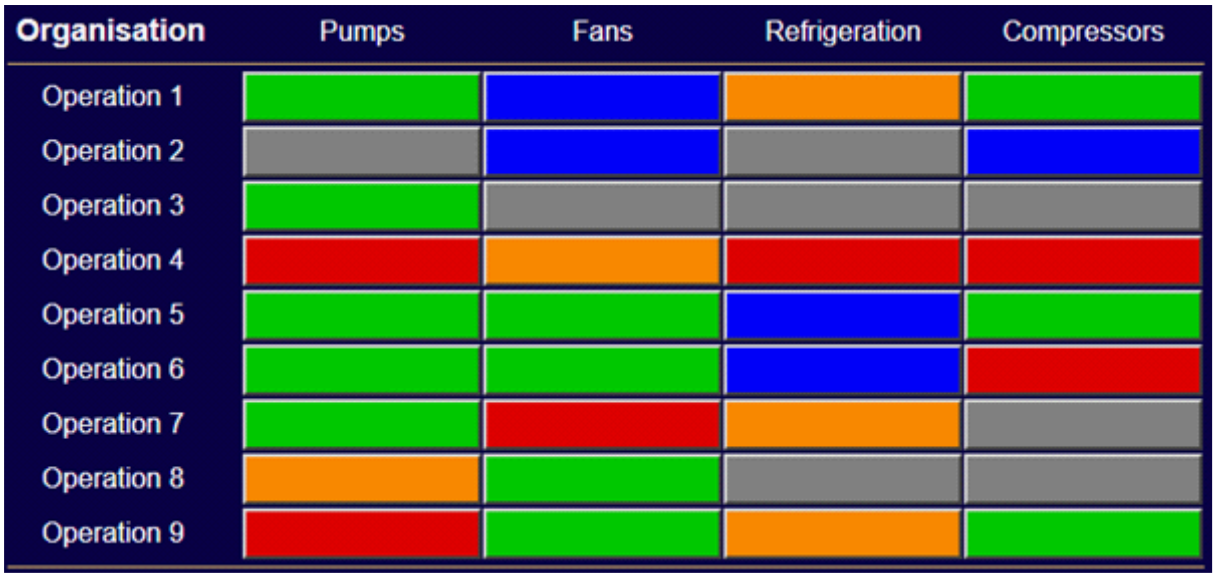


Figure 38: Organisational overview grid

Grid blocks with deeper levels are clickable, which allow navigations to lower levels. Clickable grid blocks are distinguished from non-clickable grid blocks by a pointer-cursor as opposed to a standard-cursor. Figure 39 presents navigation from an organisational overview level to an operational system overview grid by clicking a grid block.

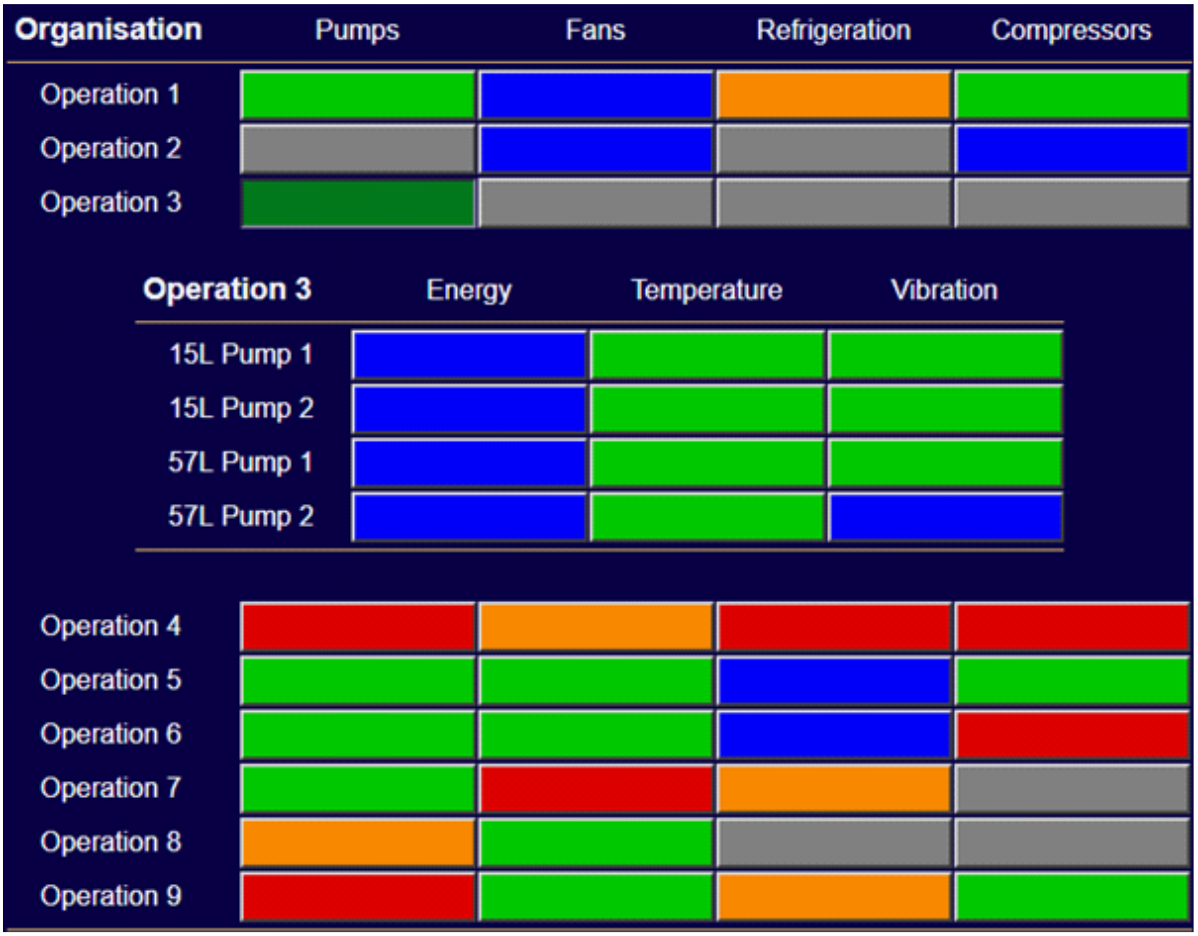


Figure 39: Operational system overview grid

Grid blocks with deeper levels inherit their status information from these deeper levels. The upper or parent grid block gets the highest priority status of all its child grid blocks. Figure 40 presents various examples of this type of status inheritance.

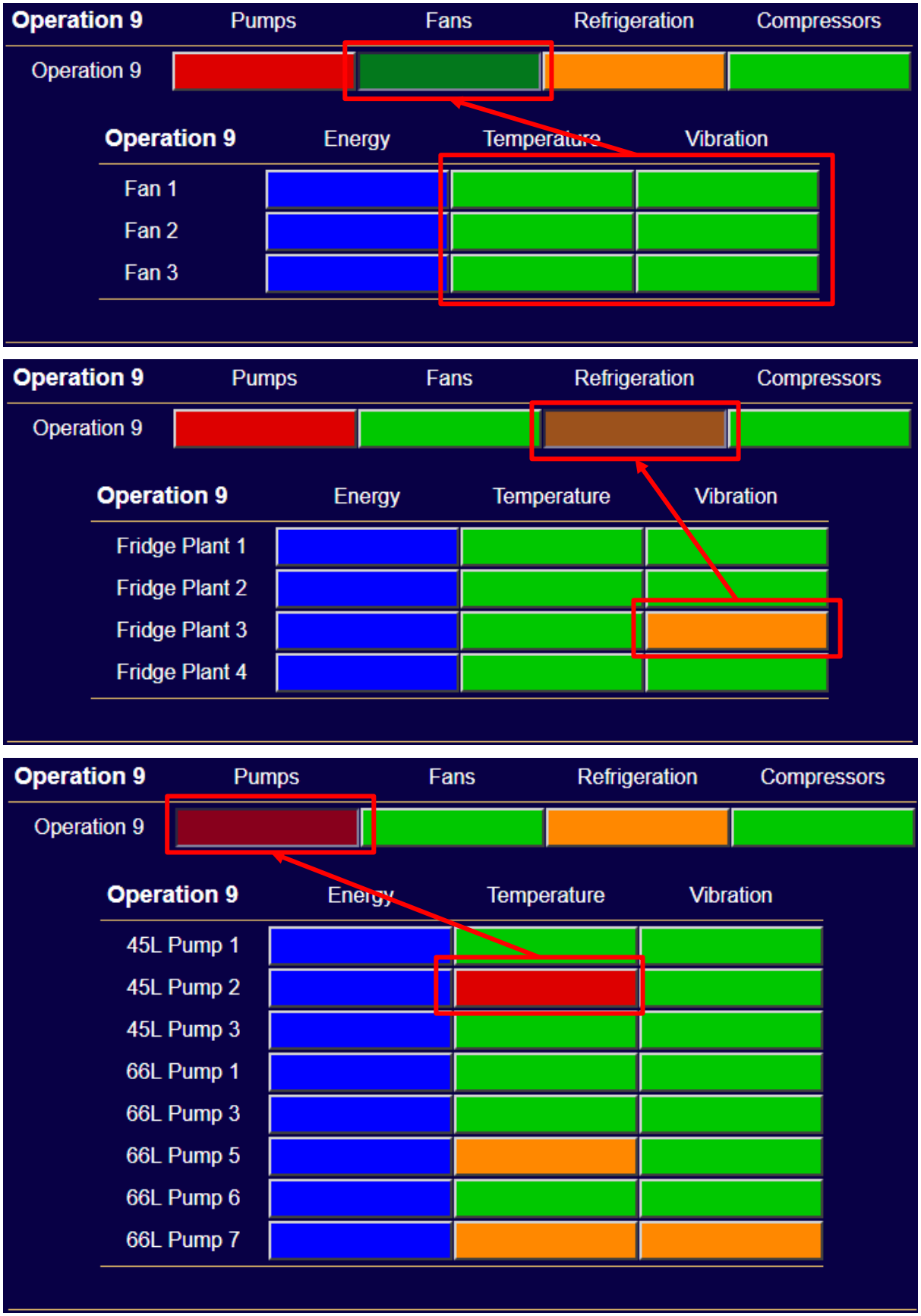


Figure 40: Various operational systems inherit the system statuses of their lower levels

The node picker, available at the top-right of the user interface, can also be used for navigation. The node picker presents the organisation's operational-type categories as a tree. These categories can be selected to navigate to a specific category. The node picker also allows quick and simple navigation to higher overview levels or back-navigation. The Level Up button automatically selects the parent operational category in the node picker tree. Figure 41 presents an example of the node picker with the Level Up button. Figure 42, Figure 43 and Figure 44 present navigation using the node picker.



Figure 41: Tree structure node Level Up button

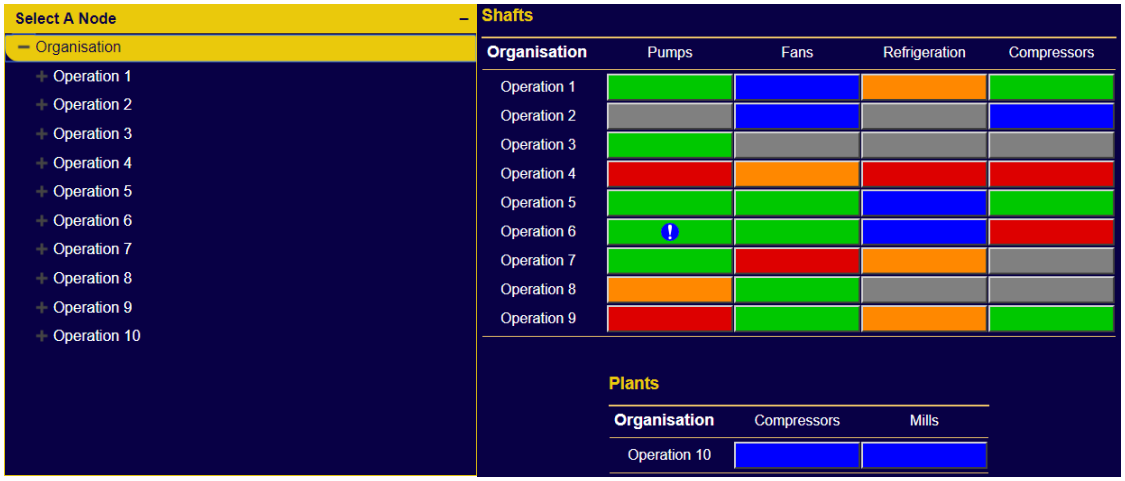


Figure 42: Navigation to organisational overview level using the node picker



Figure 43: Navigation to operational overview level using the node picker

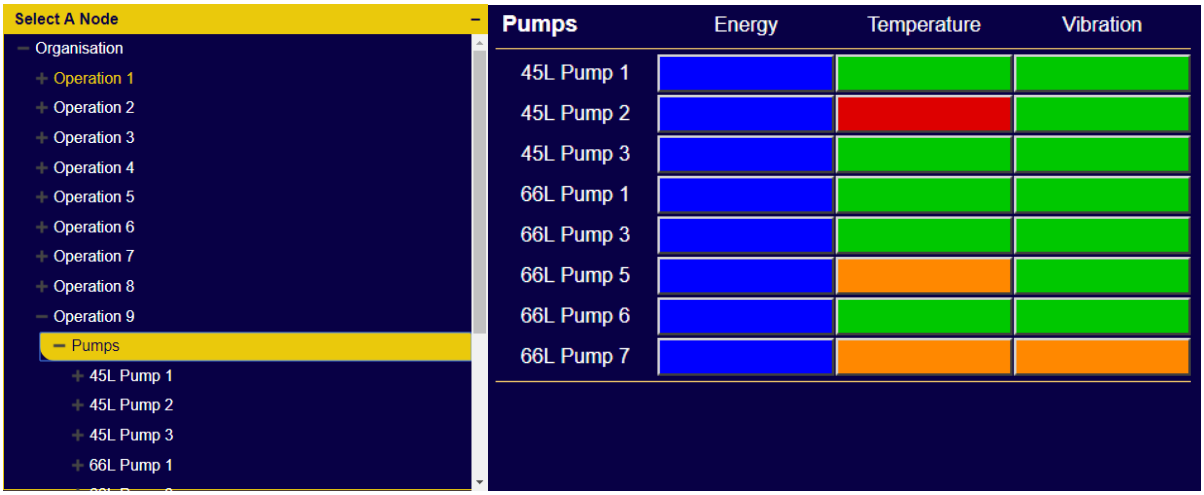


Figure 44: Navigation to operational system overview level using the node picker

4.2.4 Graphs

In addition to indicating status and navigation, grid blocks can be configured to display logged data. This typically applies to grid blocks that represent component parameters, but other grid blocks can also be configured to display data. Logged data is displayed using line graphs. The display information and the order in which these graphs are displayed are configurable, as discussed in Section 4.4.4. Figure 45 presents an example of a grid block with a linked graph.

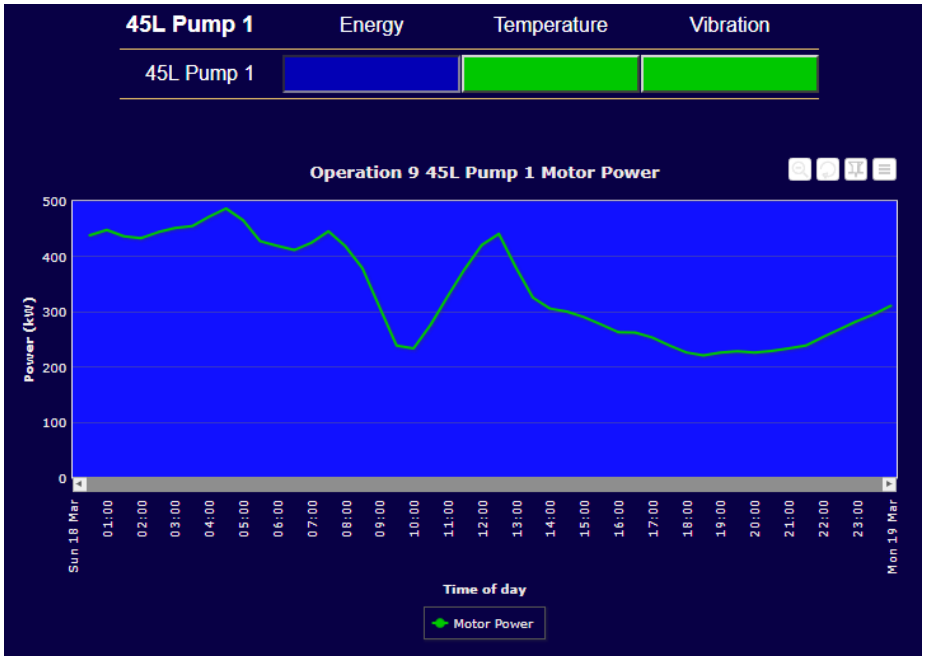


Figure 45: Grid block with graph linked

The data displayed on the graphs is tag data. Tags and tag data have been discussed in Section 3.3.3. This tag data, presented in the graphs, is typically the half-hourly averaged values of the raw logged data as discussed in Section 3.3.1 and Section 3.3.3.

Each data set can be configured with a limit. Limits are typically horizontal lines indicating a safe operating boundary for the given parameter as per the component's design specification. Figure 46 presents a graph with a configured limit value. Figure 47 presents a graph with multiple unique limits – each limit for a different data series.

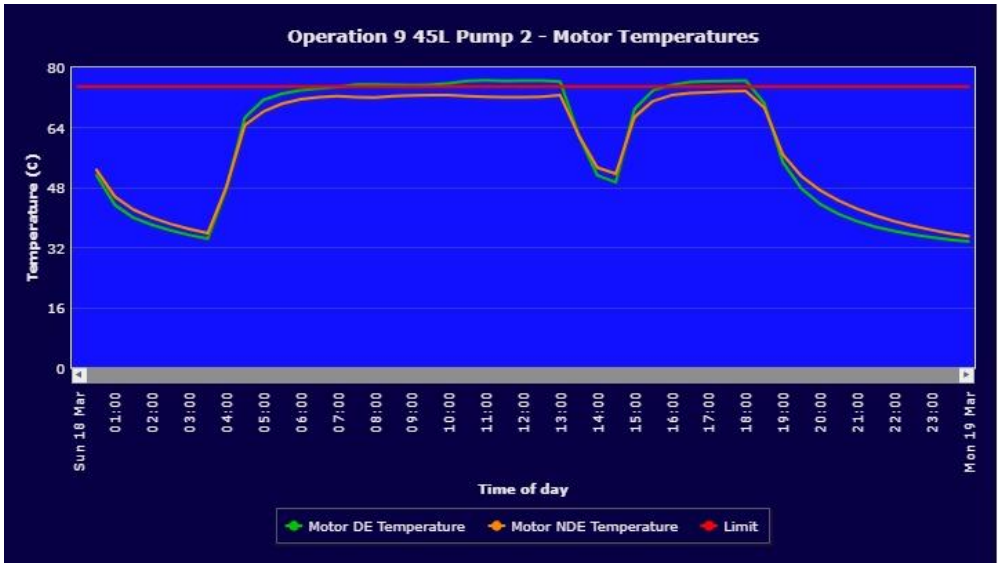


Figure 46: Graph with a single limit

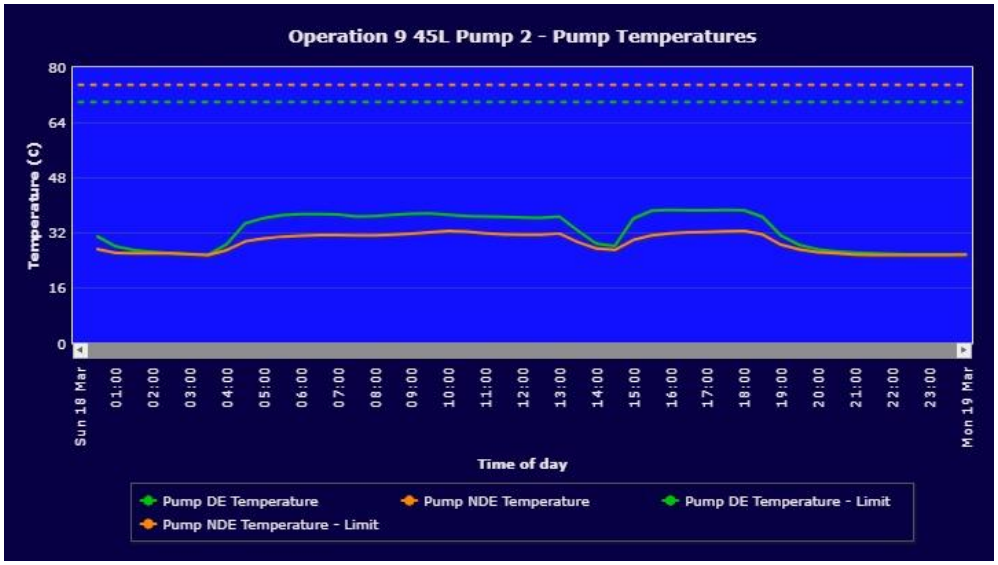


Figure 47: Graph with multiple unique limits

Historical data can be accessed using the date picker. The date range selector is used to display the data for a single day, 7 days or 14 days. The date picker and the date range selector are available at the top of the user interface. Figure 48 and Figure 49 present the date and date range selection options. Figure 50 and Figure 51 present examples of graphs displaying data for the various date ranges.



Figure 48: Date range and date picker

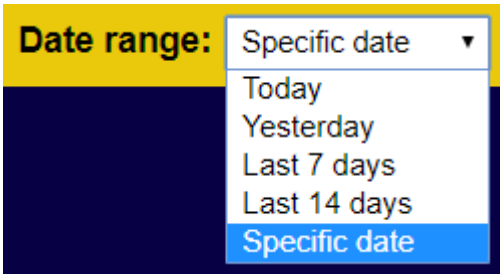


Figure 49: Date range options

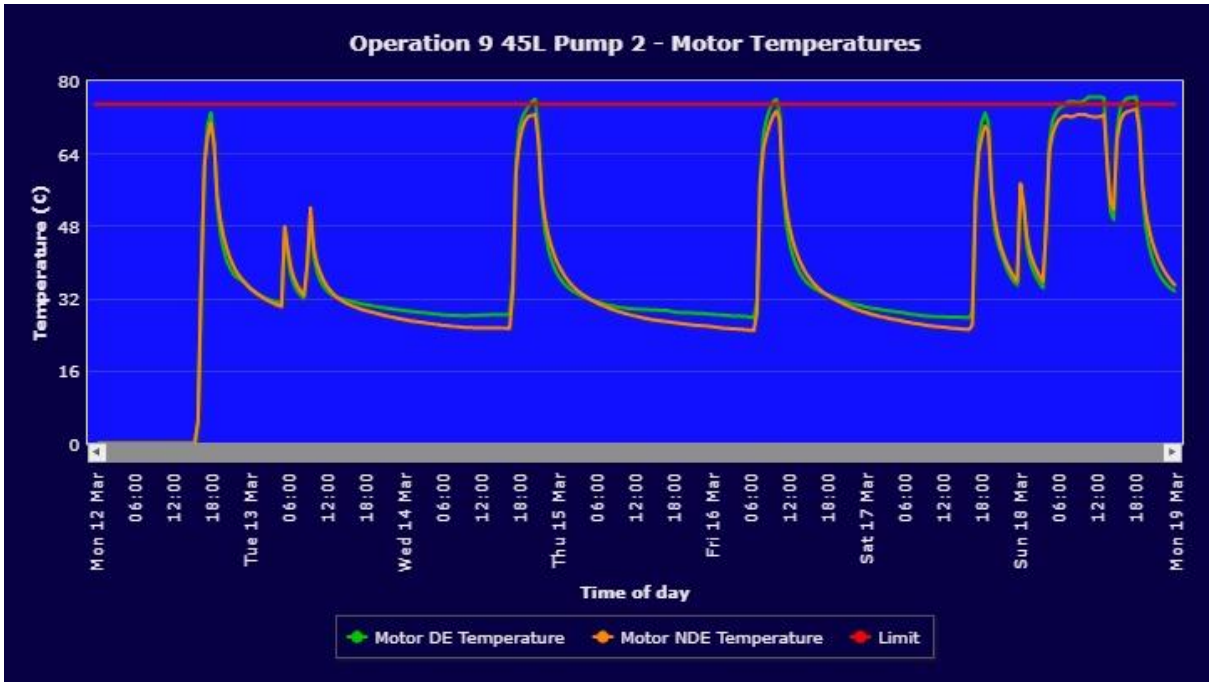


Figure 50: Graph displaying data for a seven-day period

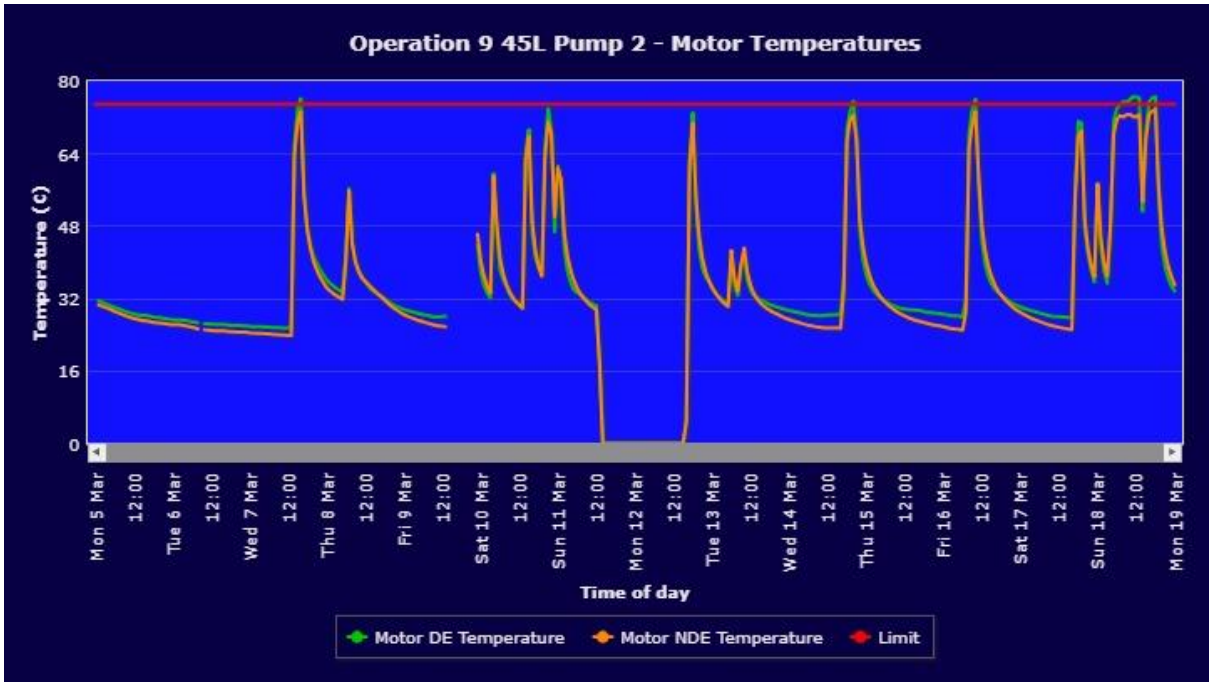


Figure 51: Graph displaying data for a 14-day period

4.3 Database

4.3.1 Node links

Structures, tree structures and nodes were described in Section 3.3.3. The tree structures and nodes provide useful features for presenting the organisation's condition monitoring data in the organisational structure. This entails linking the logged and status data directly to the tree structure nodes. The problem with this approach is the configuration time needed for a single operation. Operations typically consist of numerous components that use the same parameters for condition monitoring purposes. Figure 52 presents how this requires the creation and configuration of large numbers of identical nodes.

