Impromptu: enabling maximum insight into capital-goods machines within available analyst time by providing reusable and flexible data analysis

Citation for published version (APA):
IMPROMPTU:
Enabling maximum insight into capital-goods machines within available analyst time by providing reusable and flexible data analysis

Luc de Smet
September 2016
IMPROMPTU:
Enabling maximum insight into capital-goods machines within available analyst time by providing reusable and flexible data analysis

Eindhoven University of Technology
Stan Ackermans Institute / Software Technology

Partners
Océ – A Canon Company
Eindhoven University of Technology

Steering Group
Rob Kersemakers
Edy Klomp
Mykola Pechenizkiy

Date
September 2016

Document status
Public

The design described in this report has been carried out in accordance with the TU/e Code of Scientific Conduct.
Abstract
In the development process of high-tech machines, it is important to verify performance and quality of the machine in the field continuously. Currently, there are opportunities to increase analysis effectiveness w.r.t machine behavior analysis. Off the shelf tools are not sufficient to solve these problems. The Impromptu project shows a proof of concept to extend the current analysis framework to be reusable, flexible, and accessible. Within this project, two business cases were explored to show the accessibility and flexibility of the framework and reusability of generic data analysis techniques in the context of Océ – A Canon Company.

Keywords
Data analysis, Time series, Correlation, Anomaly detection, printing
FOREWORD

During the last years, the number of internet-connected devices has grown tremendously. The acceptance of internet-connected devices (in the form of the Internet Of Things) has enabled Océ and many other companies to collect telemetry data of devices at customer sites, allowing for analysis of device behavior. Usage statistics derived from this analysis may indicate potential future improvements, while statistical information on wear and tear enable predictive/condition based maintenance to allow unplanned downtime to be reduced. In case of newly introduced products often more parameters are logged "just in case", to allow for learning how the product behaves once deployed. Also, initially it may not be clear what parameters can be actually used to predict certain device behavior, and the analysis of these parameters may require specific specialist knowledge and manual labor. The intention of this project was to investigate the potential for an improvement in telemetry analysis automation: how to shift from manual, reactive to automated, proactive analysis.

During the project, Luc has very actively sought the cooperation with various specialists from R&D and Service to find out about their way of working, and in what ways the current data analysis processes could be improved. Taking their main challenges into account, he has picked the most promising topics: outlier detection and signals correlation. Based upon an existing analysis framework, he has built an extension framework to accelerate the development of new analysis techniques and built his own algorithm on top of that as validation. For outlier detection, he implemented a set of algorithms and validated these using specific incidents. He proved that his method can be used to detect similar issues, if minimal knowledge on expected signal behavior is applied. For signals correlation, his challenge was to apply generic correlation detection algorithms to signals with different time behavior. His investigations show that also in this case, specialist knowledge is needed to be able to detect correlation, and that black box detection is not easily done.

We appreciate how he managed to keep all stakeholders involved during his project, and consider his work as valuable input for further activities on improving data analysis processes in Océ.

Rob Kersemakers and Edy Klomp

2016-09-06
PREFACE

This report presents the Impromptu project that was carried out by me at Océ Venlo. The project was conducted as a full-time, nine-month graduation assignment in the context of a two-year technological designer program in Software Technology, known as a Professional Doctorate in Engineering (PDEng) program and also known as Ontwerpers Opleiding Technische Informatica (OOTI). This post-master's program is offered by the Eindhoven University of Technology under the auspices of the Stan Ackermans Institute.

This report is intended for my Project Steering Group (PSG), my peers at Océ and TU/e, and the Final Performance Evaluation (FPE) Committee. It is also intended for architects, software engineers, machine designers, service technicians, and data analysts with similar challenges in the field of data analysis.

This version of the report does not contain any confidential details. The full version of the report is at the disposal of Océ employees and other associates who have signed a non-disclosure agreement with Océ.

For architects, software engineers, machine designers, service technicians, and data analysts, I suggest reading the executive summary on page 7 and going into details based on preference. For my peers at Océ and TU/e, I suggest focusing on the problem description on page 20 and corresponding design on page 47. For my Final Performance Evaluation committee, I advise considering the competencies expressed in the outline and all corresponding chapters.

Luc de Smet
September 14, 2016
ACKNOWLEDGEMENTS

This project could never have been a success without the help of various people within Océ and Eindhoven University of Technology (TU/e). My supervisors, various stakeholders at Océ, and coaches at TU/e have been a great help.