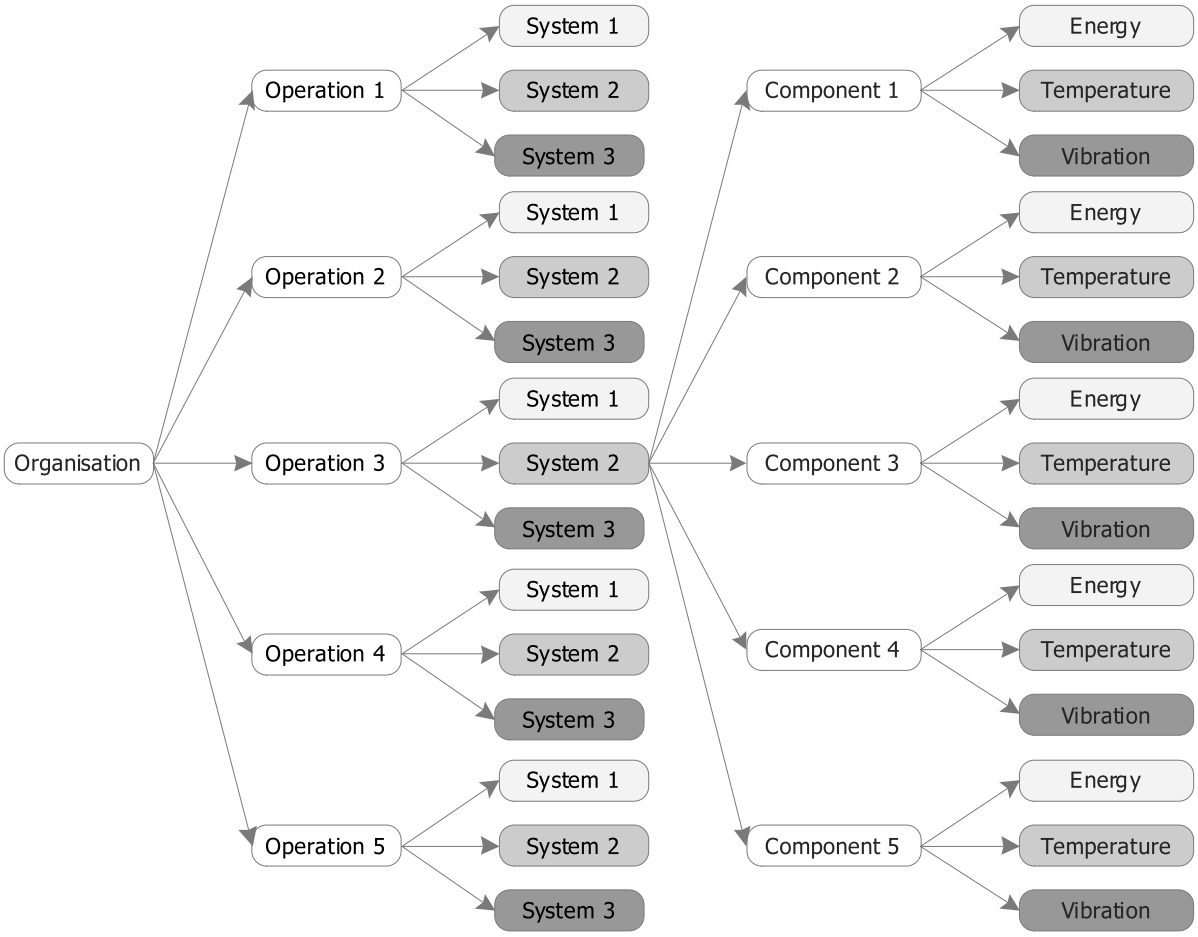


Figure 52: Illustration of the number of duplicate nodes in an organisational structure

To reduce the configuration time of the organisational condition monitoring tree structure, node links were developed to combine two separate tree structures. A node link creates a link between two nodes in two separate tree structures. Figure 53 and Figure 54 present the combination of two tree structures with node links.

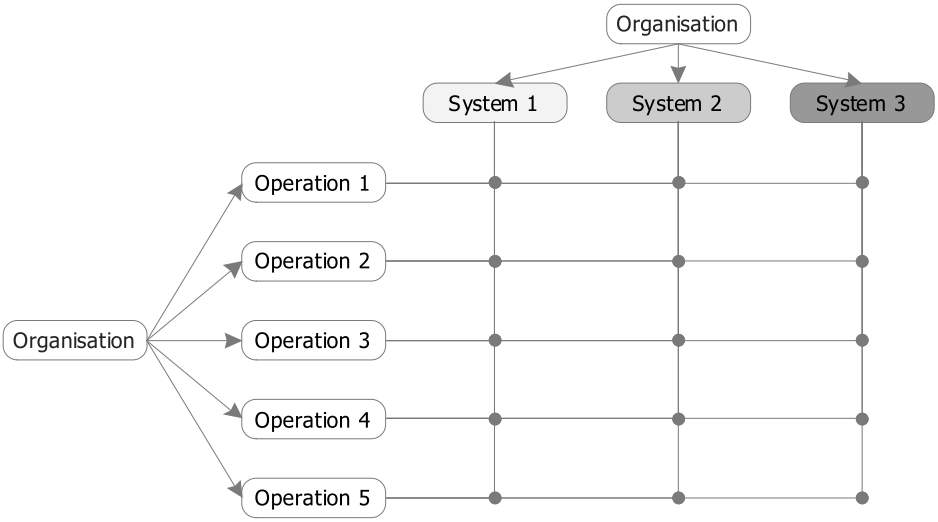


Figure 53: Combination of two tree structures to form operational system node links

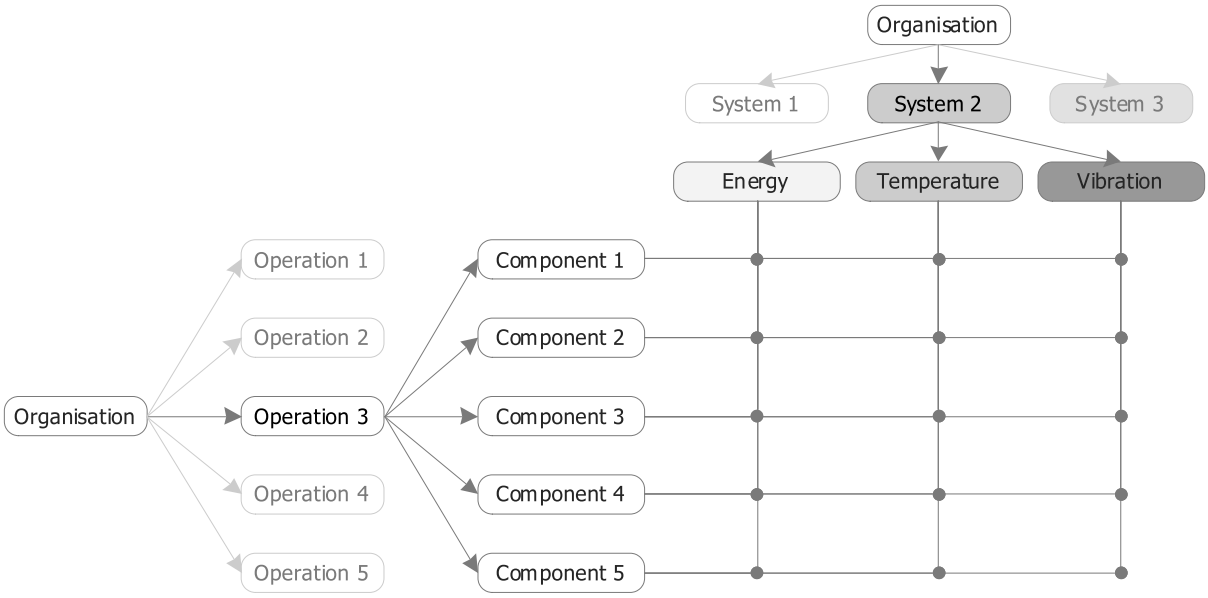


Figure 54: Combination of two tree structures to form component parameter node links

The first of the tree structures used is the organisational business tree structure. The business tree structure consists of operational-type nodes such as operations and components. Figure 55 presents a simple example of a condition monitoring business tree structure.

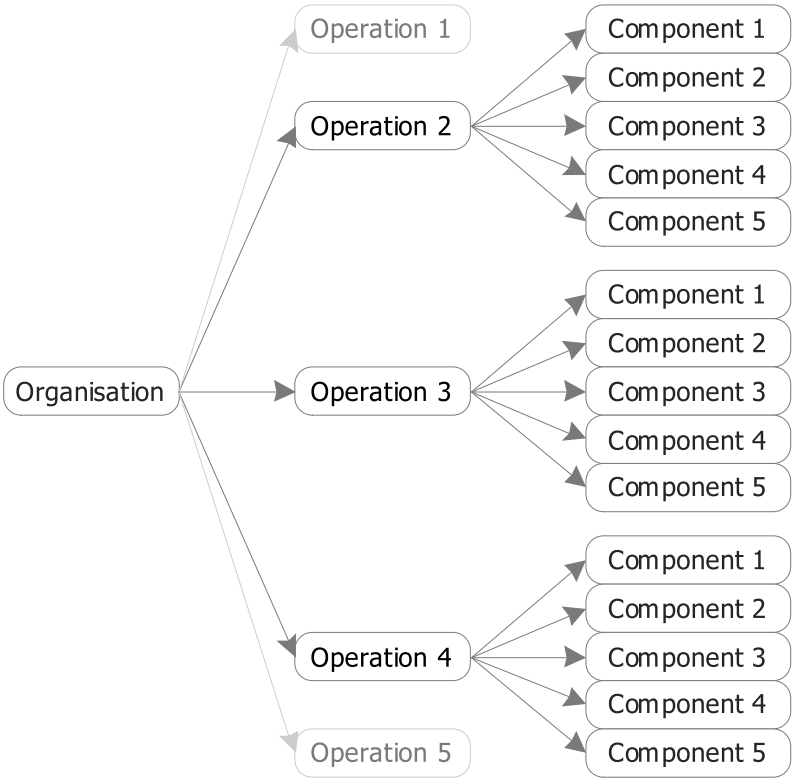


Figure 55: Condition monitoring business tree structure

The second of the tree structures used is the organisational data tree structure. The data tree structure consists of parameter-type nodes such as system groups and logged parameters. Figure 56 presents a simple example of a condition monitoring data tree structure.

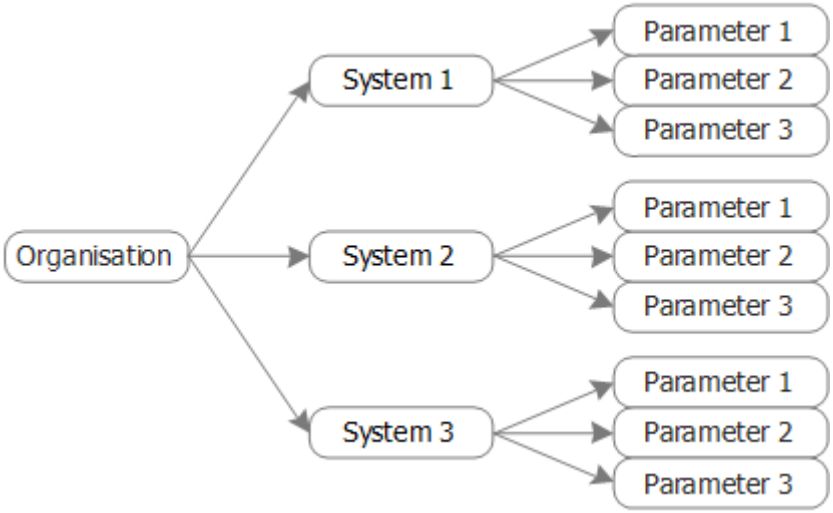


Figure 56: Condition monitoring data tree structure

Node links are used to configure the grid blocks in the front-end solution as described in Section 4.2.2 and Section 4.2.3. Each grid block on the front-end solution corresponds to a node link. Formatting information, discussed in Section 4.3.2, and graphs, discussed in

Section 4.3.3, can be linked to each node link. Figure 57 presents the node link table together with its relationship with the tree structures and node tables.

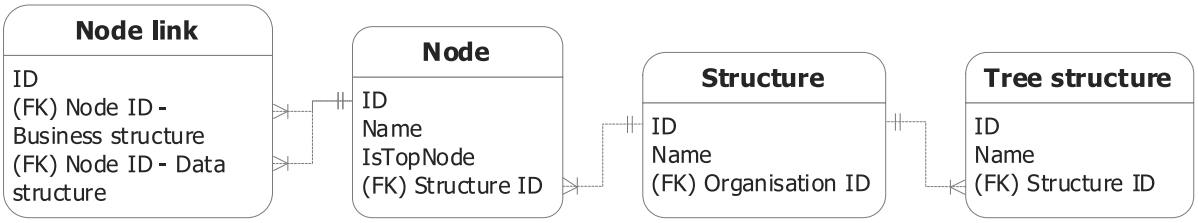


Figure 57: Partial database ERD of the node-link-related tables

With the use of the node link approach, the number of nodes required to configure an operation has been reduced significantly. Figure 58 compares the single tree structure and the two-tree structure and node link approach to support this statement. The figure compares the two approaches for a simple operation with multiple system groups, each consisting of numerous components.

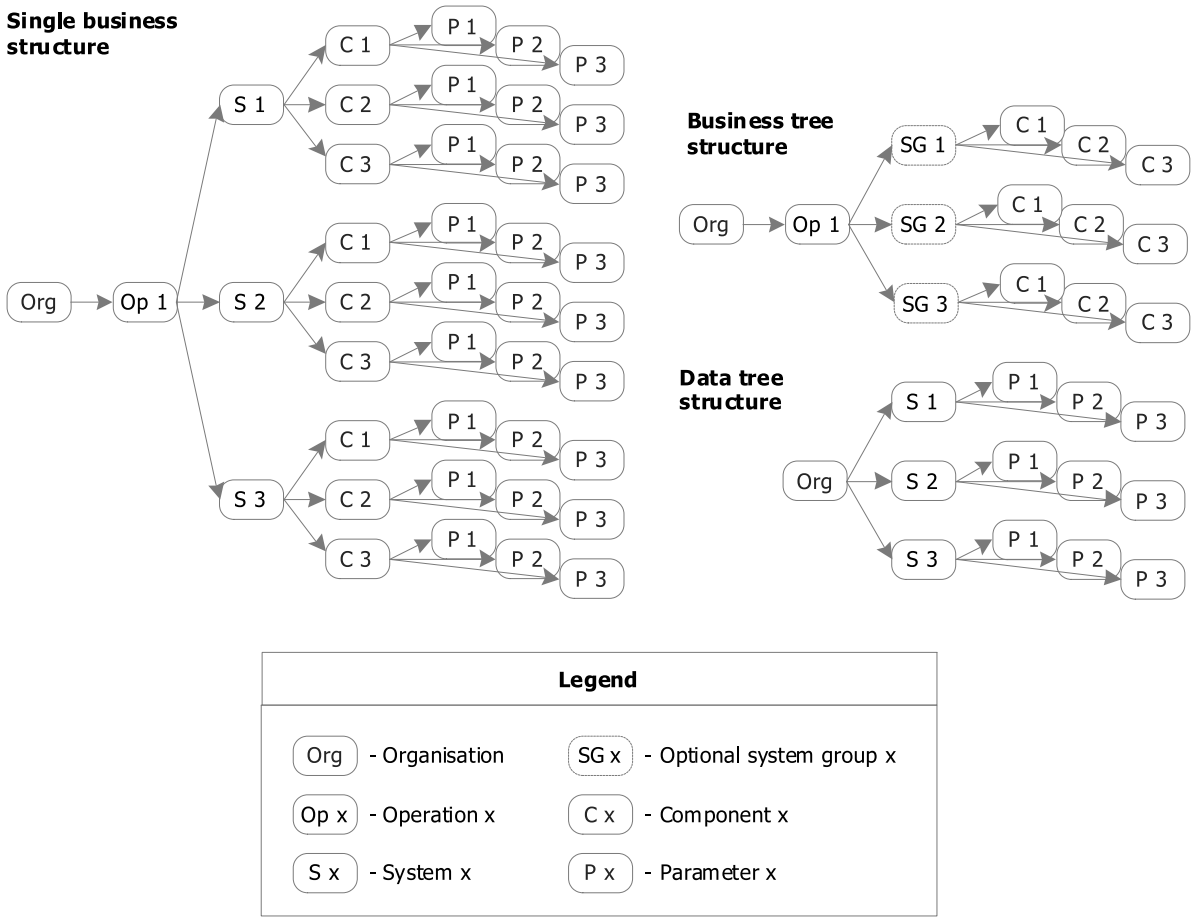


Figure 58: Comparison between single tree structure and two-tree structure approach

4.3.2 Formatting

Grid blocks, as described in Section 4.2.2 and Section 4.2.3, indicate the status of the block and can be configured to display values together with customised tooltips. These features are all part of the format information that can be linked to a node link.

The format information of a node link consists of a format type and a tag. Table 3 describes the various format types. The front-end system uses the format type to determine the type of display information the tag contains. Format types can indicate if the system should use the tag to set the colour of the block or to set the tooltip text etc.

Table 3: Descriptions of the various format types

Format type	Description
Status	Linked tag that indicates the grid block's status
Warning text	Linked tag for customised tooltip text to display on a grid block with a Warning status
Critical text	Linked tag for customised tooltip text to display on a grid block with a Critical status
Display average	Linked tag to display a customised value in the grid block

Tags, as discussed in Section 3.3.3, are used as identifiers for a data set. The tags used for formatting are typically daily tags with numeric or string value types. The tags are used to access the status, block title or display value of a node link for a given date. Figure 59 presents the format table together with its relationship with the format type, tag and node link tables.

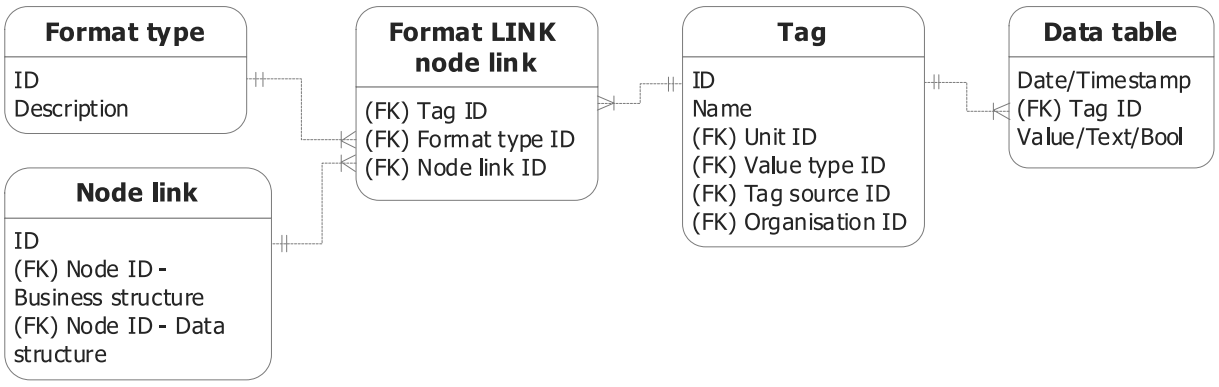


Figure 59: Partial database ERD of the format tag-related tables

4.3.3 Graphs and series

Section 4.2.4 mentions that grid blocks can be used to display logged data by using line graphs. These line graphs are configurable in terms of their basic display information and the series presented by each graph. Multiple series can be configured for a single graph and multiple graphs can in turn be configured for a single grid block. The graphs and series

information have therefore been divided into two separate tables to achieve these one-to-many relationships.

The graph table is used to define the basic display information of a graph, such as the graph title, the axis headings and the order in which the graphs are displayed for a grid block. To display the graph under a specific grid block, the graph table is linked to the node link table. Figure 60 presents the graph table together with its relationship to the node link table.

The series table is used to define the information regarding each graph series. The series table consists of the series description, the hex colour of the series, the order in which the various series should be displayed on a graph, a data tag, a limit value and a limit tag. The data tag refers to the actual logged parameter data. The limit value is used to display a constant value as a limit for the particular series. The limit tag is used in a similar way to the limit value, but since it referrers to a tag, the limit can have dynamic values as per the tag data. Note that the limit value and limit tag fields are optional. Figure 60 presents the series table and its relationship to the graph and tag tables.

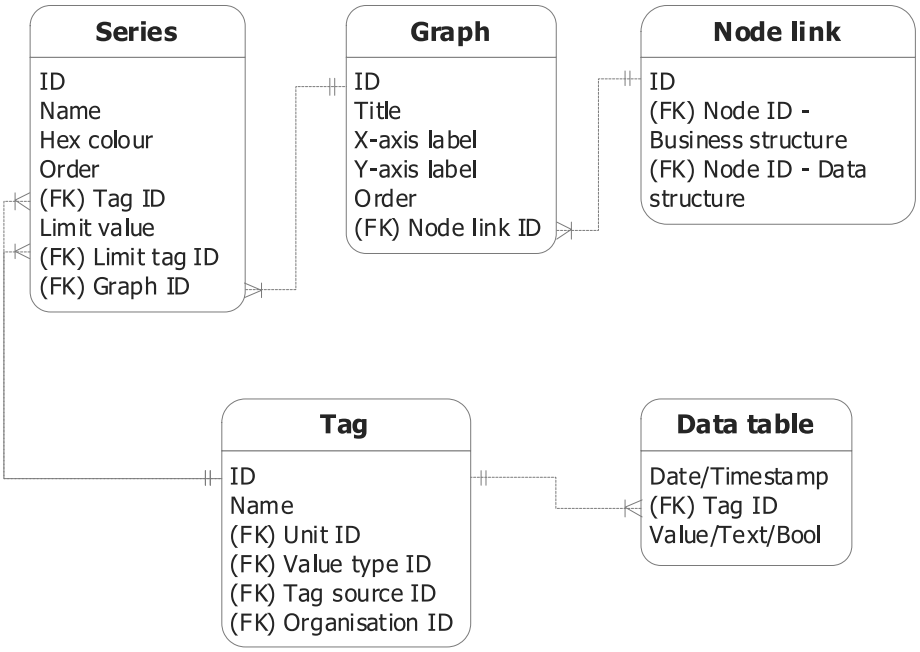


Figure 60: Partial database ERD of the graph and series related tables

4.4 Configuration tool

4.4.1 Overview

The front-end system described in Section 4.2 is configured with a user interface developed and implemented on a DBMS, as described in Section 3.3.5. The user interface is used to create, update and remove entries of the database tables described in Section 4.3. The DBMS user interface enables project engineers to configure the user interface of the front-end solution. Figure 61 presents the DBMS condition monitoring configuration tool user interface used to configure the front end of the condition monitoring system. The subsections that follow describe the various components of the user interface in further detail.

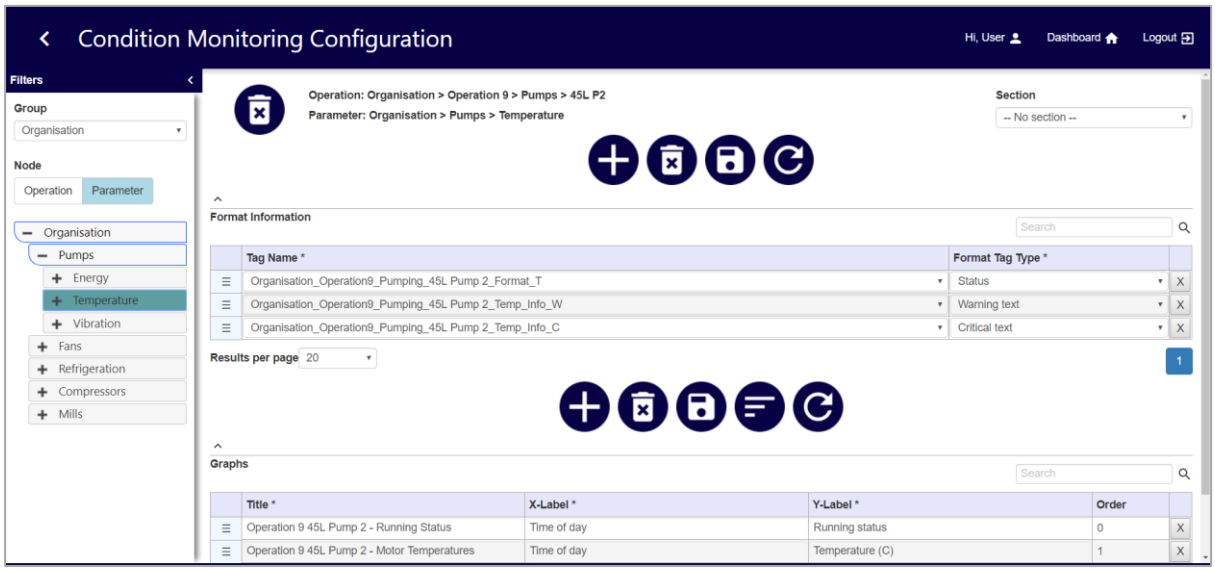


Figure 61: Basic DBMS condition monitoring configuration tool user interface

4.4.2 Node links

The node links, described in Section 4.3.1, create a link between a business tree structure node/operational node and a data tree structure node/parameter node. Figure 62 presents the node trees with which the user selects the combination of the two nodes from business and data tree structures.

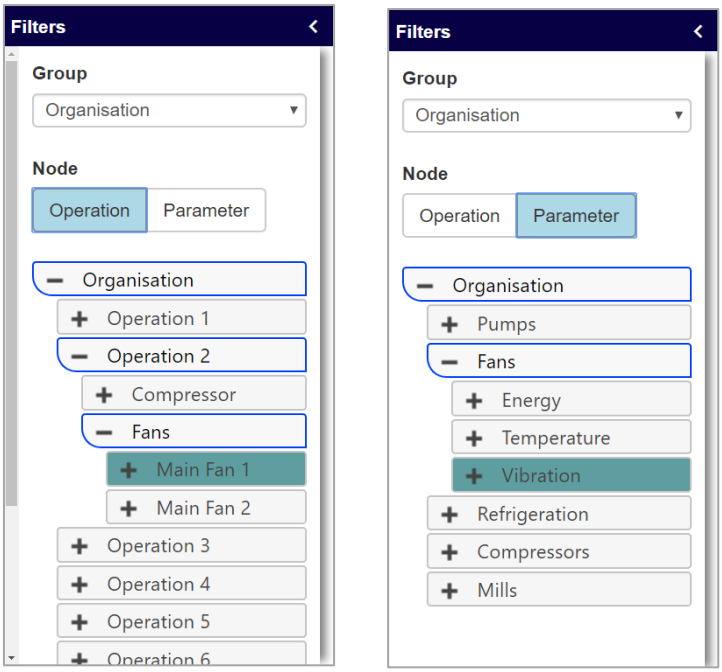


Figure 62: Left: Business tree structure node tree; Right: Data tree structure node tree

The top of the user interface displays the selected nodes with breadcrumbs. This assists the user to keep track of the selected nodes and, thus, the node link which is being created, updated or removed. Adjacent to the breadcrumbs is a Create/Delete button. This button is used to create or delete node links. Using error messages, the user is prevented from deleting node links with existing format or graph information. Figure 63 presents the Create Node Link button with its applicable breadcrumbs.

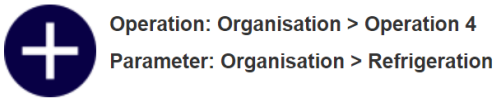


Figure 63: Create Node Link button with breadcrumbs of the selected nodes

Once a node link has been created for two selected nodes, the Create Node Link button is replaced with a Delete Node Link button. Figure 64 presents the Delete Node Link button, which replaces the Create Node Link button. In addition to the Create Node Link button being replaced with a Delete Node Link button, format information and the graph information are loaded when an existing node link is selected. The format information section is described in Section 4.4.3 while the graph information section is described in Section 4.4.4.

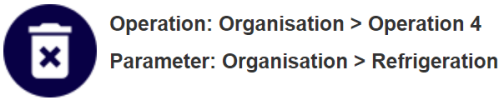


Figure 64: Delete Node Link button with breadcrumbs of the selected nodes

4.4.3 Formatting

Node link formats, as discussed in Section 4.3.2, consist of a format type and a format tag. The selected node link's format tags together with their respective format types are configured in the formatting section of the user interface. Figure 65 presents an example of a node link's formatting configuration.

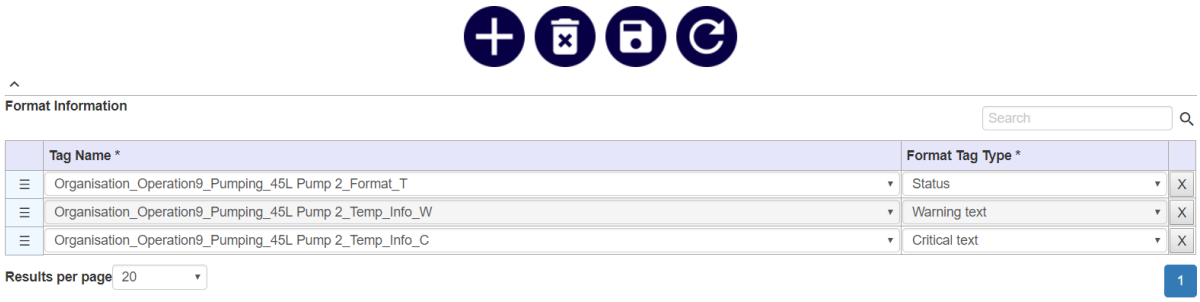


Figure 65: Node link formatting configuration

Selecting the format tag field opens a tag search dialog. Tags can be searched by name using a regex search. The searched tags appear in a list box from which they can be selected as a format tag. Figure 66 presents an example of the tag search dialog.

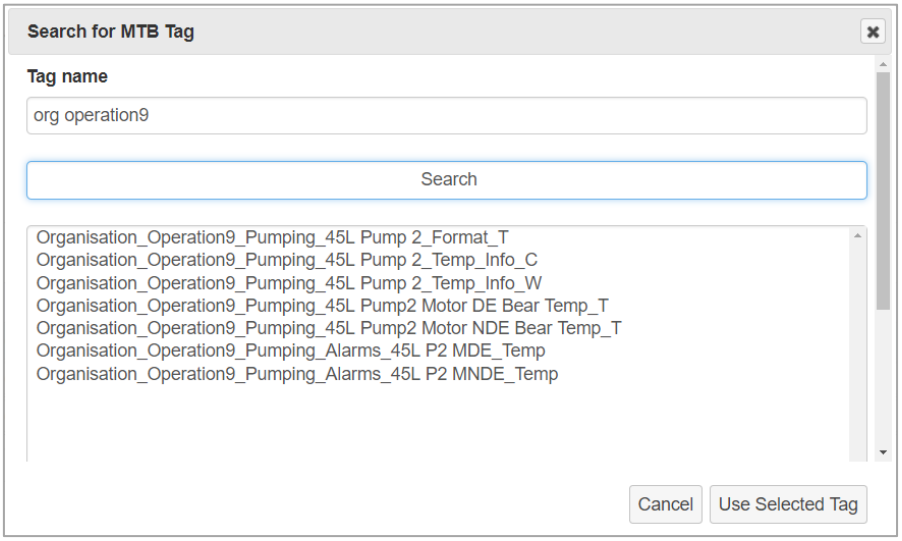


Figure 66: Tag search dialog

The format tag type drop-down indicates the tag value type and the manner in which it should be used to format the applicable front-end solution grid block. Figure 67 presents the format tag type drop-down. The options in this drop-down correspond to the format types, as described in Section 4.3.2, Table 3.

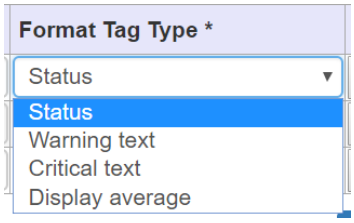


Figure 67: Format tag type drop-down menu

4.4.4 Graphs and series

The graphs and their applicable series are divided into two separate database tables, as discussed in Section 4.3.3. The graph section in the configuration user interface reflects this by dividing the section into a basic graph configuration section and a series section. The series section is hidden by default and requires an existing graph to be selected for the series section to appear. Figure 68 presents an example of the entire graph configuration section, containing graphs and series.

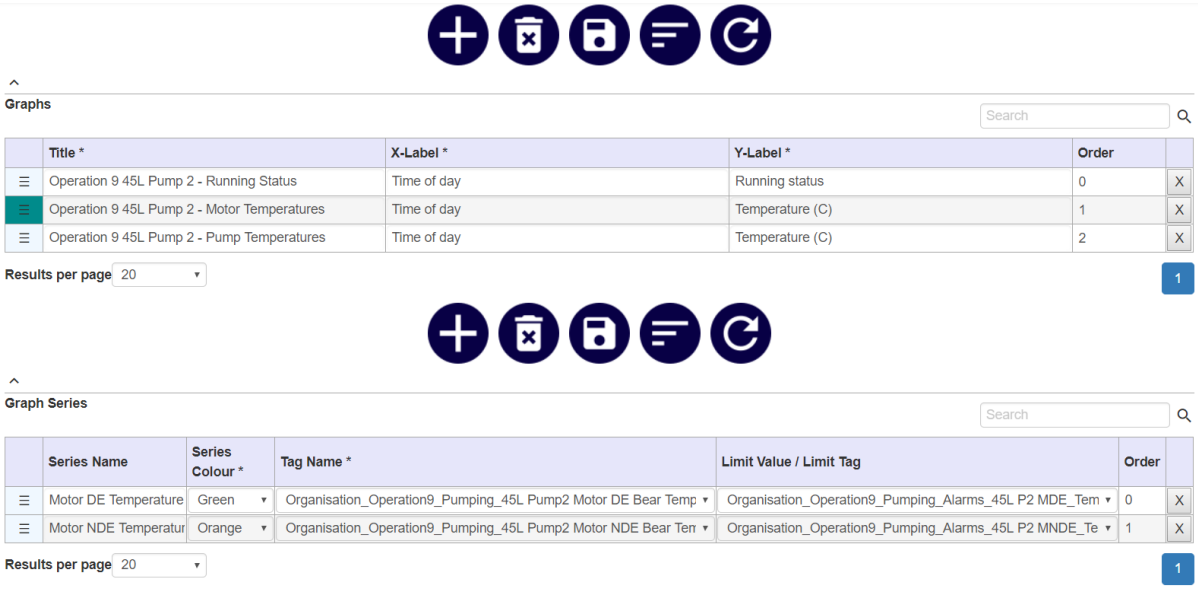


Figure 68: Node link graph configuration section

The basic graph display information can be configured in the graph subsection. The graph title and the various axis headings are simple text fields. The order in which the graphs of the selected node link are displayed is configured with an ordering dialog. The ordering dialog uses drag-and-drop style ordering of the graphs, using the graph titles as identifiers. Figure

69 presents an example of a completed graph subsection. Figure 70 presents an example of the graph ordering dialog. Graph ordering is done by dragging and dropping the graph titles.

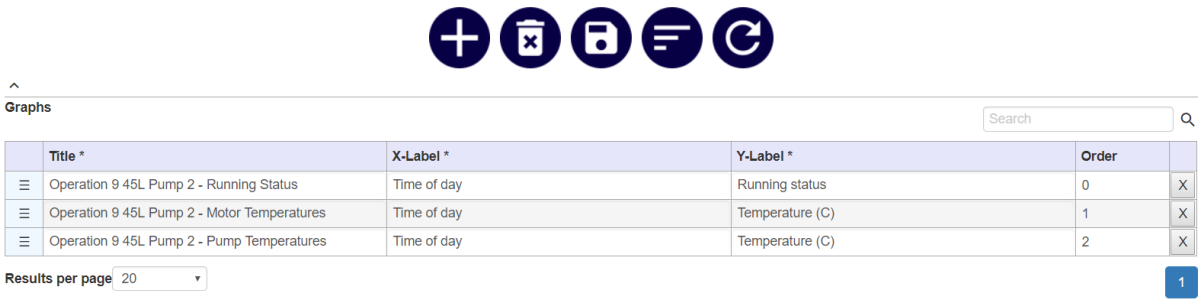


Figure 69: Node link graph only configuration

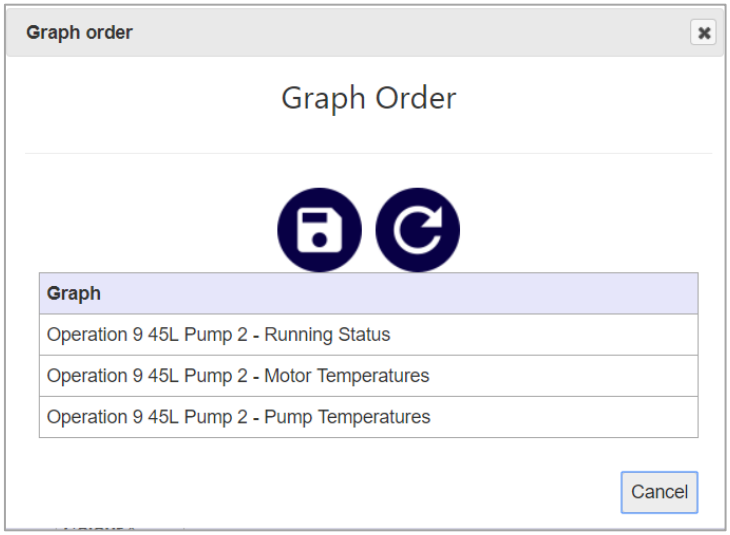


Figure 70: Graph ordering dialog

The series configuration subsection appears when an existing graph is selected in the graph configuration subsection. As discussed in Section 4.3.3, a series consists of a description, a hex colour, a data tag, an optional limit value, an optional limit tag and an order. Figure 71 presents an example of a completed series subsection.

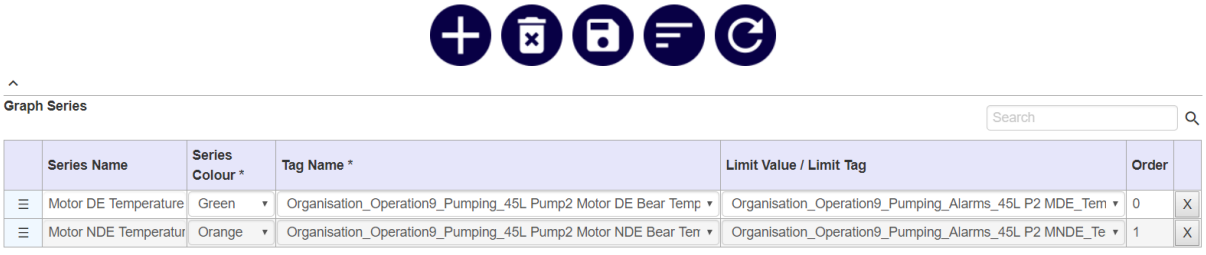


Figure 71: Node link series only configuration

The description field is a text field. The colour field lists various preselected hex colours as colour descriptions rather than hex codes. Selecting the data tag field opens a tag search dialog similar to the tag search dialog described in Section 4.4.3 and presented in Figure 66.

Selecting the limit field opens a series limit dialog with three options. These options are to have *no limit* for the series, to configure a constant limit *value* for the series, or to use a *tag* for the limit. Figure 72 presents the series limit dialog. Selecting a limit tag opens a tag search dialog similar to the tag search dialog described in Section 4.4.3 and presented in Figure 66. The order in which the series of the selected graph are displayed is configured with an ordering dialog, similar to the graph ordering dialog presented in Figure 70.

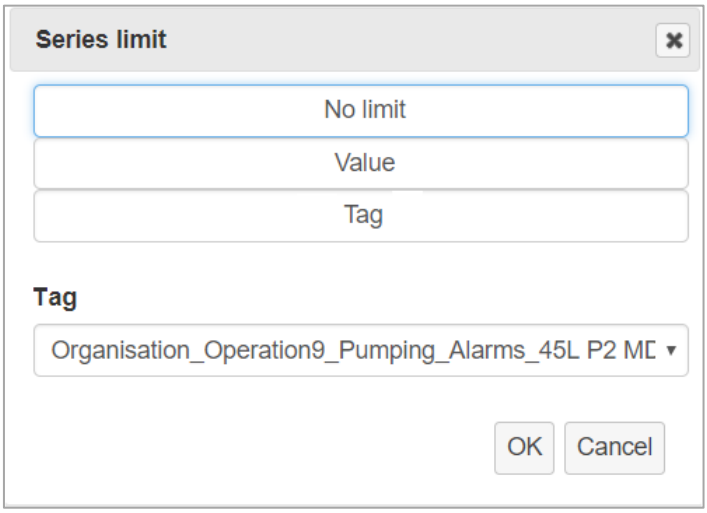


Figure 72: Series limit dialog

4.5 System logic

4.5.1 Overview

The system was developed in an extended version of the ASP.NET MVC architecture to include a data access layer (DAL) and view models (VMs). The DAL connects to the database and maps database tables to model objects. VMs are objects that bind to a view to transfer information from the controller to the view.

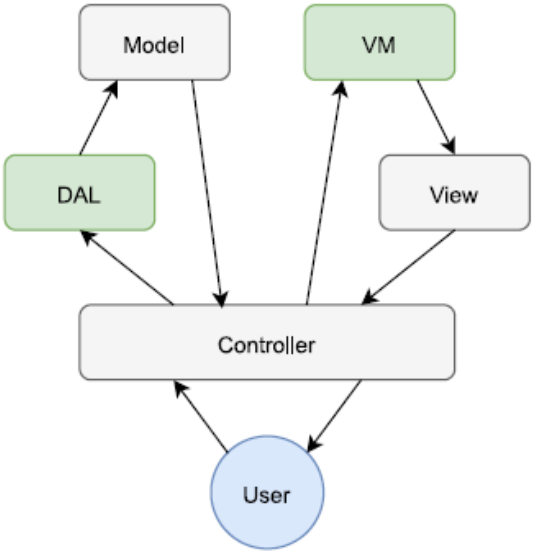


Figure 73: Extended ASP.NET MVC architecture [16]

Table 4 to Table 6 describe the properties of the applicable VMs. The properties of standard models correspond to the fields of their respective database tables. Section 4.3 presented the applicable database tables, which can be referenced for information on the applicable models. Models are identifiable by object types ending with “M”.

Table 4: Summary of applicable GridVM object properties

GridVM		
Property	**Type**	**Description**
BSNodes	List<NodeM>	Business structure (BS) nodes – used for the vertical list of the grid
DSNodes	List<NodeM>	Data structure (DS) nodes – used for the horizontal list of the grid
GridBlocks	List<GridBlockVM>	List grid blocks that represent node links

Table 5: Summary of applicable GridBlockVM object properties

GridBlockVM		
Property	**Type**	**Description**
NodeLinkID	uint	ID of the grid block – used by the client to get data from server
BSNodeID	uint	Linked data structure node ID – used during grid construction
DSNodeID	uint	Linked business structure node ID – used during grid construction
HasChildren	bool	Flag to help indicate if the block is clickable
HasCharts	bool	Flag to help indicate if the block is clickable
Condition	enum	Enumerable mechanical condition state of the represented component
DisplayInfo	GridBlockDisplayInfoVM	Display info view model object

Table 6: Summary of applicable GridBlockDisplayInfoVM object properties

GridBlockDisplayInfoVM		
Property	**Type**	**Description**
CSSClass	string	CSS class name for an applicable block colour
DisplayValue	string	Customised display value
Title	string	Customised tooltip text

4.5.2 Node link selection

The first and most common method of navigation in the front-end system is through node link selection. This occurs when the user selects a grid block and a deeper level loads. Figure 74 presents the process to load the deeper grid level from node link selection. Section 4.5.4 and Section 4.5.5 describe the grid construction and grid formatting processes respectively.

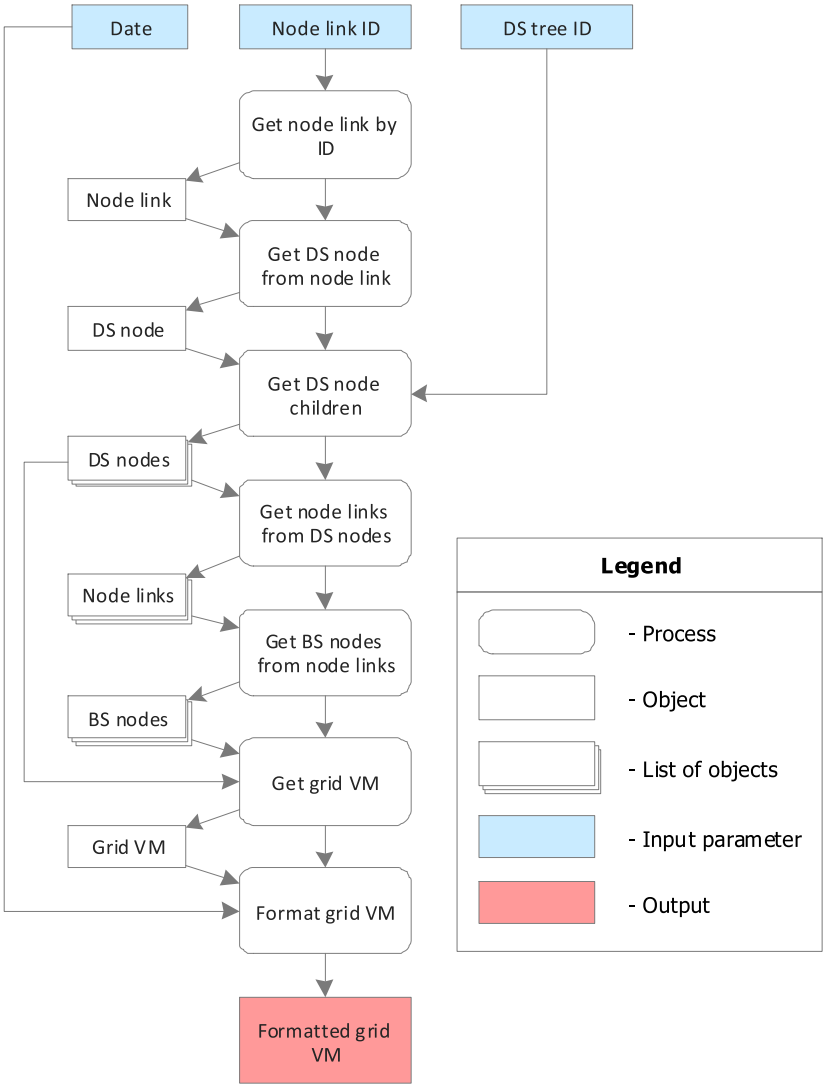


Figure 74: Process diagram of grid loading by node link selection

4.5.3 Node selection

The second method of navigation in the front-end system is through node selection. This occurs when the user selects a business structure node from the node picker and the applicable level loads. Figure 75 presents the process to load the correct grid level from node selection. Section 4.5.4 and Section 4.5.5 describe the grid construction and grid formatting processes respectively.

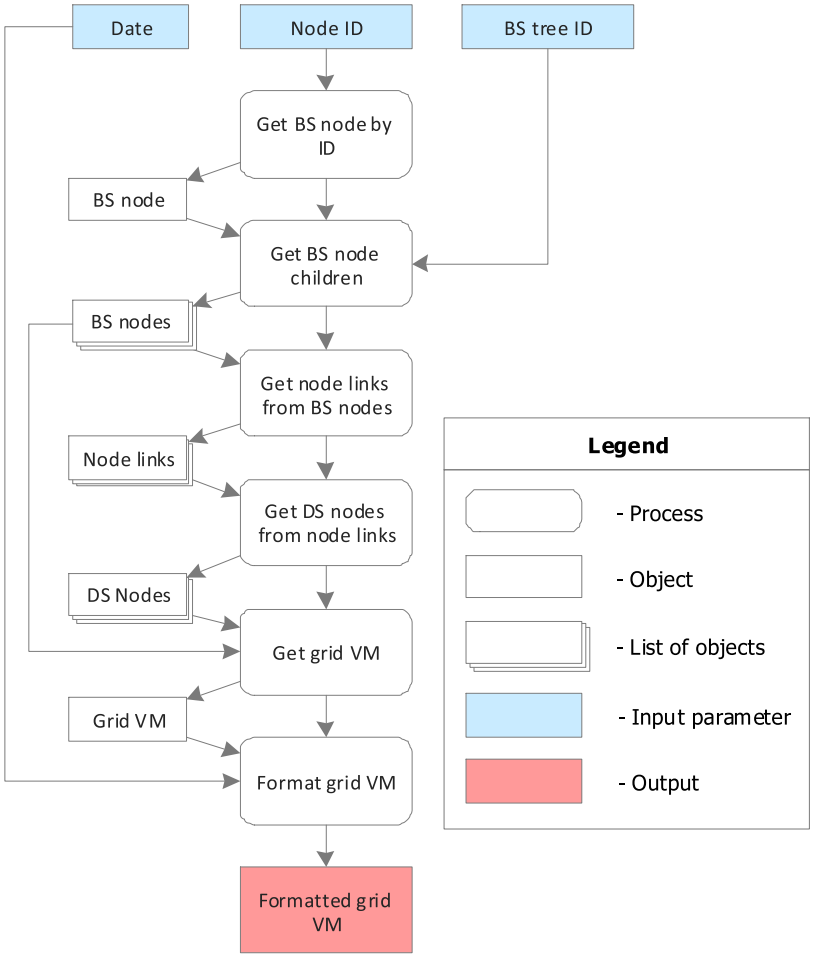


Figure 75: Process diagram of grid loading by node selection

4.5.4 Grid construction

All forms of navigation require grid construction. Figure 76 presents the grid construction process.

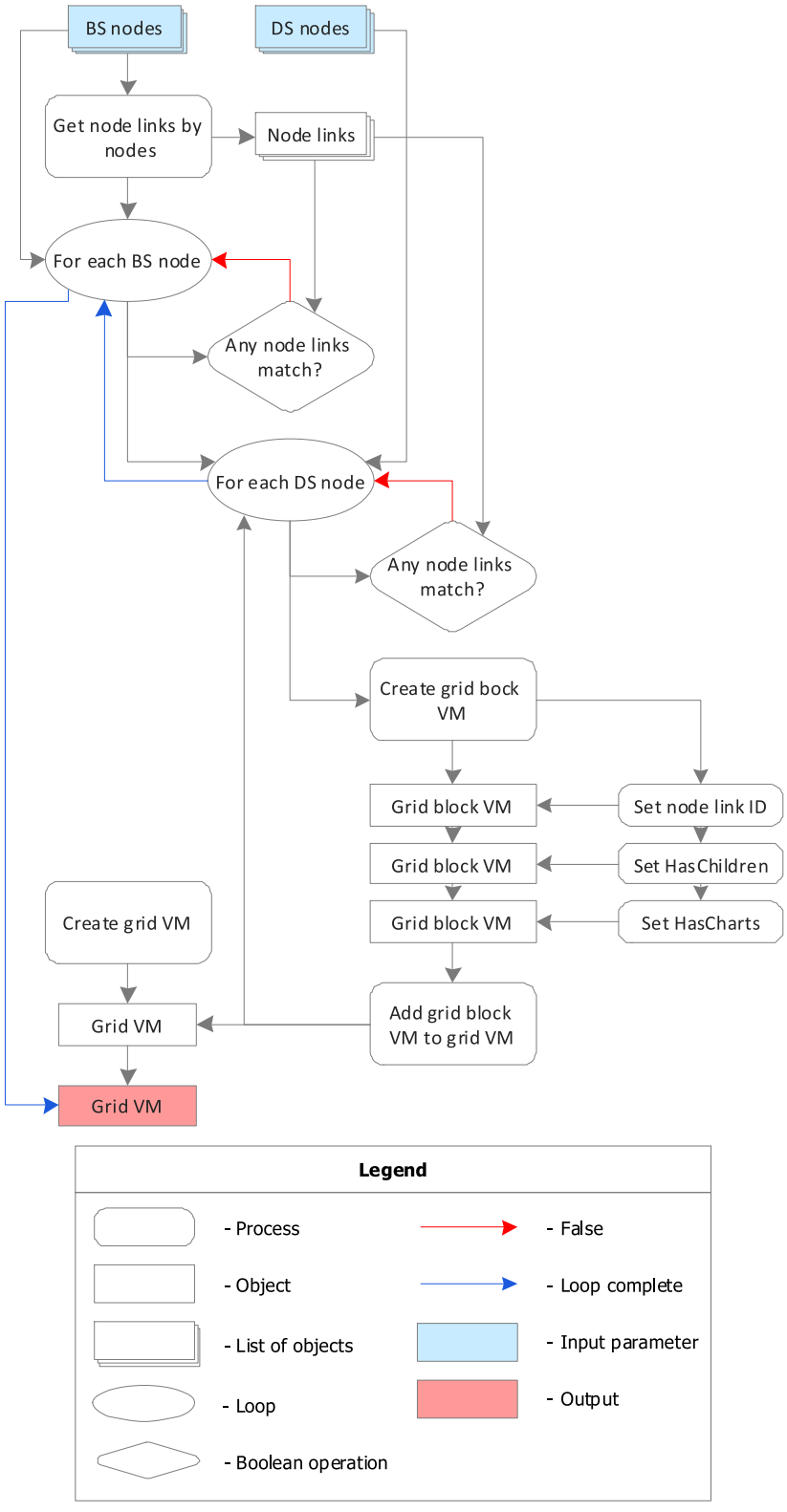


Figure 76: Process diagram of grid construction

4.5.5 Grid formatting

Once a grid has been constructed, it is formatted with the correct condition states, colour, display values and tooltip texts. Figure 77 presents the grid formatting process.

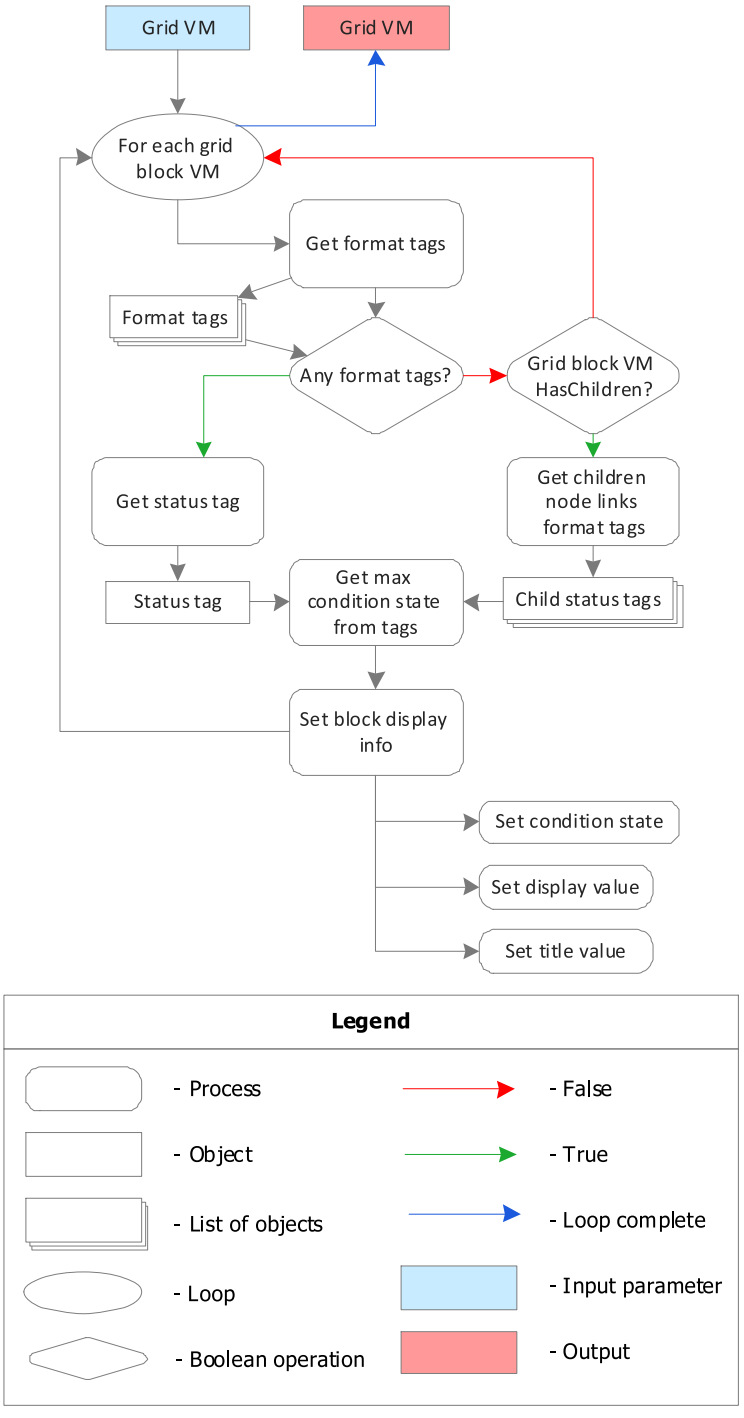


Figure 77: Process diagram of grid formatting

4.5.6 Node links with graphs

The front-end system uses FusionCharts to display raw data in the form of line graphs. FusionCharts receives a configuration object as input and handles the construction. Graphs load at the bottom of the user interface when the user selects a grid block with configured graph information. Figure 78 presents the graph configuration preparation process.

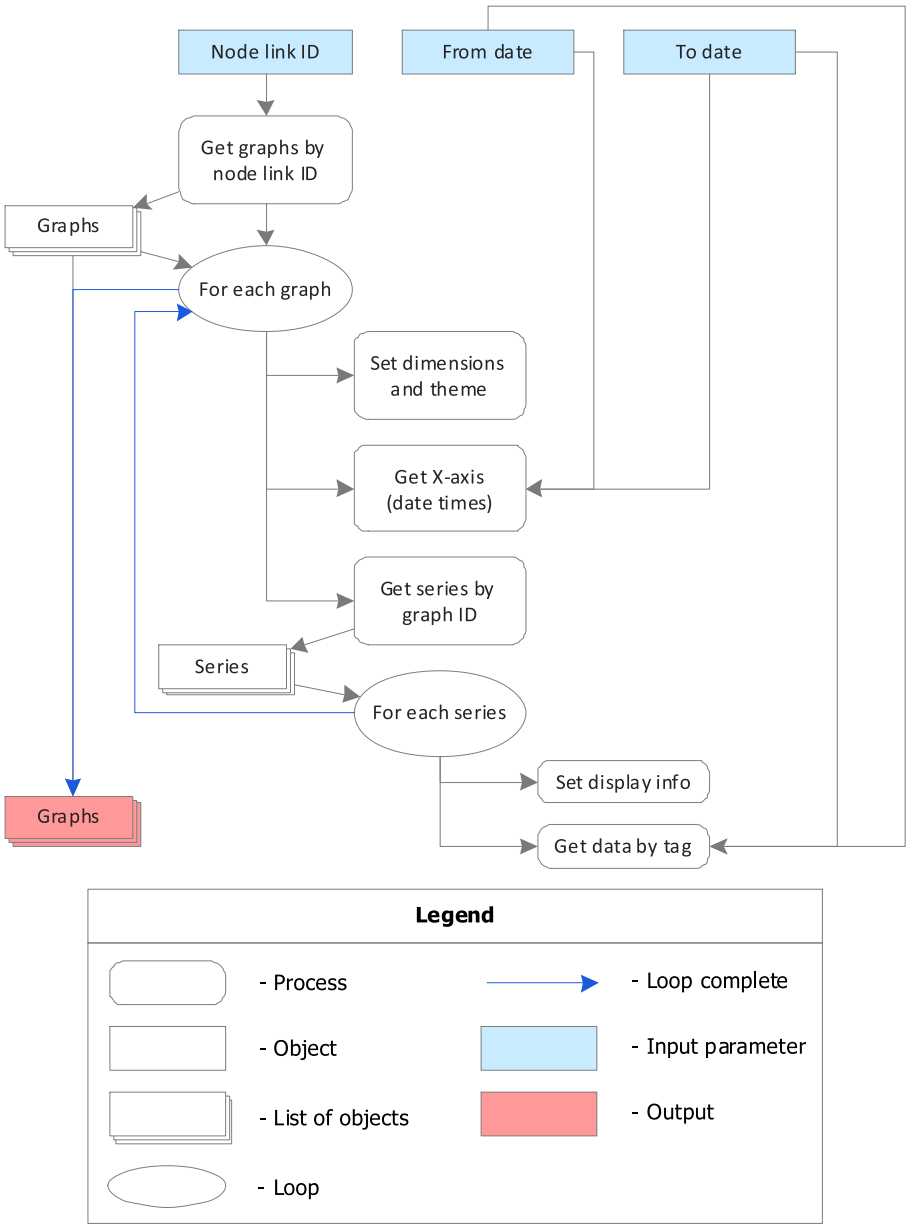


Figure 78: Process diagram of graph configuration for fusion charts

4.6 Verification

4.6.1 Accessibility

The system was implemented on an existing web-based EMS, as described in Section 4. Being web-based, the system is accessible through any modern web browser on any device with an internet connection. The system can thus be accessed remotely from almost anywhere in the world.

4.6.2 Integration

The system was implemented on an existing web-based EMS, as described in Section 4. This allows the system to utilise existing libraries and server resources. The integration with the EMS also gives the system access to existing relational database resources such as user authentication and user access control, which are described further in Section 4.6.5.

The system was configured using existing tools and a newly developed tool in an existing DBMS, as described in Section 4.4. The DBMS simplified development of the configuration tool by providing an existing connection to the relational database, user authentication and user access control.

Data can be obtained from various sources and in various forms, as described in Section 3.3. The system can therefore use existing measuring and logging infrastructure, which is installed on-site to reduce implementation resources. The generic approach to data processing, described in Section 3.3, also enables the system to utilise any data processing algorithm. The adaptability of the system to various data acquisition and data processing philosophies are described further in Section 4.6.7.

4.6.3 Information

The system provides the user with half-hourly averaged values of raw logged data. This gives the user access to near real-time raw data. Figure 79 presents an example of a condition monitoring graph presenting half-hourly averaged values. Note that the resolution of data is not a constraint of the front-end solution, but rather of the data acquisition and the relational database. For future work it is recommended that a direct connection between the system and the big-data database be developed. Big-data databases are much faster than relational databases when large data sets are handled and can therefore be used with finer resolution data.

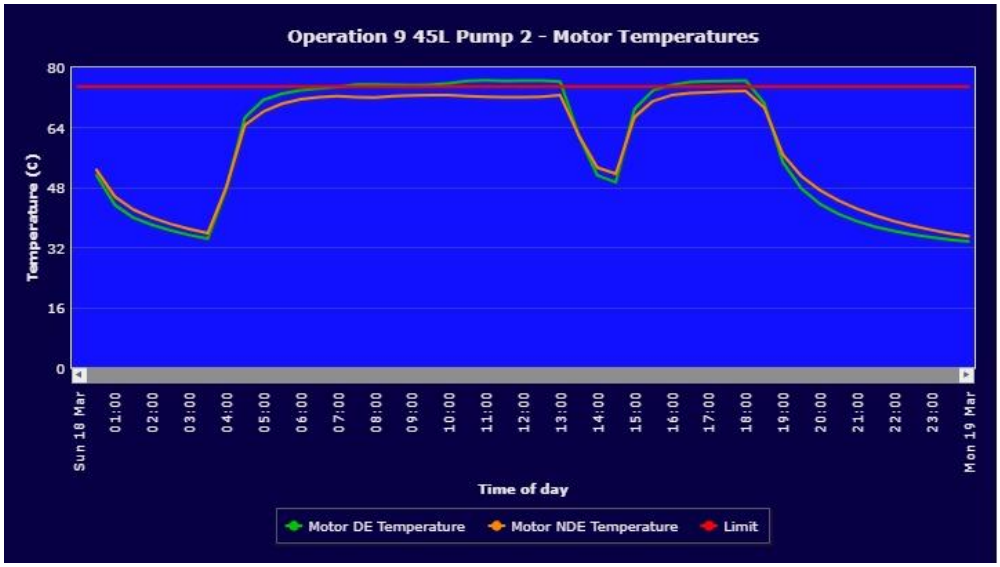


Figure 79: Graph presenting half-hourly average temperature values

Processed data typically takes on the form of the component health status. Component health status information is presented to the user in the form of coloured blocks or status blocks, as described in Section 4.2.2. Figure 80 presents an example of status blocks representing various component health statuses.

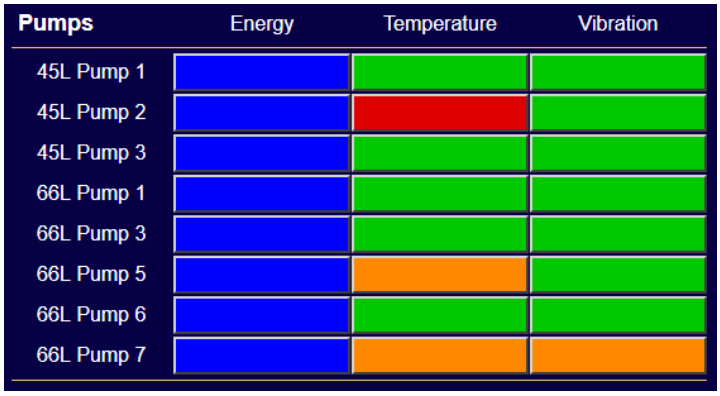


Figure 80: Status blocks representing pump health statuses

Processed data is aggregated to higher organisational levels to indicate the highest priority status information at organisational overview level. The aggregation of lower levels' processed data to higher organisational levels is described further in Section 4.6.4.

4.6.4 Multilevel

The system has multiple levels of detail. These levels typically include a parameter detail level, an operational overview level and an organisational overview level, which can be configured to the organisation's liking, as described further in Section 4.6.6.

The parameter detail level presents near real-time data to the user in the form of half-hourly averaged values. Figure 81 presents the parameter detail level of a pump's temperatures.

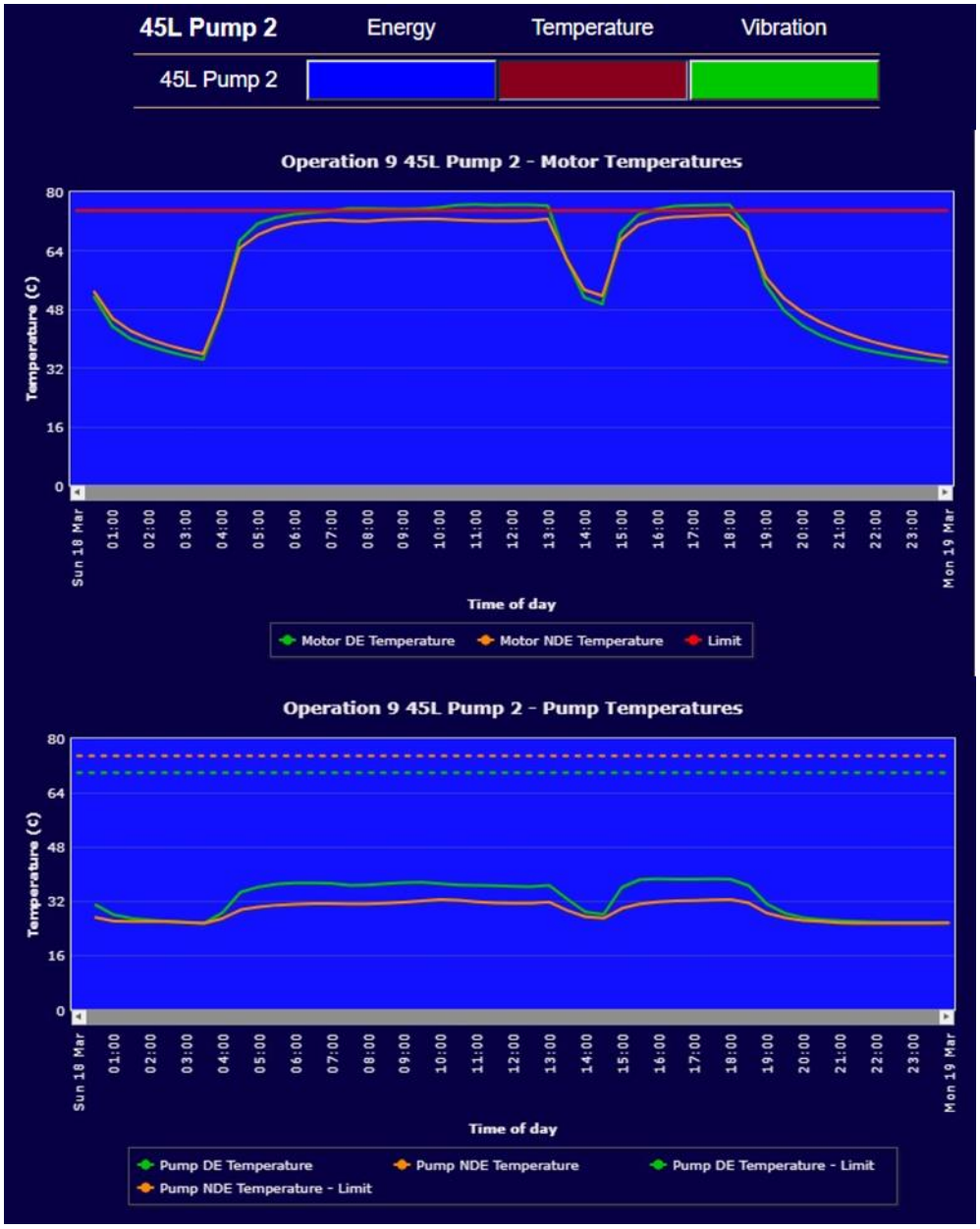


Figure 81: Parameter detail level of a deep level gold mine pump's temperatures

The organisational and operational overview levels present component parameter and operational system health statuses as coloured blocks. Each of these blocks can be used to navigate to a deeper or higher level of detail. The status of each overview level block is determined by each of its detail level blocks. Section 4.2.3 presented the navigation from an organisational overview to an operational overview level. Section 4.2.3 also described and presented the dependence of overview level status blocks on their respective detail level blocks.

4.6.5 Access control

The system limits the user to the information applicable to his/her position in the organisation. Users are linked to nodes in the business tree structure to control their access to information. If a user is linked to a node, the user gains access to all the information linked to that node and all of the information of the node's descendants. Figure 82 presents the options of a user with full access to the tree structure. Figure 83 presents the options of a user with limited access to the tree structure.



Figure 82: User with full access. Left: Condition monitoring node tree, Right: User access control setup



Figure 83: User with limited access. Left: Condition monitoring node tree; Right: User access control setup

4.6.6 Configurability

Configuring the system for first time use in an organisation requires setting up the organisational structure. Figure 84 presents the organisation's business and data tree structures, which are set up using the existing DBMS node builder tool. For demonstration purposes, a new organisation for machining services is created.

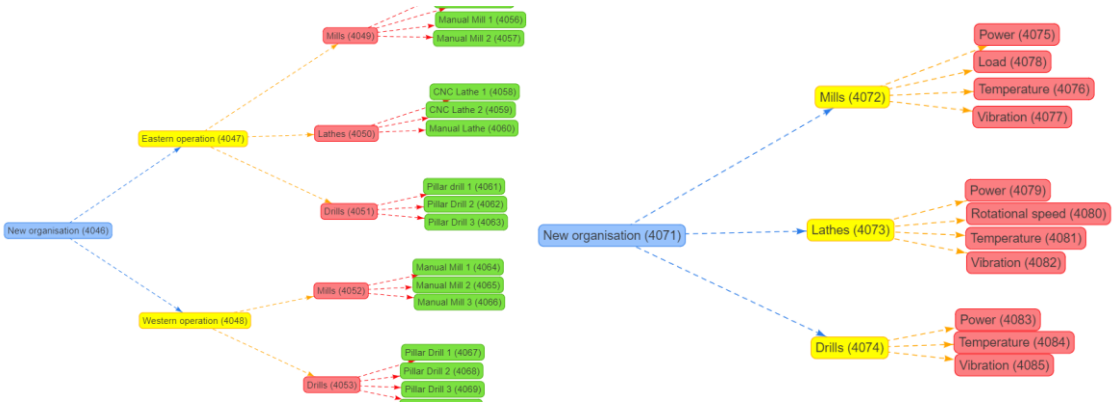


Figure 84: Left: New organisational condition monitoring business tree structure; Right: New organisational condition monitoring data tree structure

The next phase of the configuration process is to create node links between the applicable tree structure nodes. Section 4.4.2 described and presented the process of creating new node links. Figure 85 illustrates how to create a node link in the new organisation. Figure 86 demonstrates that the front-end system reflects the newly created node links. Note that these node links have no condition monitoring information and therefore have a no-data status. Figure 87 illustrates how a grid block becomes selectable when deeper level node links are created beneath it.

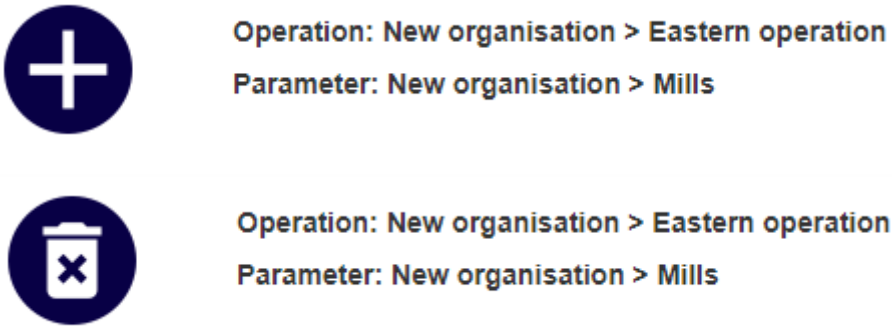


Figure 85: Top: Two nodes selected before creating a node link; Bottom: Two nodes selected after creating a node link



Figure 86: Node links on the front-end system

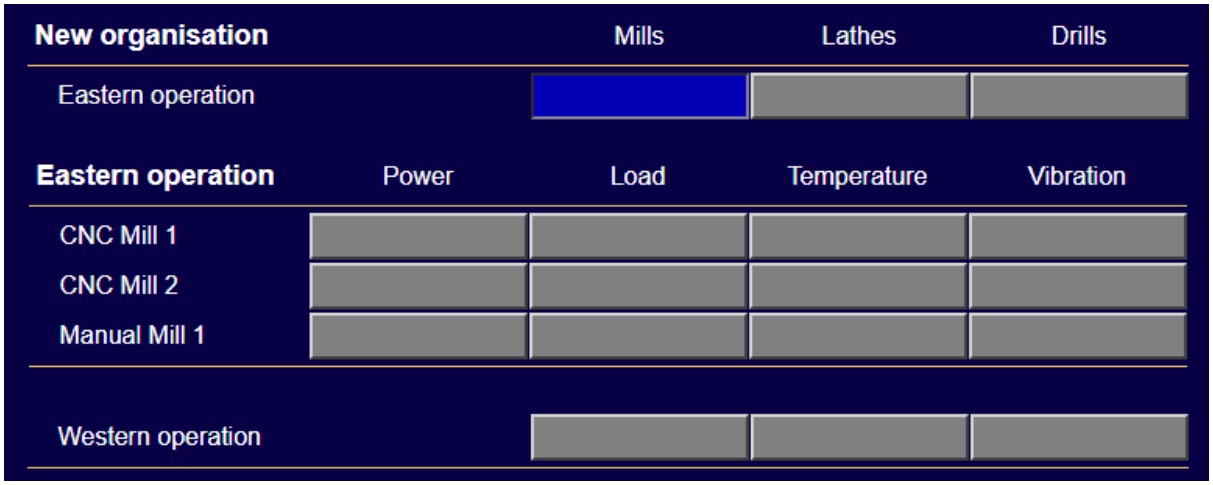


Figure 87: Node link with a deeper level of navigation

Once all the applicable node links have been created, information can be linked to the node links. Both status block formatting and graph information are added to the node links in the developed system configuration tool as described in Section 4.4. The node link status information links the front-end solution's status blocks to database status values. Figure 88 presents the effect of the status information configuration. Figure 89 presents the effect of customised tooltip messages and display values configuration.

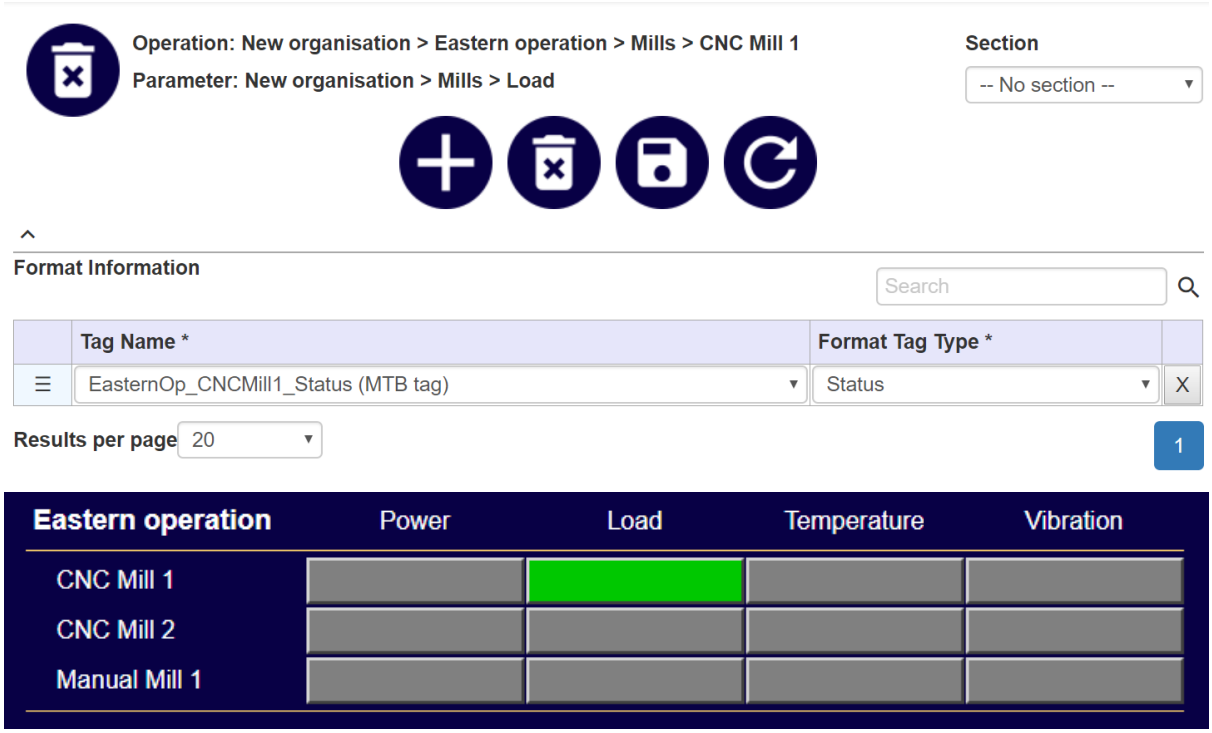


Figure 88: Top: Status information configuration; Bottom: Status on front-end system

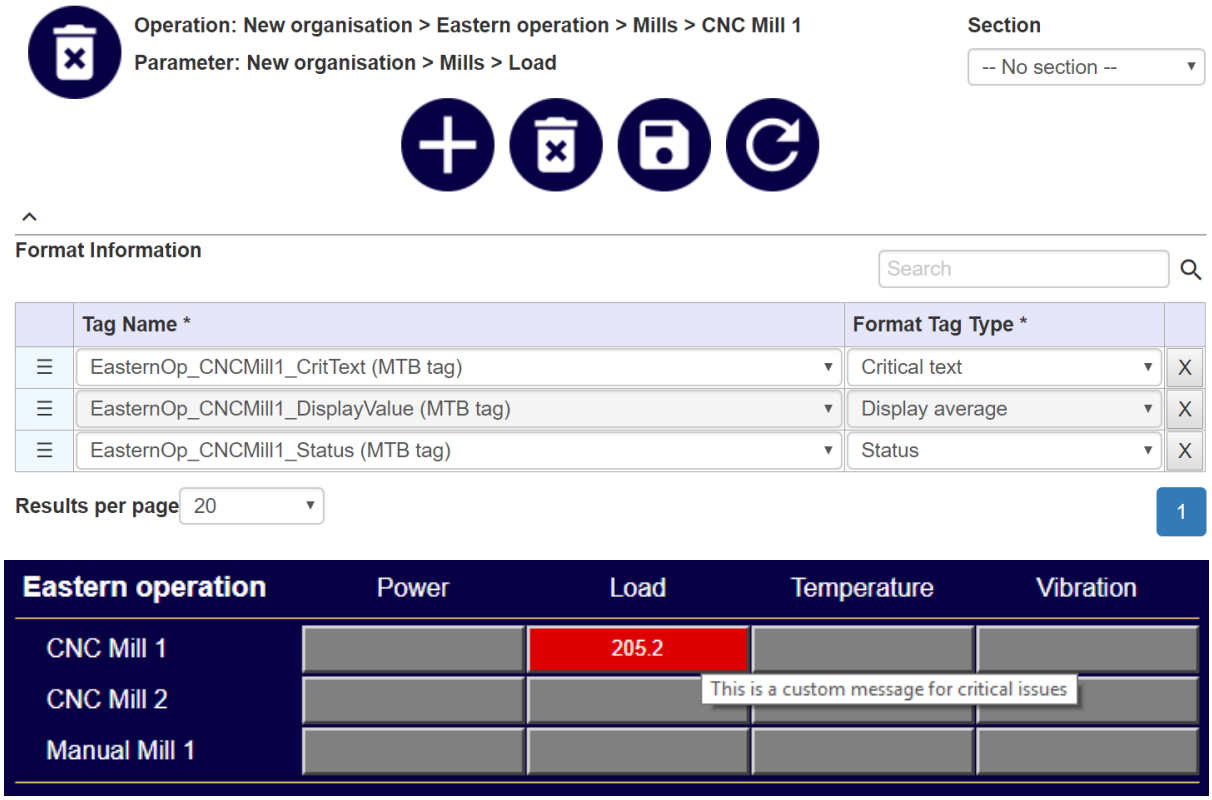


Figure 89: Top: Customised tooltip text and display value configuration; Bottom: Customised tooltip text and display value on front-end system

Graphs and series are also configurable for each node link, as described in Section 4.4. Figure 90 presents the effects of configuring a graph and series.

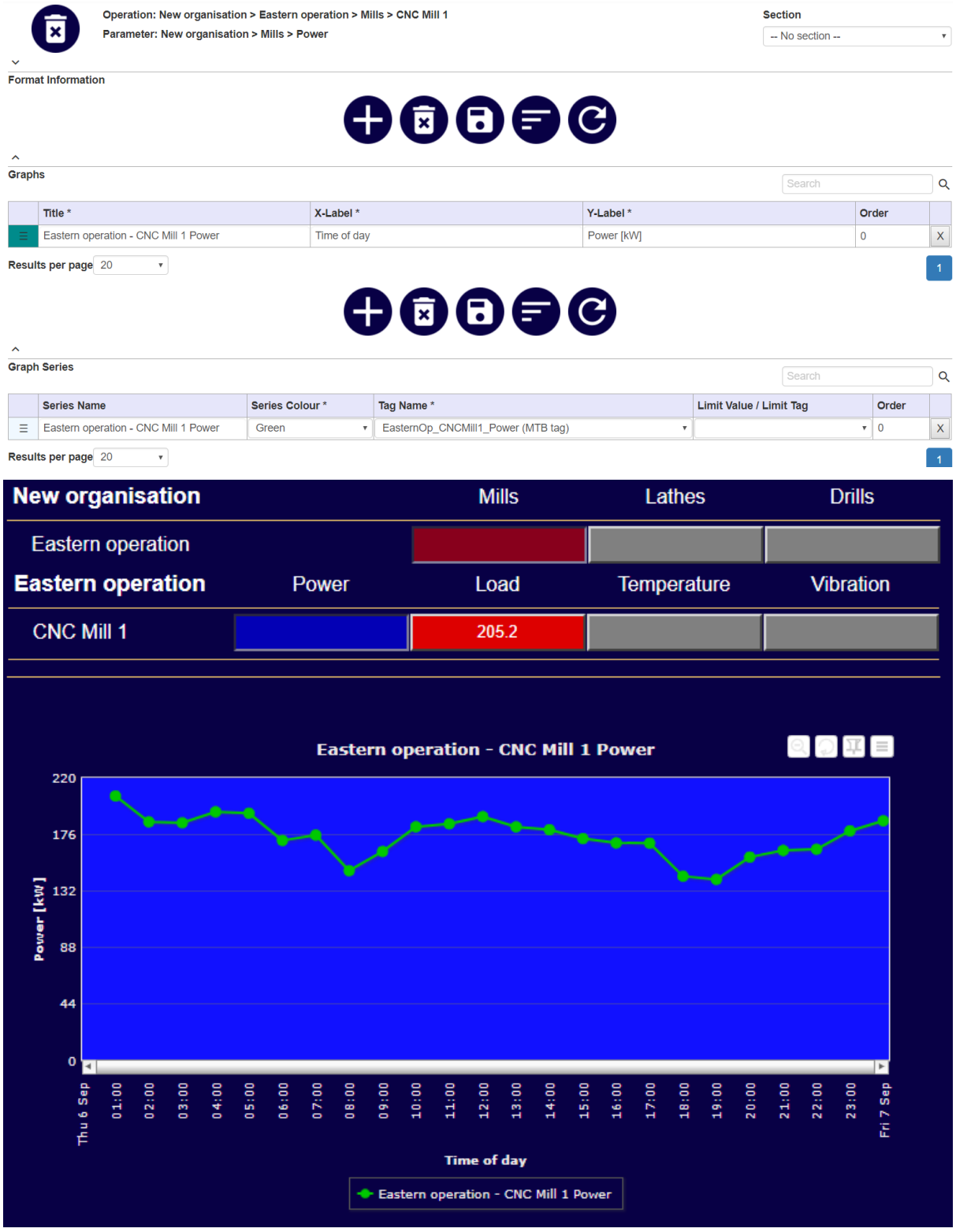


Figure 90: Top: Graph and series configuration; Bottom: Graph and series on front-end system

4.6.7 Adaptability

The system is adaptable in terms of data acquisition and data processing. Various data acquisition processes and sources can be used as long as the data is transferred to the relational database and linked to a tag. Chapter 3, Section 3.4.2 presented how the system can use various data acquisition processes and data processing techniques.

The type of data used in the system depends on the industry, organisation and types of components used. The system can be configured for numerous component types and, by extension, numerous parameter types. New components and parameter types are added to the system by creating new data tree structure nodes. Section 4.2 and the majority of this section presented an implementation of the developed system on a mining system. Section 4.6.6 presented the process of implementing the condition monitoring system in a new organisation in a different industry. Similarly, the system can be implemented in numerous industries.

4.7 Summary

The developed condition monitoring system consists of three main components, namely, a front-end system, a relational database and a configuration tool. The front-end system forms part of a web-based EMS as a tool. This is the component of the condition monitoring system that presents information to stakeholders. The front-end system incorporates user interface design principles that have been identified in the literature review in Chapter 2, Section 2.3.

The relational database integrates with the existing relational database of the EMS. The developed system can therefore leverage off the EMS's user access control and tree structures. Node links were developed to improve the scalability and maintainability of the tree structures in the condition monitoring system. Each node link represents a grid block on the front-end system. A node link's status data, display values and customised tooltip text are configurable. A node link can also display raw data in the form of line graphs. These graphs and their data sets are also configurable.

The configuration tool forms part of a DBMS. The DBMS offers functionality to configure tree structures. The configuration tool has the functionality to combine those tree structures and form node links. The configuration tool enables the user to set up format information for a node link. The configuration tool also has the functionality to configure graphs and series for a selected node link.

The front-end system's logic comprises four major elements. The system needs to load a grid based on either node selection or node link selection. Grid blocks without status information

checks deeper levels for status information. Finally, grid blocks with graphs need to load the applicable graph configuration information and series data.

The developed system was verified regarding the requirements defined in Chapter 3, Section 3.2. Each requirement was verified through a demonstration. This suggests that the foundation has been laid for the system to achieve the study objectives of reducing maintenance reaction times through cognitive load reduction and data accessibility while minimising system configuration time.

5 Results

5.1 Preamble

This chapter presents and discusses the results obtained from implementing the system in an industrial organisation. Section 5.2 gives background on the implementation of the system and the available resources at the time. Section 5.3 provides background on the organisation in which the system was implemented and the existing processes at the time. Section 5.4 validates that the system has met the objectives of the study, which were described in Section 1.6. The system is validated by presenting relevant results and interpreting the results regarding the relevant objectives.

5.2 Implementation

The condition monitoring system was implemented by an energy services company (ESCo) with an existing industrial client base. At the time of implementation, these clients did not have active condition monitoring solutions other than control room monitoring or SCADA systems. There are multiple advantages to using an ESCo with an existing client base to implement the system. These advantages include but are not limited to:

- **Existing infrastructure** – Remote logging systems are already implemented in most client operations, which in turn reduce implementation time and cost.
- **Existing resources** – The ESCo uses a web-based EMS. An overview of the typical functionality of an EMS was provided in Chapter 3, Section 3.3.
- **Existing customer relations** – Professional relationships and trust between the ESCo and the industrial clients have been built up over years of service. This professional trust can prove invaluable for motivating the implementation of a new condition monitoring system.

A prototype version of the condition monitoring system was developed and implemented for two months prior to the development of the official system. The prototype system was developed using a rapid development approach instead. This enabled the system's user interface, functionality and data processing algorithms to be refined continuously, but restricted scalability. The prototype was a useful tool for defining requirements for the official system.

5.3 Background on case study

The system was implemented in a South African gold mining organisation that has been operational for more than 65 years. At the time of writing, the mining organisation had more

than 30 operations. These operations can be categorised as office parks, mining shafts and gold plants.

The condition monitoring system was implemented in nine mining shafts on four systems: compressed air, refrigeration, dewatering or pumping, and ventilation systems. Implementation of the condition monitoring system across the nine operations occurred over a period of approximately one year.

The prototype version of the condition monitoring system was implemented on two operations for the first two months prior to developing the official system. The official system replaced the prototype system on the two operations for which implementation was already done and the system was expanded to the additional seven operations over the following 10 months.

The condition monitoring system have been implemented on a total of 28 operational systems across the nine operations. The condition monitoring system monitors a total of 201 components across these nine operations. The measured parameters typically vary from system to system but can generally be grouped into one of three categories, namely, energy, temperature or vibration. In total, the condition monitoring system presents the data and information of more than 1600 measurements.

5.4 Validation

5.4.1 Cognitive load

5.4.1.1 Objective

This subsection aims to validate the following study objective, stated in Section 1.6:

“Reduce the cognitive load placed on management level staff.”

5.4.1.2 Data

The system significantly reduces the cognitive load at each of its navigation levels and thus reduces the amount of information that management level staff need to analyse. Table 7 summarises the total amount of information that the condition monitoring system presents daily. The remainder of this section illustrates how the various levels of navigation reduce this large volume of information to more manageable quantities.

Table 7: Summary of the amount of information that the system monitors

Organisation	
Variable	**Count**
Operations	9
Systems	28
Components	201
Parameters	528
Tags (data only)	1 690
Data Points	81 120

Average per operation	
Variable	**Count**
Systems	3.1
Components	22.3
Parameters	58.7
Tags (data only)	187.8
Data Points	9 013.3

Each navigation level categorises applicable information in the form of a grid. Colours are used to indicate the highest priority mechanical health status of each grid element. Colour codes assist with cognitive load reduction in two ways. Colours draw the attention of the user to the areas of high importance, which naturally reduces the amount of visual processing required from the user. Green (good) grid blocks require no further inspection or investigation and can therefore be ignored. The number of elements on which the user is required to focus is thus reduced.

The organisational overview navigation level categorises each operation into its applicable systems. Figure 91 presents an example of the organisational overview level. Figure 92 and Figure 93 compare the number of parameters represented by the organisational overview level with the number of elements on the user interface. These elements include status blocks, status blocks with status information (formatted status blocks), and the daily average number of warnings and critical status blocks. Figure 92 presents the data along a linear scale to get a feel for the scale of the data. Figure 93 presents the data along a logarithmic axis to view the values of the data with greater accuracy.

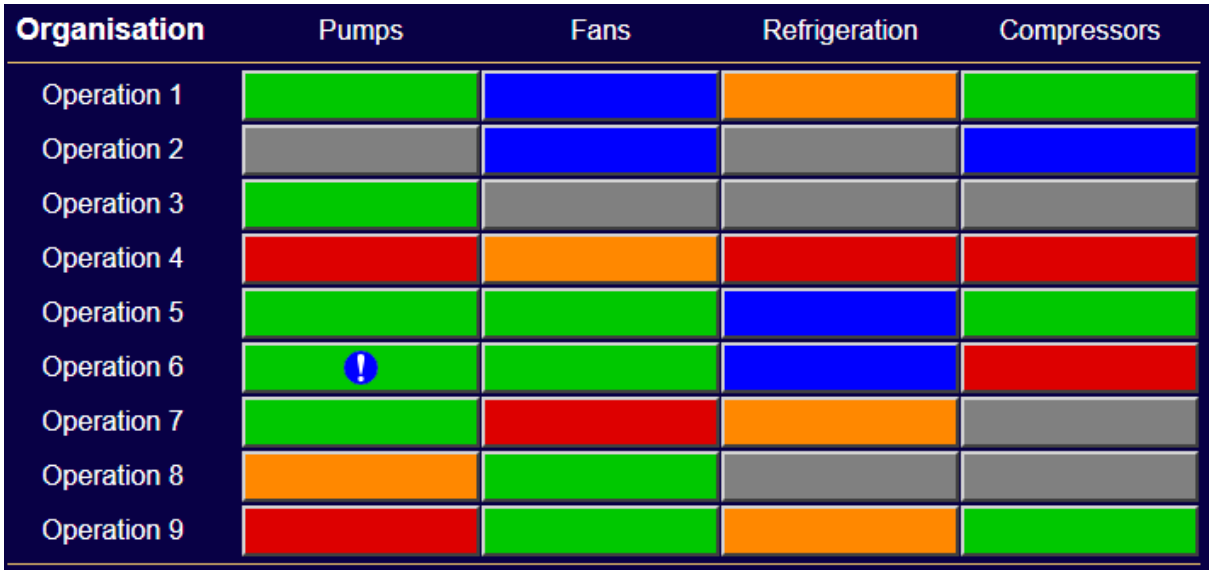


Figure 91: Organisational overview level

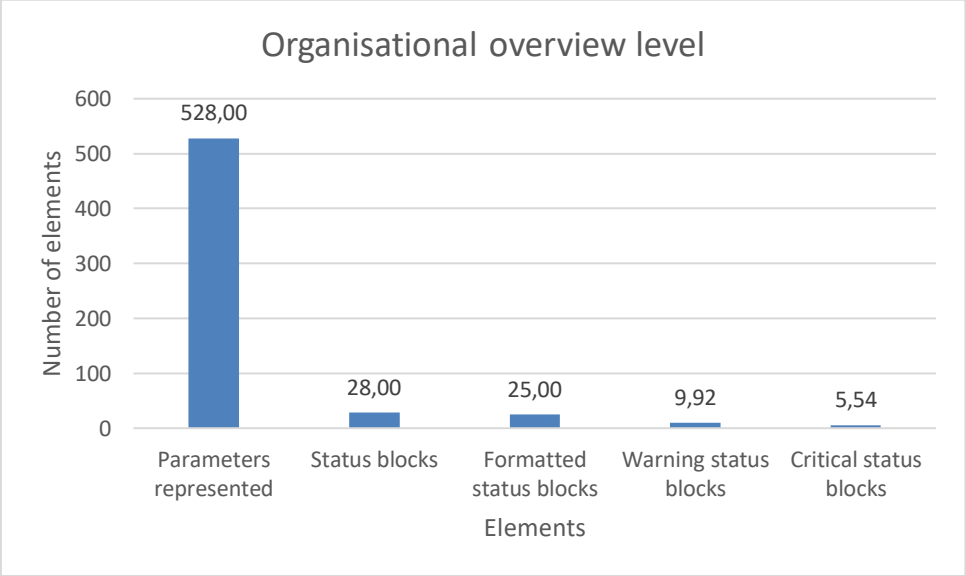


Figure 92: Linear comparison of the number of parameters and the number of visual elements at the organisational overview level

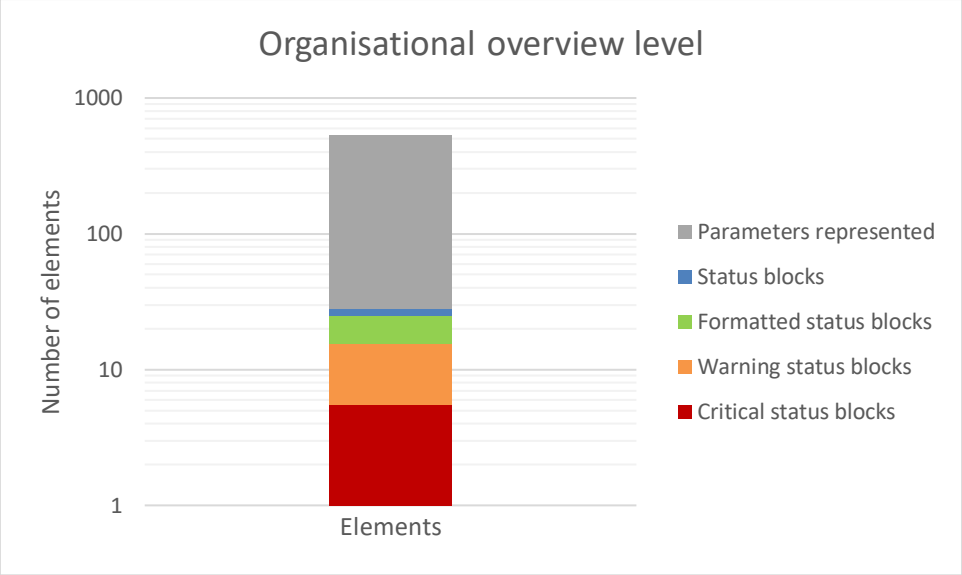


Figure 93: Logarithmic comparison of the number of parameters and the number of visual elements at the organisational overview level

The operational system overview navigation level categorises each component of a selected system in an operation into its applicable parameters. Figure 94 presents an example of the operational system overview level. Figure 95 and Figure 96 compare the average number of parameters represented by the operational system overview levels with the average number of elements on the user interface per operational system overview level. These elements include status blocks, status blocks with status information (formatted status blocks), and the daily average number of warnings and critical status blocks. Figure 95 presents the data along a linear scale to get a feel for the scale of the data. Figure 96 presents the data along a logarithmic axis to view the values of the data with greater accuracy.

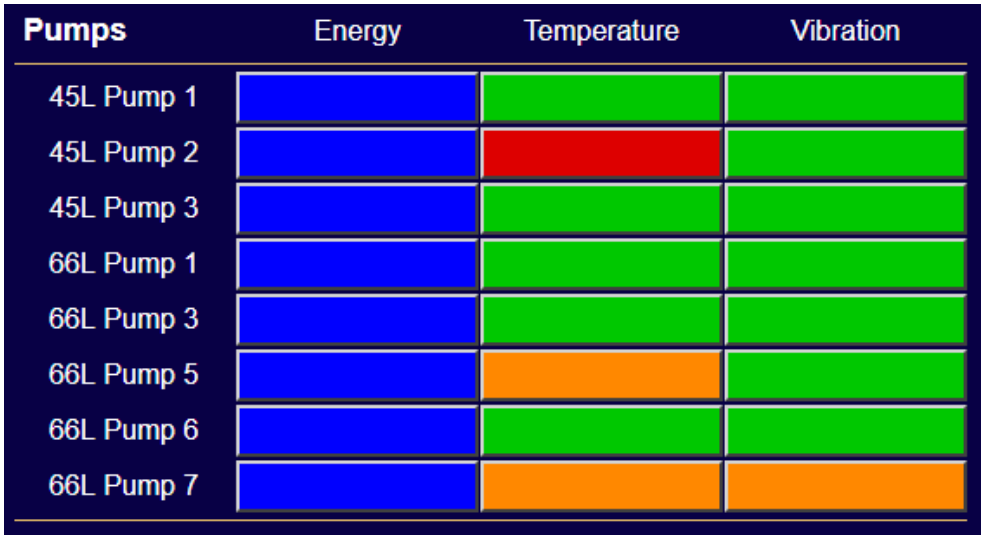


Figure 94: Pumping operational system overview level

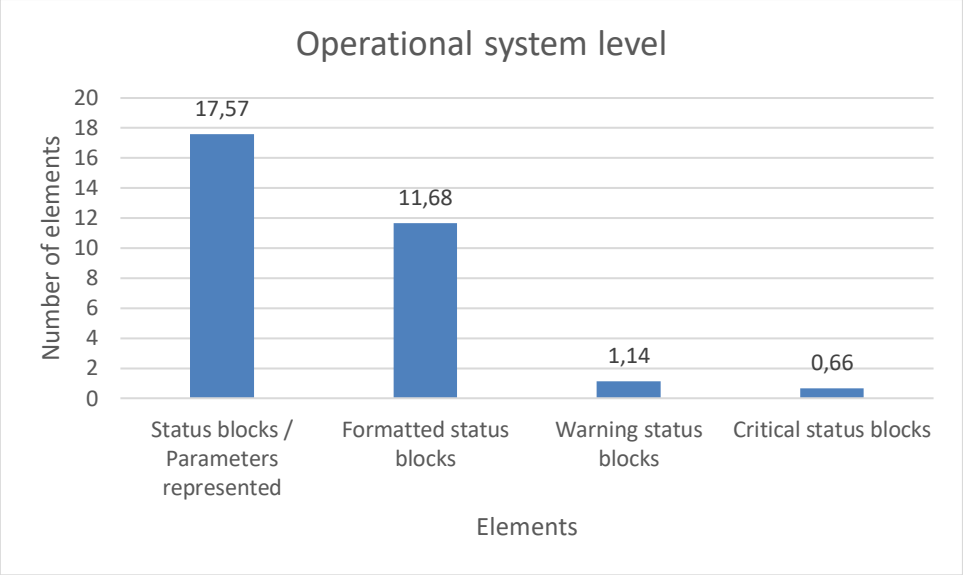


Figure 95: Linear comparison of the average number of parameters and the average number of visual elements at operational system overview levels

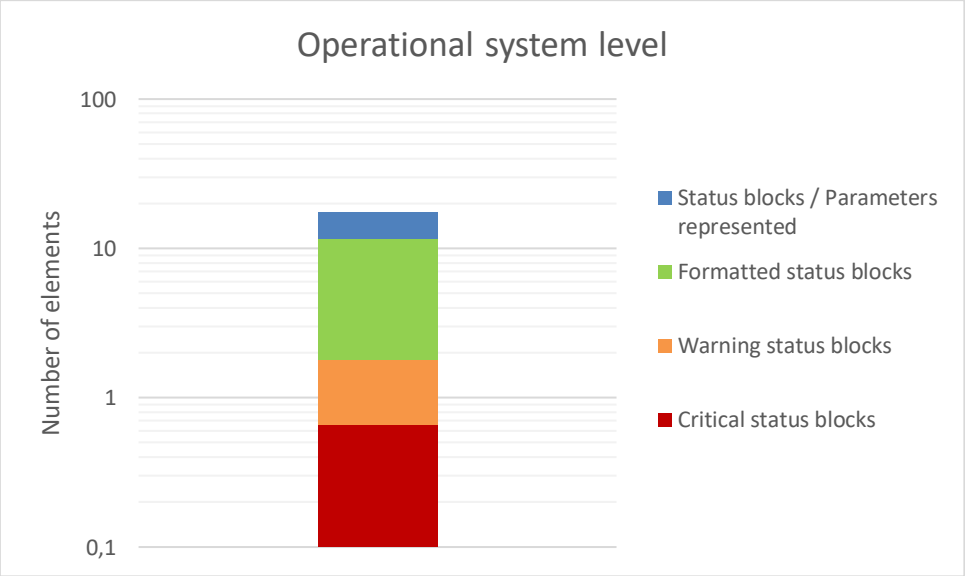


Figure 96: Logarithmic comparison of the average number of parameters and the average number of visual elements at operational system overview levels

The component parameter detail navigation level presents a series of time-domain graphs of the parameter category. Figure 97 presents an example of the component parameter detail level. The purpose of this level is to provide the user with raw data of the various measurements across the component. Limits are used to clearly indicate if a parameter is operating close to, or over its predefined limit. Cognitive load is limited at this level by limiting the number of series per graph to two. It is further recommended that the number of graphs that is set up for a parameter group should not exceed five, as motivated in Chapter 2, Section 2.3.1. Additional parameter groups should rather be created at operational system overview

level to avoid cognitive overload at component parameter detail level. Figure 98 and Figure 99 compare the daily average number of data points presented by the parameter detail level with the average number of series and graphs per component parameter detail level. Figure 98 presents the data along a linear scale to get a feel for the scale of the data. Figure 99 presents the data along a logarithmic axis to view the values of the data with greater accuracy.

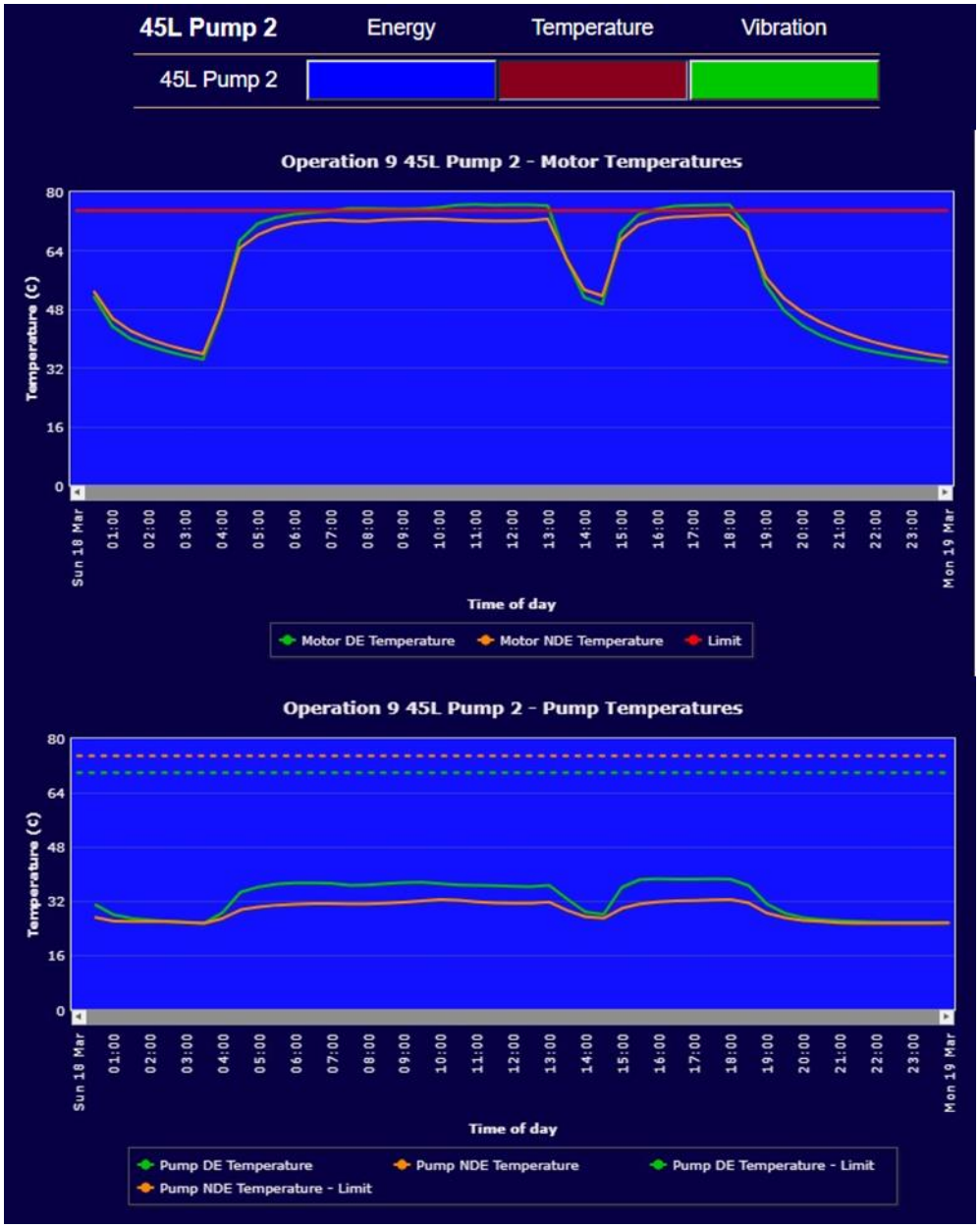


Figure 97: Pump temperature parameter detail level

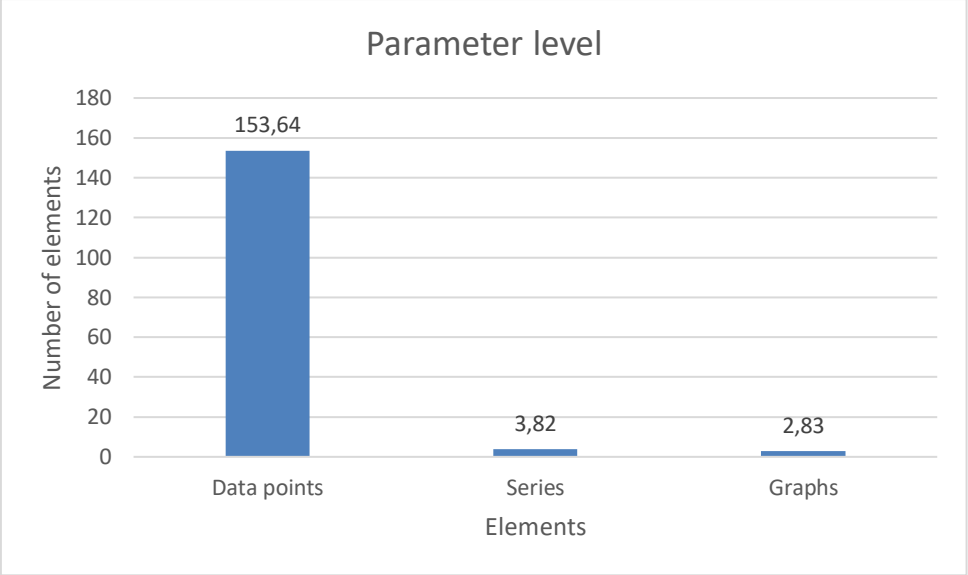


Figure 98: Linear comparison of the average number of data points and the average number of visual elements at parameter detail levels

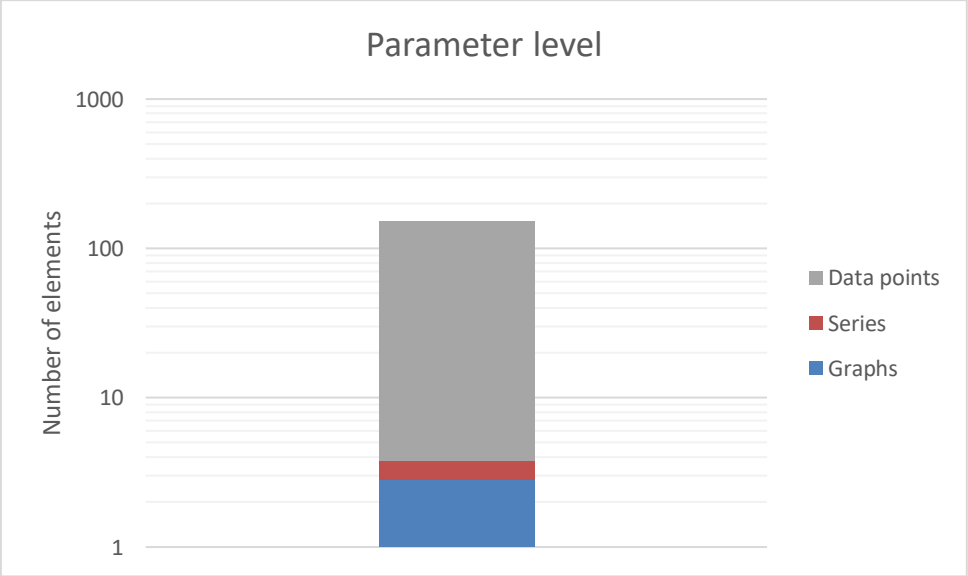


Figure 99: Logarithmic comparison of the average number of data points and the average number of visual elements at parameter detail levels

5.4.1.3 Interpretation

The organisational overview level condenses an immense number of parameters and data points, as presented in Table 7, into a summary of the mechanical operating conditions of the various operational systems, as presented in Figure 91. Colour codes assist in drawing the user's focus to high priority operational systems. High priority elements are typically warning and critical status blocks, with additional emphasis on critical status blocks.

Figure 92 illustrates the scale of reduction in cognitive load at organisational overview level. The 528 parameters represented by the organisational overview have been reduced to 28 elements. Figure 93 indicates that the number of critical status blocks is on average almost two orders of magnitude smaller than the overall number of parameters represented.

The organisational overview level consists of 28 status blocks or operational systems. Of these 28 operational systems, an average of 15.4, or 55%, are high priority operational systems. Furthermore, an average of only 6.6, or 24% of the operational systems, are critical operational systems.

The organisational overview level reduces cognitive load by reducing the amount of information presented to the user significantly. Due to the sheer amount of data represented by the overview level, the number of elements requiring the user's attention is still a bit high. Further investigation for possible improvements and optimisation is recommended.

The operational system overview level indicates the mechanical operating conditions of the various parameters of the selected operational system, as presented in Figure 94. Colour codes assist in drawing the user's focus to high priority operational systems. High priority elements are typically warnings and critical status blocks with additional emphasis on critical status blocks.

At the operational system overview level, each status block represents a parameter. Figure 95 illustrates the scale of reduction in cognitive load at the operational system overview level for an average operational system. Figure 96 indicates that the number of high priority status blocks is on average a full order of magnitude smaller than the overall number of parameters represented.

The average operational system consists of 17.6 status blocks or parameters. Of these 17.6 parameters, an average of 1.1, or 6.3%, are high priority parameters. Furthermore, an average of only 0.66, or 3.8% of the parameters, are critical parameters.

The operational overview level reduces cognitive load by presenting all the applicable information to the user and drawing the user's focus to high priority elements on the user interface. The average number of high priority elements per operational overview level is acceptably low and is a significant reduction from the amount of information presented to the user.

The parameter detail level presents the raw data of a parameter for a selected operational system as presented in Figure 97. The main purpose of this navigation is not to reduce

cognitive load, but to provide information; however, cognitive overload should be avoided. This is achieved with graphs and graph limits.

Graphs condense large data sets into visual lines, which are much easier to interpret. Figure 98 illustrates the scale of reduction in cognitive load at the parameter detail level for an average operational system parameter.

Figure 99 indicates that the number of series per parameter detail level is on average 1.5 orders of magnitude lower than the number of data points presented. Figure 99 also indicates that the average number of graphs per parameter detail level is less than three and that the average number of series per parameter detail level is less than four. This is an acceptably low number of data sets requiring the user's focus, which will not cause cognitive overload.

Graph limits further assist in preventing in cognitive overload. Parameters that are operating above their design limits are clearly visible and draws the user's attention.

At all navigation levels, the cognitive load has been reduced significantly. The overview levels reduce large volumes of data into colour-coded blocks that indicate the severity of the health status of equipment. These colour codes assist stakeholders to focus only on the highest priority information. The number of elements that require the user's attention is acceptable at organisational level and ideal at operational system level. The parameter detail level's purpose is not to explicitly to reduce cognitive load, yet it does not induce cognitive overload. The first objective of the study has therefore been achieved.

5.4.2 Reaction time

5.4.2.1 Objective

This subsection aims to validate the following study objective, stated in Section 1.6:

"Reduce maintenance reaction times."

5.4.2.2 Data

The system provides early warnings against equipment failure by detecting and informing management personnel about developing faults during operation. The reaction time for maintenance can be defined as the number of days since a warning has been issued about a component's condition up to when corrective action is taken. A component with a warning or critical status that returns to a good status is an indication that maintenance was applied to the component. Figure 100 illustrates the way maintenance reaction time is measured for the purposes of this study.

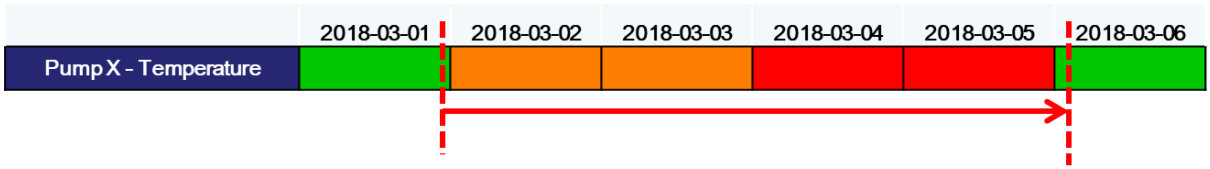


Figure 100: Maintenance reaction time measurement

The number of high priority parameters can influence the average reaction times. The more parameters operating at high priority conditions at any given time, the more maintenance resources are required. With finite maintenance resources, it is expected that maintenance reaction times increase with an increase in high priority parameters. Table 8 summarises important concepts for interpretation of this subsection.

Table 8: Concepts applicable to the results of this subsection

Concept	Description
Status count	The number of parameters for which health status calculations were done
Alerts count	The number of high priority parameter indicators during the month
Reaction count	The number of corrective actions taken during the month
Average reaction time	The average time for corrective action on a high priority parameter

Figure 101 presents the monthly average reaction time for equipment maintenance across the entire organisation. In addition to the reaction time, Figure 101 presents the number of high priority parameters (alert count), the number of parameters on which maintenance was done (reaction count), and the overall number of parameters (status count) across the organisation. Figure 102 compares the reaction time data, presented in Figure 101, with the reaction count across the entire organisation. Figure 103 presents the average monthly reaction times relative to the number of reactions for each month at organisational level.

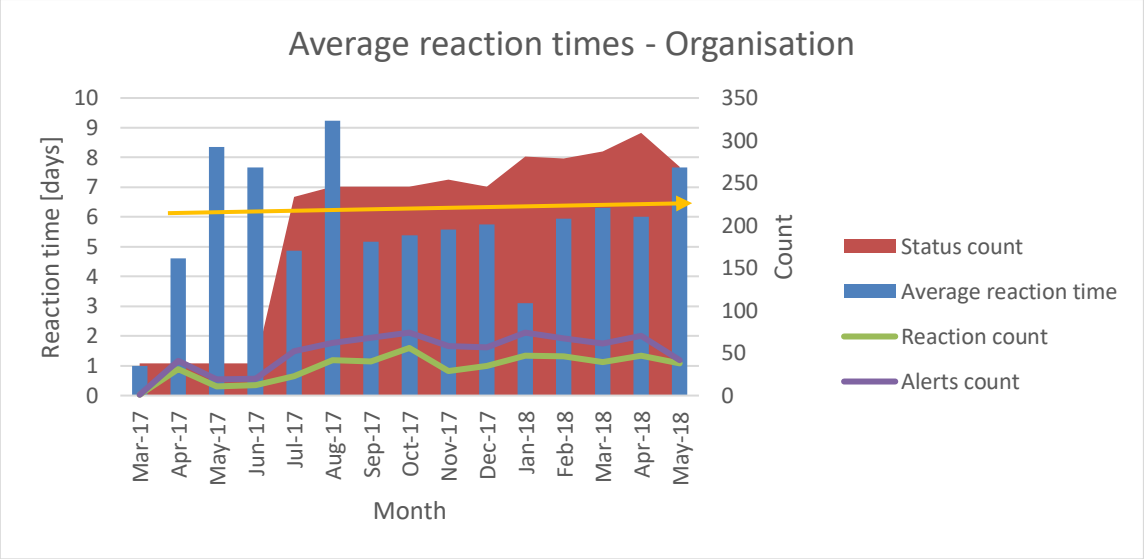


Figure 101: Average reaction times for the organisation over 15 months

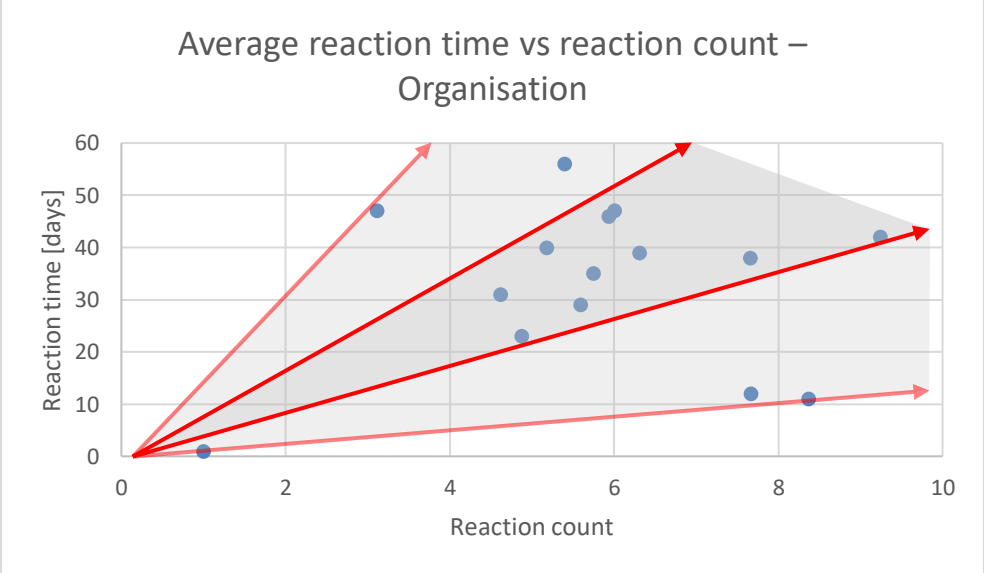


Figure 102: Average reaction times vs reaction count for the organisation over 15 months

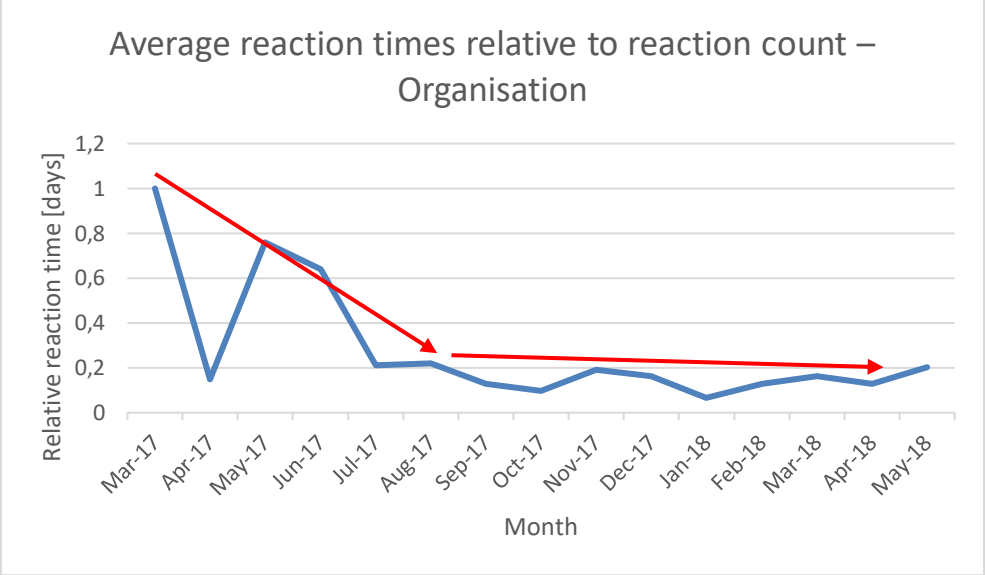


Figure 103: Average reaction times relative to the reaction count for the organisation over 15 months

Figure 101 indicates no clear upward or downward trend over time in the average maintenance reaction times across the organisation. A significant drop in the reaction time can be observed at the fifth month (July 2017) of implementation. This is largely due to the significant increase in operations on which the system was implemented, which is evident by the increase in the overall status count. This increase in the overall status count pulls the overall average reaction time down. Individual operational investigations will provide better insight into the average reaction times during this period. The spikes at the sixth month (August 2017) and the 15th month (May 2018), and the dip at the 11th month (January 2018) seem to be anomalies when compared with the rest of the data.

Figure 101 further does not seem to show a clear correlation between the number of alerts or reactions and the average reaction time. This is further supported by Figure 102 that shows poor correlation between the number of reactions and the average reaction time (area between two outer arrows). It is expected that more reactions would overload the available maintenance resources and cause longer maintenance reaction times with at least some rough linear correlation. The fact that no correlation is evident at organisational level is potentially due the sheer number of factors influencing the maintenance reaction times. Some of these factors include: some operations or systems taking preference over others regarding maintenance resources; the size of operations; the cost of some systems; the availability of parts; and human factors such as management and maintenance philosophies. This is supported by the improved correlation when some of the outliers are disregarded (area between two centre arrows in Figure 102).

Figure 103 shows a clear drop in the relative reaction times within the first few months of implementation. After this period, the relative average reaction time stabilises below 0.2 days per reaction. Note that this stabilisation occurs in the same month that the system expanded rapidly. The lower relative reaction times during the latter stages of implementation is rather due to a much larger number of recorded reactions, thus a larger denominator, than due to an improvement in reaction times. The drop during the first few months of implementation does indicate that the system did bring important issues to the fore and created a drive for improved maintenance.

The data for the entire organisation's maintenance may be skewed by smaller operations that does not contribute significantly to the overall bottom line of the organisation. It may therefore be useful to investigate the effects that the system has on the larger operations such as Operation 4 and Operation 9. Figure 104 and Figure 105 present the average maintenance reaction time for Operation 4 and Operation 9, together with the status counts, alert counts and reaction counts. Similar to Figure 102, Figure 106 and Figure 107 compare the monthly average reaction times with the reaction count at operational level. Similar to Figure 103, Figure 108 and Figure 109 present the average monthly reaction times relative to the number of reactions for each month at operational level.

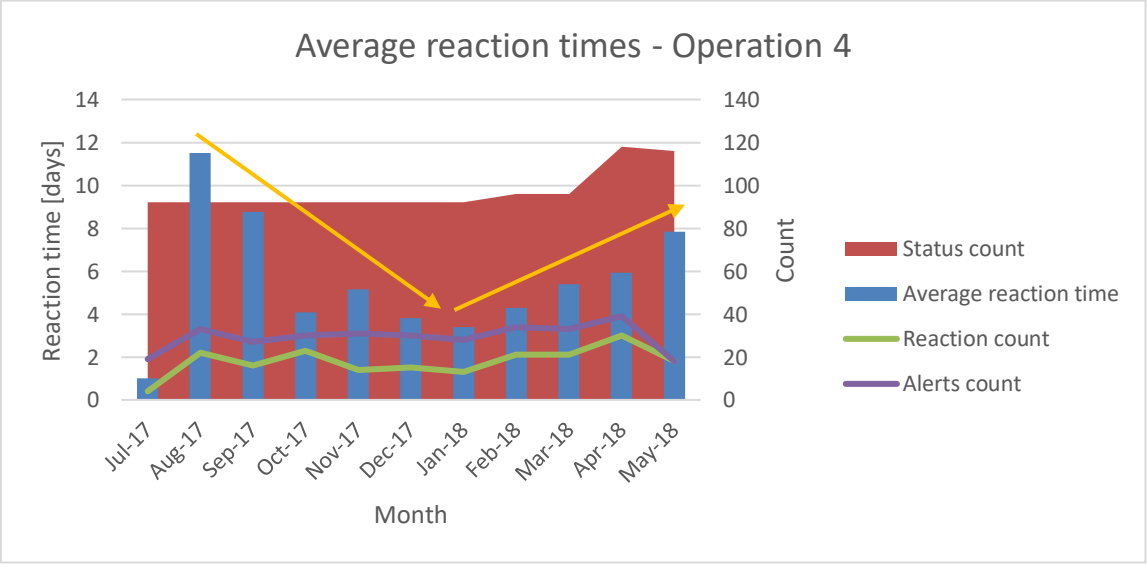


Figure 104: Average reaction times for Operation 4 over 11 months

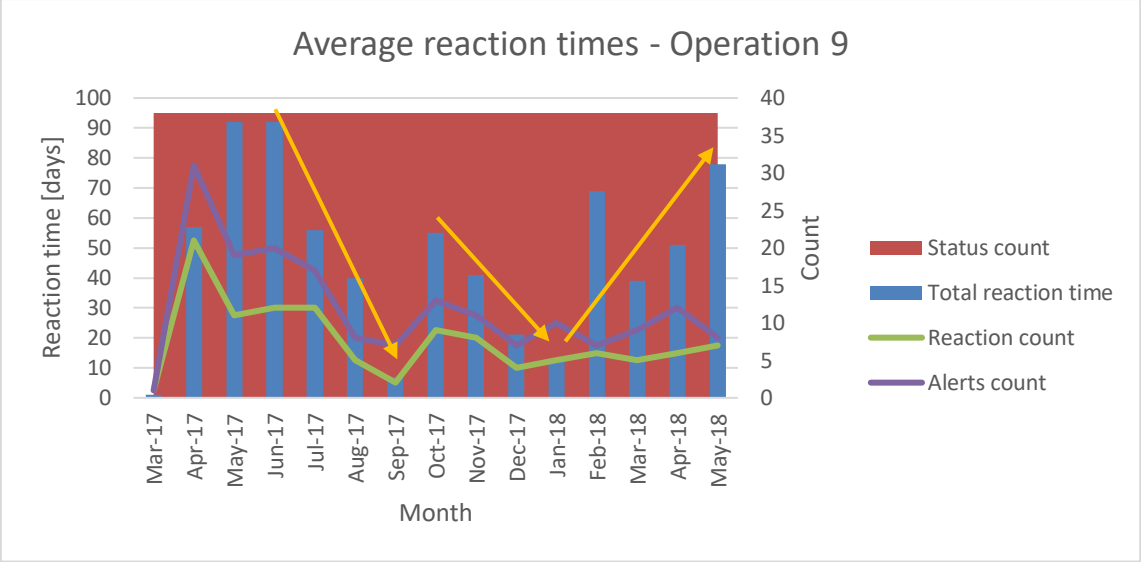


Figure 105: Average reaction times for Operation 9 over 15 months

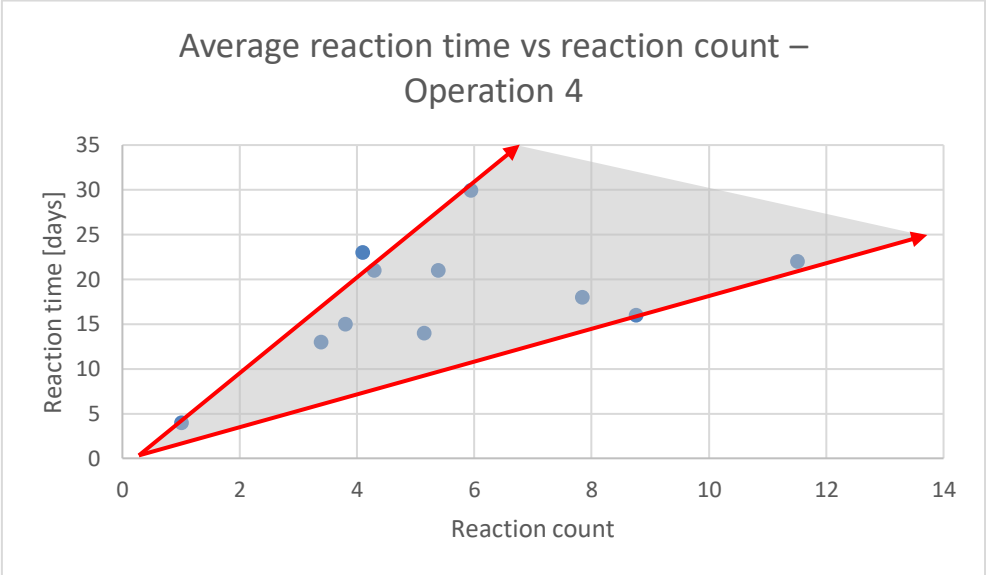


Figure 106: Average reaction times vs reaction count for Operation 4 over 11 months

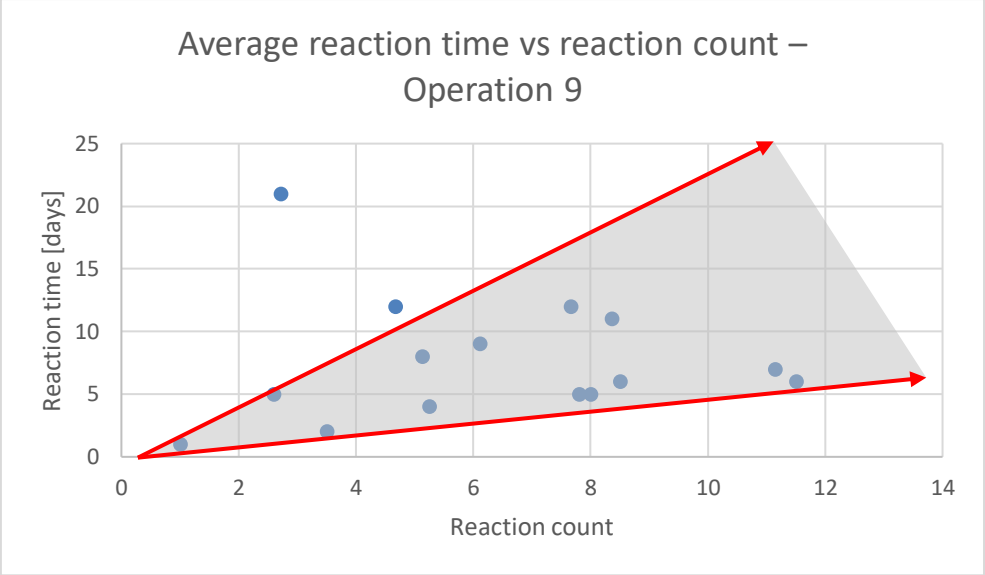


Figure 107: Average reaction times vs reaction count for Operation 9 over 15 months

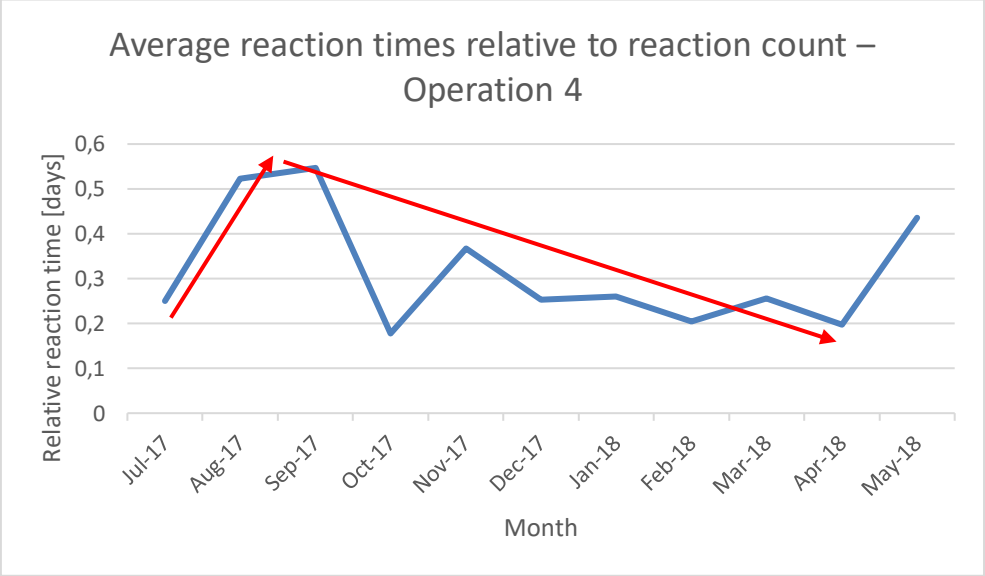


Figure 108: Average reaction times relative to the reaction count for Operation 4 over 11 months

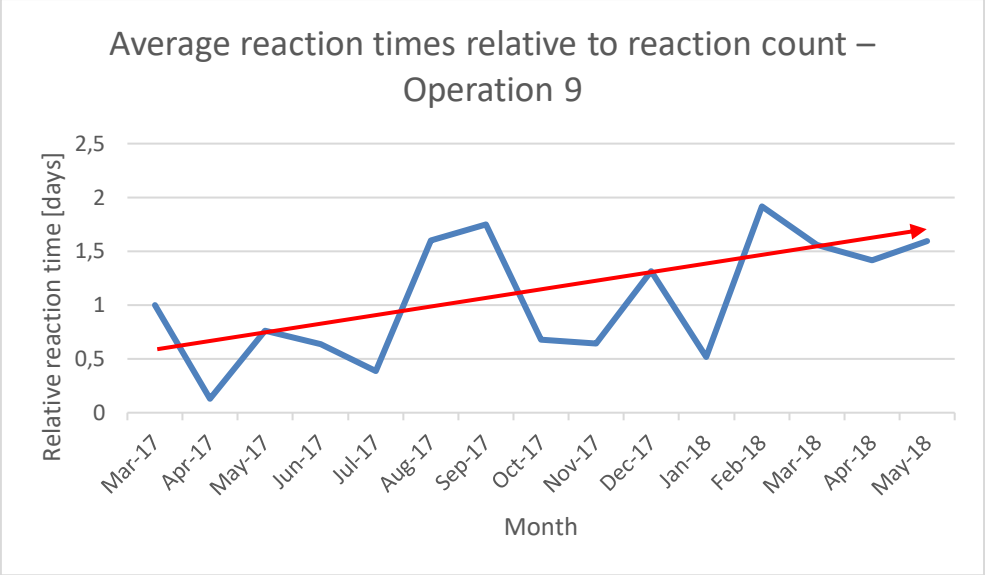


Figure 109: Average reaction times relative to the reaction count for Operation 9 over 11 months

Figure 104 indicates a clear downward trend in average maintenance reaction times at Operation 4 for 2017. Unfortunately, maintenance reactions times have since increased steadily. Figure 105 shows a similar trend in reaction times for Operation 9 than for Operation 4 in Figure 104. Operation 9's reaction times reduced steadily during 2017, except for October 2017, which was followed by an upward trend during 2018.

The increase in reaction times for Operation 9 from September to October 2017 could possibly be due to the low alert count and low reaction times during September 2017. This could have caused a lax attitude towards maintenance at the operation during the following month. Note that this possible explanation is based on the available information. Other reasons such as a shift in production requirements can possibly be to blame. For the remainder of 2017, the reaction times have steadily reduced again.

Figure 105 indicates a significant reduction in Operation 9's alert count during 2017. This is an indication that the system alerted maintenance managers and decision makers of the multitude of components operating under faulty mechanical conditions. Maintenance could then be strategically applied to reduce the number of faulty components operating on one of the organisation's critical operations.

Figure 106 and Figure 107 support the expected correlation between the reaction count and the average reaction times. There is still some spread present with the occasional outlier, but both Figure 106 and Figure 107 indicate a general linear correlation between an increase in reaction count and the average reaction times (areas between arrows). The reasons for the spread in data at operational level are similar to the reasons for the spread in data at

organisational level in Figure 102. Maintenance philosophy, parts availability, and system size may vary between systems. The data is however less spread out than at organisational level because some of the factors influencing average reaction times are removed or kept constant during analysis of the individual operation's reaction times. Variations between operations' maintenance philosophies and preference to maintenance resources are no longer applicable. The personnel in charge of maintenance management are typically the same across the operation. Some of the human variable factors are thus also kept constant.

Figure 108 indicates that the average relative reaction times in Operation 4 have steadily decreased since implementation of the system. The spike during May 2018 was largely due to the month's reaction count being anomalously low. This creates a small denominator when calculating the relative reaction time and causes a high relative reaction time. Figure 109 indicates unstable average relative reaction times in Operation 9 with an upward trend. This may look suboptimal, but upon inspection of Figure 105, the upward trend can again be attributed to the alert and reaction counts that are steadily decreasing. This indicates that Operation 9 prioritises reducing the overall number of faulty equipment over reducing reaction times.

The different operations in a large industrial organisation are typically managed by different staff. Even though an organisation may have a defined maintenance philosophy and strategy, these may vary from operation to operation. The trends in the average reaction times across the various operations may give insight into these differences. Figure 110 presents the average reaction times of the various operations and compares them with the overall average reaction times of the organisation.

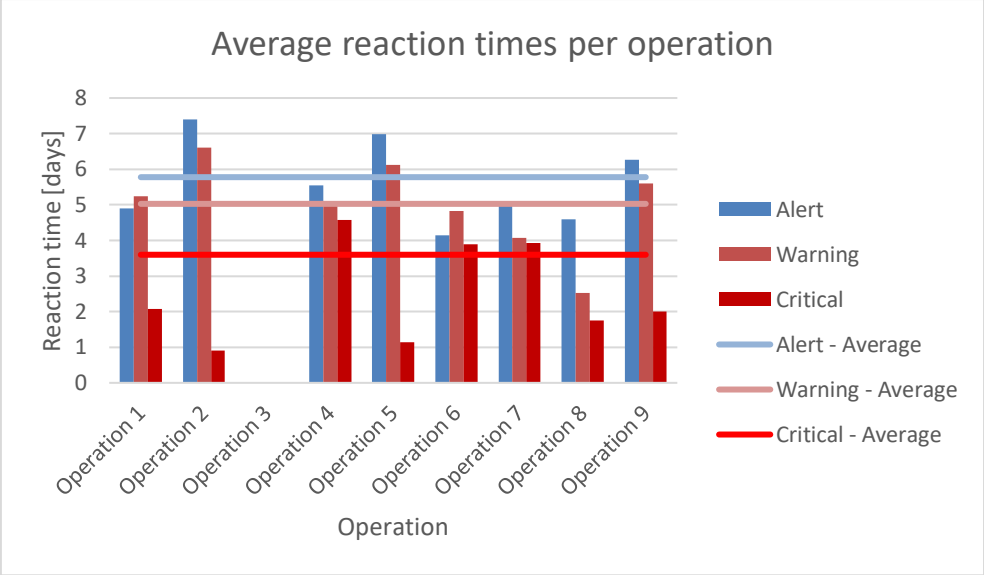


Figure 110: Average reaction times of the various operations

Figure 110 gives insight into how the maintenance philosophies of various operations can influence the organisation's average reaction times. Operation 2 and Operation 5 increase the average reaction times of the organisation. These operations however tend to focus on critical status parameters as their reaction times to critical parameters are very low. Operation 1 and Operation 8 also follow this approach of reacting quickly to critical parameters.

5.4.2.3 Interpretation

At organisational level, no clear trend in average reaction times is observable. Initially the reaction times were erratic from month to month. After approximately four months, the reaction times stabilised without increasing or decreasing. Further investigation indicated some linear correlation between the average reaction times and the number of reactions if outliers were discarded. When the relative reaction times are investigated, a clear downward trend can be observed across the entire organisation.

In Operation 4, the average reaction times gradually decreased during 2017. Unfortunately, these reaction times started to increase in 2018. The linear correlation between the reaction times and the number of reactions is stronger at this operational level than at the organisational level. The relative reaction times showed a clear downward trend for the most part during the implementation period – even 2018 when the reaction times increased.

In Operation 9, the average reaction times decreased for most of 2017. Unfortunately, as with Operation 4, these reaction times started to increase in 2018. The linear correlation between the reaction times and the number of reactions is also stronger at this operational level than at the organisational level. The relative reaction times did not decrease, but instead showed a

steady increase during the implementation period. Further investigation indicated that this was due to a steady reduction in the number of faulty equipment. This operation therefore places greater emphasis on reducing the number of faulty equipment rather than reducing the reaction times.

Overall, the condition monitoring system reduced the maintenance reaction times during the implementation period. In the cases where reaction times were not reduced, the overall number of faulty equipment was reduced, which is also beneficial. The system can therefore be used to improve maintenance in other ways than reducing the reaction times. The second objective of the study has therefore been achieved.

5.4.3 Configuration time

5.4.3.1 Objective

This subsection aims to validate the following study objective, stated in Section 1.6:

"Minimise the time needed to implement condition monitoring in an organisation."

5.4.3.2 Data

The data on the configuration time of existing condition monitoring systems is limited at best. The configuration time of the developed system can therefore not be compared with the configuration time of existing condition monitoring systems. The configuration time of the developed system is therefore compared with the configuration times of the various phases of the developed system. The phases of the developed system in chronological order are the prototype system, a beta system using a single tree structure, and the final system using two tree structures and node links. The configuration time is also compared with the overall implementation time. This validates that the configuration of the system is not a bottleneck in the implementation process and also not resource-intensive.

Configuration of the system consists of four phases: configuration of the organisational structures, configuration of the status block formatting, configuration of the graphs, and configuration of the data displayed on the graph. The configuration process was described in detail in Chapter 4, Section 4.6.6. Raw data of the configuration times for the various phases of development is available in Appendix A. The raw configuration data have been averaged and extrapolated to give an average configuration time per operation. Figure 111 breaks the average configuration time of an operation down for each of the development phases.

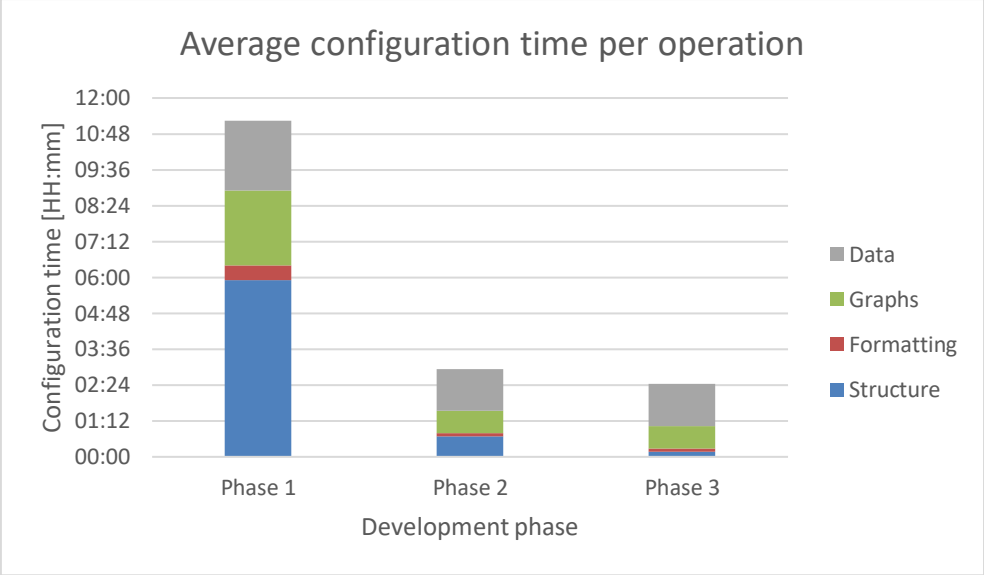


Figure 111: Average time to configure an average operation

5.4.3.3 Interpretation

Figure 111 indicates a significant reduction in the configuration time from Development Phase 1 to Development Phases 2 and 3 in all regards. Development Phase 2 was focused on developing a configurable framework. The fact that a significant reduction in configuration time is visible is a testament to the success of the configurability of the system.

Development Phase 3 was specifically focused on streamlining the configuration of the structure. A significant reduction in configuration time of the structure is visible from Development Phase 2 to 3. This indicates that Development Phase 3 succeeded in optimising the configuration of the system structure.

There is no reduction in configuration time for formatting, graphs and data from Development Phase 2 to 3. The configuration time for formatting is not a concern as it forms an insignificant part of the overall configuration time. The configuration time of graphs and data is hard to reduce since the graph and series attributes are customisable to a large extent. Additional reduction in configuration times would therefore result in a reduction in customisability of the graphs and series.

Development Phases 2 and 3 serve an additional purpose to simply reducing the configuration time of the system. Development Phase 1 had the major drawback of being configured in code. This means that developers were the only personnel able to configure the system. The code base for Development Phase 1 also became extremely large, which in turn made code maintainability near impossible. Development Phases 2 and 3 solved this problem by utilising database functionality and configuration tools. This enabled a wider range of personnel to

configure the system. This also enabled the system to expand significantly without any additions to code, resulting in a smaller code base and improved code maintainability.

5.5 Summary

The developed condition monitoring system was implemented by an ESCo in a South African gold mining organisation. The system was implemented on nine operations and four major systems (compressed air, dewatering, refrigeration and ventilation). This totalled to 28 operational systems, 201 components and 1690 measurements.

The first main study objective was to reduce the cognitive load on management level staff. The results validated that this objective has been achieved. The cognitive load has been reduced significantly at all levels of navigation. However, the cognitive load is still a bit higher than ideal at the organisational overview level though.

The second main objective of the study was to reduce maintenance reaction times. The results indicated that this objective has been achieved for the most part. The second objective is therefore considered achieved. In the cases where reaction times were not reduced, the overall number of faulty equipment was reduced, which is also beneficial.

The results also indicate a relationship between the average number of reaction times and the number of reactions in a month. The more reactions and, thus, the more maintenance is required, the longer it takes to apply maintenance.

The final main study objective was to minimise the configuration time. The system configuration time has been reduced significantly since the first iterations of the system. The structure and formatting configuration times for an operation have been reduced to a point where they are almost negligible. The graph and data configuration times currently take up most of the configuration time. This is largely due to the sheer number of measurements in the condition monitoring system and thus the number of data sets that require configuration.

6 Conclusions and recommendations

6.1 Conclusions of the conducted work

Proactive maintenance reduces maintenance costs by maintaining equipment before failure occurs thereby avoiding expensive repair costs. Despite the advantages of proactive maintenance, many industrial organisations still rely on reactive maintenance strategies. Reasons for this include pressure to maintain production rates and the ambiguity of the implications of delayed maintenance.

Condition monitoring is a useful tool to assist with proactive maintenance by providing stakeholders with information on the status of equipment health. Basic condition monitoring systems detect faults and report on component mechanical health. Advanced condition monitoring systems can identify root causes of faults and predict remaining useful equipment life.

Condition monitoring consists of three phases, namely, data acquisition, data processing, and information transfer and visualisation. Data acquisition involves measuring a physical phenomenon, such as vibration or temperature, converting these measurements to digital values, and storing the digital values. Data processing involves fault detection, diagnosis and prognosis. The information transfer and visualisation phase refers to the way a condition monitoring system informs stakeholders of the equipment health. This can be in the form of SMSs, emails, reports or a user interface.

Condition monitoring literature places significant emphasis on the data processing aspect of condition monitoring. Literature on condition monitoring information transfer and visualisation is very limited. Investigating the information transfer and visualisation aspect of existing condition monitoring systems indicated that most condition monitoring systems emphasise raw detail views without presenting operational or organisational overviews. NRG System's TurbinePhD™ is an exception with fleet view functionality that gives an overview of the health status of all the wind farms in an organisation. This system is however specialised in the wind turbine industry and cannot be used as a generic system for industrial condition monitoring.

An operational or organisational overview of the condition of equipment is a necessity. Management level decision makers operate across numerous operations and are responsible for maintenance resource allocation. Their job description comes with an inherently high level of cognitive load. These stakeholders cannot realistically perform manual investigation on all the equipment in all the operations to make informed decisions.

There is therefore a need for need for a condition monitoring system that can be used in numerous industrial industries and that has operational and organisational overview functionality. For the condition monitoring system to improve proactive maintenance, the information needs to be transferred to relevant stakeholders as effectively as possible. Thus, the system should reduce cognitive load at management level without sacrificing the data resolution and detail. Data and information should be remotely accessible, and the system should be scalable and adaptable to the organisation's needs.

Cognitive load theory suggests six methods, applicable to a user interface design, to reduce cognitive load. These methods are segmenting (present information in succession), pretraining (train users before they use the system), weeding (weed out non-essential information), signalling (emphasise essential information), aligning words and pictures, and individualisation (customised system to the user's ability).

Combining cognitive load theory with user interface design principles lays a blueprint for managing a user's focus and attention. User interfaces should be harmonious in terms of element patterns, colour, colour tone, size and alignment. Introducing anomalies into any of these attributes will draw a user's attention and focus. User interface design should also emphasise familiarity to create an intuitive feel. This is essential for creating a satisfying user experience and encouraging system adoption.

A proven method to achieve a configurable, scalable and maintainable software system is through modular framework development. Modular frameworks consist of small independent building blocks of code (modules) that integrate a larger base of software with customisable base functionality (framework). Modular software frameworks minimise technical debt build-up while enabling rapid development.

A web-based condition monitoring system was developed for information transfer and visualisation, with special emphasis on cognitive load reduction and overview functionality. The system serves as a plug-and-play system for data acquisition and data processing. The system is adaptable to numerous manual and automated data acquisition and data processing steps.

The condition monitoring system integrates with existing EMSs as an available tool in the EMSs. This allows the condition monitoring system to leverage off existing user authentication and access control, tree structure functionality and the existing relational databases.

The condition monitoring system expands standard tree structure functionality by combining the nodes of two separate tree structures to form node links. Node links significantly improve

the scalability of the system over the use of a single tree structure. A node link typically represents an operational system at the organisational level or a component parameter at the operational system level.

Node links are configurable in terms of the formatting and the data to display. Formatting consists of status data, display values and customised tooltip text. Line graphs present raw data for node links. A graph's basic attributes, such as title and axis headings, are configurable. Graphs present data in the form of series. The descriptions, colour, data and limits of these series are configurable.

A configuration tool was developed to configure the front-end condition monitoring system. This tool integrates the DBMS of the EMSs, which typically have existing node builder tools that simplify the configuration of tree structures. The condition monitoring configuration tool combines the nodes from two separate tree structures to create and configure node links.

The system was implemented by an ESCo with existing infrastructure on a gold mining organisation in South Africa. At the time of implementation, the mining organisation was running more than 30 operations with no formal condition monitoring systems other than basic control and SCADA systems. The system was implemented on nine of these operations and four major systems (compressed air, dewatering, refrigeration and ventilation) for 15 months. This totals 28 operational systems, 201 components and 1690 measurements.

The condition monitoring system proved to reduce the cognitive load at organisational level significantly. The number of elements requiring the user's focus and attention at this level is, however, still slightly above optimal at between 5.5 and 15.4 depending on the priority. Ideally, this should be below five in total. Note that this is still a great reduction from the 528 parameter groups that the organisational overview level represents. The cognitive load reduction at operational system overview and parameter detail levels is completely satisfactory.

The developed system proved to reduce maintenance reaction times for the most part. The relative reaction times at Operation 4 as well as at organisational level indicated a clear downward trend. However, at Operation 9, the maintenance reaction times increased steadily. Upon further investigation it was identified that this operation's number of faulty pieces of equipment steadily decreased since initial implementation, which is beneficial. The reduction in faulty equipment caused the increase in relative reaction times due to a decreasing denominator.

The analysis of maintenance reaction times further revealed a correlation between the number of reactions and the average reaction times. As expected, the more maintenance is required (higher reaction count), the longer it takes to react and apply maintenance to equipment.

The system configuration times have been minimised to a large extent. Initial iterations of the system required development skills to configure the system, which created technical debt. A second iteration improved the configurability and maintainability of the system so that project engineers could set up an operation. This iteration also reduced the configuration time significantly. The latest iteration of the system further improved the configuration time by reducing the number of duplicate nodes.

The condition monitoring system achieved the three main study objectives. The developed system was able to improve proactive maintenance by reducing the cognitive load placed on management level personnel and improving information accessibility. Furthermore, the system configuration time minimised. The study is therefore considered a success.

6.2 Recommendations for future work

6.2.1 Cognitive load

Cognitive load at organisational level is still a bit high and there is room for improvement. A simple method that can improve this is by using gradient colours rather than flat colours. The intensity of the colour can indicate how critical a status block is. An operational system status block with 50% of its components critical can be completely red, while an operational system with only 5% of its components critical can be green or orange with a hint of red. This way it will be easy to identify the most critical operational systems and prioritise them over less critical operational systems.

Another way a user's attention and focus can be drawn to an element is through movement. Introducing slight movements to elements, such as pulsing, can be a useful addition to the front-end system's user interface. Note that movements such as these may only be compatible with certain web browsers.

6.2.2 Functionality

Many condition monitoring systems have component layout diagrams. Component diagrams are complex to set up and configure due to their variable nature. The diagrams were therefore not included in the developed system. It may however be useful to investigate ways of incorporating such functionality into the system. It would typically be presented at parameter detail levels in parallel with graphs.

This idea of viewing the component's real-time data in image form can be extended further. With thermography, the camera creates a thermal map of the component at which it is pointed. If the system allows the live streaming of video content, this thermal map can be displayed in real time on the condition monitoring system.

6.2.3 Configuration

The configuration times of the graphs and, more specifically, the series are still a bit slow. The latest iteration of the system only focused on improving the configuration times of the structure and not the information linked to a node link. Improving this configuration time is difficult due to the sheer number of measurements per operation. There is however still some room for optimisation in the configuration of text attributes of graphs and axis titles. These text fields can have default values based on the selected node links. This will reduce the amount of typing needed during configuration of graphs and series.

Currently, the configuration tool differs significantly in layout and feel from the front-end system. Developing a configuration tool that looks like the front-end system can improve the user experience for the engineers that configure the system. Instead of having both node trees on the left-hand side of the user interface, the data structure node tree can be moved to the top of the screen, similar to the front-end system. Node link blocks can then be predrawn and selected to create, edit and delete node link information. This will emphasise the feeling of "building up" the front-end system as you configure the various node links.

6.2.4 Implementation

The fact that reaction times have increased in 2018 for two major operations is a problem. This could be due to system negligence or simply a lack of resources. Either way, management level personnel need to be aware of the overall performance trends of the maintenance strategy. Work on a reporting system that tracks the long-term trends in equipment conditions is already underway. Dr Stephan van Jaarsveld will be presenting the system at the 29th Annual Conference of the South African Institute for Industrial Engineering in October 2018.

The system was only implemented in a single organisation in the mining industry. The system verification indicated that the system is generic and adaptable enough to be compatible with other industries. These industries can potentially extend beyond industrial environments. An example of such an application is in the commercial vehicle industry where the conditions of a fleet of transport vehicles can be monitored and visualised.

6.2.5 Value

The study investigated the effects of the developed system on the reaction times to apply maintenance. The study concluded that the reaction times were reduced for the most part, but the study did not quantify the monetary value of this reduction. The fact that maintenance is now done in a more timeous fashion will reduce the costs by avoiding expensive equipment repair or replacement costs. The may however increase slightly due to the increased maintenance frequency. The monetary value may further be influenced by the equipment operating more efficiently and according to design specifications.

The reduction in cognitive load at management level assists managers in making maintenance decisions. With this system, these personnel do not have to spend as much time investigating and identifying where maintenance is required. The value of the freed-up time in these personnel's schedule is currently unknown and may need to be investigated.

REFERENCE LIST

[1] J. Lee, “Machine performance monitoring and proactive maintenance in computer-integrated manufacturing: Review and perspective,” *Int. J. Comput. Integr. Manuf.*, vol. 8, no. 5, pp. 370–380, 1995.

[2] J. Jonker, “Automated mine compressed air control for sustainable savings,” M.Eng. dissertation, North-West University, Potchefstroom, 2016.

[3] J. P. de Jager, “Investigating the effect of pump availability on load shift performance,” M.Eng. dissertation, North-West University, 2015.

[4] H. J. Groenewald, “A performance-centered maintenance strategy for industrial DSM projects,” Ph.D. thesis, North-West University, Potchefstroom, 2015.

[5] H. L. Grobbelaar, “Maintenance procedures on DSM pumping projects to improve sustainability,” M.Eng. dissertation, North-West University, 2014.

[6] T. Smith, “Automated control of mine dewatering pumps,” M.Eng. dissertation, North-West University, Potchefstroom, 2014.

[7] M. S. Ballal, G. C. Jaiswal, D. R. Tutkane, P. A. Venikar, M. K. Mishra, and H. M. Suryawanshi, “Online condition monitoring system for substation and service transformers,” *IET Electr. Power Appl.*, vol. 11, no. 7, pp. 1187–1195, 2017.

[8] J. Yan, M. Koç, and J. Lee, “A prognostic algorithm for machine performance assessment and its application,” *Prod. Plan. Control*, vol. 15, no. 8, pp. 796–801, 2004.

[9] M. Jasiulewicz-Kaczmarek, “SWOT analysis for planned maintenance strategy: A case study,” *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 674–679, 2016.

[10] R. L. M. Kotze and J. K. Visser, “An analysis of maintenance performance systems in the South African mining industry,” *S. Afr. J. Ind. Eng.*, vol. 23, no. 3, pp. 13–29, 2012.

[11] C. J. Visagie, “Successful outsourcing of maintenance in the mining industry: Methods and principles,” M.Eng. dissertation, University of Johannesburg, Johannesburg, 2005.

[12] J. Adamczyk and A. Lutynski, 'Role of engineering diagnostics and testing in maintenance of surface mining machines', in *Condition Monitoring of Open Cast Mining Machinery*. Wroclaw, Poland: Walter Bartelmus, 2006, p. 17-18.

[13] N. K. Myshkin and L. V. Markova, *On-line Condition Monitoring in Industrial Lubrication and Tribology*, vol. 8. Minsk, Belarus: Springer International Publishing, 2016, p. ii, xvii, xix.

[14] J. A. Stols, H. G. Brand, and W. Schoeman, “A condition monitoring system using basic equipment protection measurements,” in *Proceedings of the 28th Annual Conference of the Southern African Institute for Industrial Engineering, SAIIE*, 2017.

[15] S. Budzan, D. Buchczik, M. Pawełczyk, and R. Wyżgolik, “Rotating machinery diagnostics based on fusion of infrared and vibration measurements,” in *Advances in Condition Monitoring of Machinery in Non-Stationary Operations*, Cham, Switzerland: Springer International Publishing AG, 2018, pp. 203–212.

[16] M. C. van der Bank, “A modular framework for the effective development of web-based applications,” M.Eng. dissertation, North-West University, 2017.

[17] R. C. Atkinson and R. M. Shiffrin, “Human memory: A proposed system and its control processes,” *Psychol. Learn. Motiv.*, vol. 2, pp. 89–195, 1968.

[18] S. van Jaarsveld, “Developing an integrated information system to assess the operational condition of deep level mine equipment,” Ph.D. thesis, North-West University, Potchefstroom, 2017.

[19] K. Dziedziech, A. Jabłoński, and T. Barszcz, “Optimization of calculations for wireless condition monitoring systems,” in *Advances in Condition Monitoring of Machinery in Non-Stationary Operations*, Cham, Switzerland: Springer International Publishing AG, 2018, pp. 123–132.

[20] P. Różak, J. Zieliński, P. Czop, A. Jabłoński, T. Barszcz, and M. Mareczek, “Supervised classification methods in condition monitoring of rolling element bearings,” in *Advances in Condition Monitoring of Machinery in Non-Stationary Operations*, Cham, Switzerland: Springer International Publishing AG, 2018, pp. 133–146.

[21] C. U. Mba, S. Marchesiello, A. Fasana, and L. Garibaldi, “Fault detection in gears using stochastic resonance,” in *Advances in Condition Monitoring of Machinery in Non-Stationary Operations*, Cham, Switzerland: Springer International Publishing AG, 2018, pp. 55–70.

[22] P. Potočnik and E. Govekar, “Semi-supervised vibration-based classification and condition monitoring of compressors,” *Mech. Syst. Signal Process.*, vol. 93, pp. 51–65, 2017.

[23] N. Tondon and A. Choudhury, “A review of vibration and acoustics measurement methods for the detection of defects in rolling element bearing,” *Tribol. Int.*, vol. 32, no. 8, pp. 469–480, 1999.

[24] J. Wodecki, P. Stefaniak, M. Polak, and R. Zimroz, “Unsupervised anomaly detection for conveyor temperature SCADA data,” in *Advances in Condition Monitoring of Machinery in Non-Stationary Operations*, Cham, Switzerland: Springer International Publishing AG, 2018, pp. 361–369.

[25] M. Witos, M. Zieja, A. Szczepankowski, and J. Szymczak, “Structural health monitoring of aero-engines in non-stationary operations,” in *Advances in Condition Monitoring of Machinery in Non-Stationary Operations*, Cham, Switzerland: Springer International Publishing AG, 2018, pp. 1–11.

[26] A. Fourati *et al.*, “Current signal analysis of an induction machine with a defective rolling bearing,” in *Advances in Condition Monitoring of Machinery in Non-Stationary Operations*, Cham, Switzerland: Springer International Publishing AG, 2018, pp. 45–54.

[27] D. Buchczik and M. Pawełczyk, “Calibration of accelerometers using multisinusoidal excitation,” in *Advances in Condition Monitoring of Machinery in Non-Stationary Operations*, Cham, Switzerland: Springer International Publishing AG, 2018, pp. 213–222.

[28] J. N. du Plessis, “Development of a supervisory system for maintaining the performance of remote energy management systems,” Ph.D. thesis, North-West University, Potchefstroom, 2014.

[29] M. Kedadouche, Z. Liu, and M. Thomas, “Bearing fault feature extraction using autoregressive coefficients, linear discriminant analysis and support vector machine under variable operating conditions,” in *Advances in Condition Monitoring of Machinery in Non-Stationary Operations*, Cham, Switzerland: Springer International Publishing AG, 2018, pp. 339–351.

[30] M. Nie and L. Wang, “Review of condition monitoring and fault diagnosis technologies for wind turbine gearbox,” *Proc. CIRP*, vol. 11, no. 2013, pp. 287–290, 2013.

[31] J. Son, S. Zhou, C. Sankavaram, X. Du, and Y. Zhang, “Remaining useful life prediction based on noisy condition monitoring signals using constrained Kalman filter,” *Reliab. Eng. Syst. Saf.*, vol. 152, pp. 38–50, 2016.

[32] J. N. du Plessis, R. Pelzer, and J. C. Vosloo, “Sustaining the performance of diverse energy management systems through reactive maintenance,” in *Proceedings of the 12th Conference on the Industrial and Commercial Use of Energy, ICUE*, 2015.

[33] J. Tautz-Weinert and S. J. Watson, “Using SCADA data for wind turbine condition monitoring: A review,” *IET Renew. Power Gener.*, vol. 11, no. 4, pp. 382–394, 2017.

[34] W. J. van Blerk, “Bicycle ride evaluation and optimisation,” B.Eng. dissertation, University of Pretoria, Pretoria, 2016.

[35] F. Paas, J. Tuovinen, H. Tabbers, and P. W. M. van Gerven, “Cognitive load measurement as a means to advance cognitive load theory,” *Educ. Psychol.*, vol. 38, no. 1, pp. 63–71, 2010.

[36] F. G. W. C. Paas and J. J. G. van Merriënboer, “Instructional control of cognitive load in the training of complex cognitive tasks,” *Educ. Psychol. Rev.*, vol. 6, no. 4, pp. 351–371, 1994.

[37] B. Xie and G. Salvendy, “Prediction of mental workload in single and multiple task environments,” *Int. J. Cogn. Ergon.*, vol. 4, no. 3, pp. 213–242, 2000.

[38] P. M. Linton, B. D. Plamondon, A. O. Dick, A. C. B. Jr., and R. E. Christ, “Operator workload for military system acquisition,” in *Applications of Human Performance Models to System Design*, Boston, MA: Springer, 1989, pp. 21–46.

[39] R. E. Mayer and R. Moreno, “Nine ways to reduce cognitive load in multimedia learning,” *Educ. Psychol.*, vol. 38, no. 1, pp. 43–52, 2003.

[40] P. D. Mautone and R. E. Mayer, “Signaling as a cognitive guide in multimedia learning,” *J. Educ. Psychol.*, vol. 93, no. 2, pp. 377–389, 2001.

[41] R. Moreno and R. E. Mayer, “Cognitive principles of multimedia learning: The role of modality and contiguity,” *J. Educ. Psychol.*, vol. 91, no. 2, pp. 358–368, 1999.

[42] R. E. Mayer and V. K. Sims, “For whom is a picture worth a thousand words? Extensions of a dual-coding theory of multimedia learning,” *J. Educ. Psychol.*, vol. 86, no. 3, pp. 389–401, 1994.

[43] A. Blair-Early and M. Zender, “User interface design principles for interaction design,” *Des. Issues*, vol. 24, no. 3, pp. 85–107, 2008.

[44] M. W. M. Jaspers, T. Steen, C. van den Bos, and M. Geenen, “The think aloud method: A guide to user interface design,” *Int. J. Med. Inform.*, vol. 73, no. 11–12, pp. 781–795, 2004.

[45] J. Raskins, “Intuitive equals familiar,” *Commun. ACM*, vol. 37, no. 9, pp. 17–19, September, 1994.

[46] C. Courage and K. Baxter, *Understanding Your Users: A Practical Guide to User Requirements*. San Francisco, CA: Morgan Kaufmann, 2005.

[47] A. V. Hausman and J. S. Siekpe, “The effect of web interface features on consumer online purchase intentions,” *J. Bus. Res.*, vol. 62, no. 1, pp. 5–13, 2009.

[48] M. Csikszentmihalyi and P. H. Mirvis, “Flow: The psychology of optimal experience,” *Acad. Manag. Rev.*, vol. 16, no. 3, pp. 636–640, 1991.

[49] D. N. Smith and K. Sivakumar, “Flow and Internet shopping behavior: A conceptual model and research propositions,” *J. Bus. Res.*, vol. 57, no. 10, pp. 1199–1208, 2004.

[50] K. S. Song, A. M. Fiore, and J. Park, “Telepresence and fantasy in online apparel shopping experience,” *J. Fash. Mark. Manag.*, vol. 11, no. 4, pp. 553–570, 2007.

[51] W. Swasty and A. R. Adriyanto, “Does color matter on web user interface,” *CommIT*, vol. 11, no. 1, pp. 17–24, 2017.

[52] W. Cunningham, “The WyCash portfolio management system,” in *OOPSLA ’92 Addendum to the Proceedings on Object-oriented Programming Systems, Languages, and Applications (Addendum)*, 1993, vol. 4, no. 2, pp. 29–30.

[53] E. Parodi, S. Matalonga, D. Macchi, and M. Solari, “Comparing technical debt in student exercises using test driven development, test last and ad hoc programming,” in *Proceedings of the 2016 42nd Latin American Computing Conference, CLEI 2016*, 2017.

[54] D. Calisi, A. Censi, L. Iocchi, and D. Nardi, “OpenRDK: A modular framework for robotic software development,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2008.

[55] C. Y. Baldwin and K. B. Clark, “Design rules, Volume 1: The power of modularity,” *Acad. Manag. Rev.*, vol. 26, no. 1, pp. 130–133, 2001.

[56] Y. Cai and K. Sullivan, “A formal model for automated software modularity and evolvability analysis,” *ACM Trans. Softw. Eng. Methodol.*, vol. 21, no. 4, 2012.

APPENDIX A

1. Overview

This appendix presents the raw configuration time data of the various development phases.

2. Phase 1

This section presents the logged configuration times during Development Phase 1 in Table 9 to Table 12.

Table 9: Raw data of minutes spent configuring grid blocks during Development Phase 1

Grid blocks			
Date	**Grid blocks**	**Duration [min]**	**Duration per block [min]**
2017-01-16	19	165	8.7
2017-03-14	1	150	13.1
2017-03-16	24	150	6.3
2017-03-20	18	165	9.2
2017-03-22	10	165	8.0
2017-04-04	1	15	15.0

Table 10: Raw data of minutes spent configuring format tags during Development Phase 1

Format tags			
Date	**Tags**	**Duration [min]**	**Duration per tag [min]**
2017-01-20	7	60	8.6
2017-01-24 to 2017-01-25	4	1	0.2
2017-01-30	1	1	0.5
2017-02-13	8	45	5.6
2017-02-20	14	45	3.2
2017-03-02	6	45	7.5
2017-03-14	3	7	2.1
2017-03-16 to 2017-03-17	27	13	0.4

Table 11: Raw data of minutes spent configuring graphs during Development Phase 2

Graphs			
Date	**Graphs**	**Duration [min]**	**Duration per graph [min]**
2017-01-16	8	135	16.9
2017-01-16	16	120	7.5
2017-01-17	36	105	2.9
2017-01-17 to 2017-01-18	12	90	7.5
2017-01-24 to 2017-01-25	20	39	1.9
2017-01-30	9	59	6.5
2017-03-02	10	26	2.5
2017-03-14	5	39	7.6
2017-03-16	32	90	2.8
2017-03-16 to 2017-03-17	18	45	2.5
2017-03-20	18	30	1.7

Table 12: Raw data of minutes spent configuring data tags during Development Phase 3

Data tags			
Date	**Tags**	**Duration [min]**	**Duration per tag [min]**
2017-01-18	18	90	5.0
2017-01-23	16	90	5.6
2017-01-24 to 2017-01-25	35	66	1.9
2017-01-30	4	90	22.5
2017-01-30	14	93	6.6
2017-03-02	18	50	2.8
2017-03-14	9	56	6.2
2017-03-16 to 2017-03-17	135	303	2.2
2017-03-22	18	15	0.8
2017-03-22	17	86	5.0
2017-04-04	10	60	6.0

3. Phase 2

This section presents the configuration times during Development Phase 2 in Table 13.

Table 13: Configuration times for a partial operation during Development Phase 2

Configuration duration [sec]					
	Structure	**Formatting***	**Graphs***	**Data***	**Total**
Operation	27	N/A	N/A	N/A	27
Pumps	32	N/A	N/A	N/A	32
Pump 1	45	26	61	82	214
Pump 2	44	25	57	78	204
Pump 3	40	23	53	76	192
Pump 4	41	24	54	74	193
Fans	26	N/A	N/A	N/A	26
Fan 1	44	25	58	79	206
Fan 2	43	22	52	73	190
Fan 3	40	21	50	74	185
Fan 4	36	21	50	71	178

*Note: Fields marked with (*) are identical between Development Phase 2 and 3. No changes were made to these aspects of the configuration tool; therefore, values can be shared.

4. Phase 3

This section presents the configuration times during Development Phase 3 in Table 14.

Table 14: Configuration times for a partial operation during Development Phase 3

Configuration duration [sec]					
	Structure	**Formatting***	**Graphs***	**Data***	**Total**
Operation	23	N/A	N/A	N/A	23
Pumps	21	N/A	N/A	N/A	21
Pump 1	19	26	61	82	188
Pump 2	20	25	57	78	180
Pump 3	18	23	53	76	170
Pump 4	20	24	54	74	172
Fans	22	N/A	N/A	N/A	22
Fan 1	24	25	58	79	186
Fan 2	21	22	52	73	168
Fan 3	19	21	50	74	164
Fan 4	20	21	50	71	162

*Note: Fields marked with (*) are identical between Development Phase 2 and 3. No changes were made to these aspects of the configuration tool; therefore, values can be shared.

5. Combined data

This section presents the extrapolated configuration times of the various configuration times in Table 15.

Table 15: Extrapolated confirmation times of various development phases

	Structure	Formatting	Graphs	Data	Total
Phase 1	05:55:00	00:29:12	02:29:36	02:20:50	11:14:38
Phase 2	00:41:24	00:06:03	00:44:57	01:24:13	02:56:37
Phase 3	00:10:54	00:06:03	00:44:57	01:24:13	02:26:07

*Time format: [HH:mm:ss]