First and foremost, I want to thank my supervisors. Edy Klomp has been a tremendous help throughout the project and always made time to help or review work. Rob Kersemakers has been a source of company knowhow and feedback. Mykola Pechenizkiy provided help on scientific analysis and gave feedback. From both TU/e and Océ I have had great support from other staff as well. In particular, thanks to Ad Aerts for arranging this project, Ronald Fabel for being the responsible manager at Océ, Lucas Cras for introducing me to Océ and keeping in touch, and Desiree van Oorschot for lightning-fast responses to my requests. I also want to thank the coaches from TU/e involved with my project. In particular I thank Harold Weffers for good conversation and project process insight.

Next I want to thank the others of the ORS (Remote service) team which I was part of. Thanks Georgios Metaxas, Ruurd van der Meer, Tim Paffen, Tomi Rhee, and of course Edy and Rob, for a truly “gezellig” atmosphere.

This project started very broadly and various stakeholders were considered for help. In particular I want to the Machine module designers, service personnel, and others directly involved in giving me valuable feedback on prototypes and insight into the domain: Bert van Beek, Emile van Gerwen, Frank Zeelen, Henri Hogenhorst, Joost Janse, Marcel Haenen, and Ruud Jacobs.

In addition, I want to thank all other colleagues I interviewed and gave me insight into the ways of Océ and data analysis within Océ: Arjan van der Hoogt, Bert Jan Woltinge, Bram Thijsen, Cas van Elderen, Corinna May, David Riedel, Dennis Rijbroek, Eric Staal, Eugen Schindler, Gé Kessels, Henk van den Broek, Jennek Geels, Johan Hoogendoorn, Klaas Jan Wierda, Laurens Kuipers, Lei Hendriks, Lonneke Teeuwen, Lou Somers, Mario Lange, Mark ten Have, Michel de Groot, Patrick Vestjens, Peter Kruizinga, Rakesh Partapsing, Remco Hammen, Roel Straetemans, Roelof Hamberg, Takeshi Namikata, Tim Witter, Ton Gerrits, Ton van Buren, William Lossie, and Wolfgang Rutter. I also want to thank the people on the floor who have directly or indirectly helped me throughout these nine months. Thanks to Hans Kessels, Henk Thijsen, Jan Saris, Jeroen Janssen, Jos Jans, Peter Cornelissen, Peter Teeuwen, and Willy Venbrux.

Finally, I want to thank Alina Calugar and Dan Cringus for driving me to and from Océ for (almost) the entire project and for providing me with enjoyable conversations and insight into the company.

Luc de Smet

Date 2016-09-14
EXECUTIVE SUMMARY

In the development process of high-tech machines, it is important to verify performance and quality of the machine in the field continuously. The current company strategy is to deploy machines at companies that are early adopters for early validation and verification in the field. Canon customers consider these machines to be capital goods. The goal is to continuously increase machine maturity by improving design and the improving the ability to predict defects. To effectively improve design and predict defects, one needs to understand the behavior of the machines in the field. This report concerns a project enabling to gain more insight given the available time for specialists in the R&D and service departments. I propose an extension of the current analysis tools with a reusable and flexible data analysis approach; this project presents such an approach, called Impromptu.

This approach is developed and deployed within the R&D department of Océ, A Canon Company in Venlo. On their behalf, Impromptu was co-developed with Océ initiatives, in particular ODAS (a data analysis framework based on the Jupyter technology\(^1\)). The solution focuses on machine module designers and service product specialists. Their involvement is low risk (i.e., in the scope of this project, they are accessible), their input is necessary for sound analysis (i.e., domain knowledge is critical for a valid analysis), and their potential gains are high (such insight can be applied to product design and service strategy).

Currently, there are opportunities to improve analysis tools and the data format to increase analysis effectiveness. At this moment, it can take days to gain insight into the abnormal behavior of a machine over a longer time period. Until now, supporting the tens of machines in the field of the carrier project has been no issue. Prospective growth to over a hundred machines in the field for the next year makes the current approach to analysis infeasible in the near future.

Specific bottlenecks in analysis are automation, utilizing multiple datasets at once, and predicting machine behavior. In addition, there is no common way to find analysis scripts and the reusability of these scripts is generally low. Finally, there is no effective way to share domain knowledge and analysis results strongly related to the data.

Off the shelf tools are not sufficient to solve these problems. Popular analysis tools and frameworks such as KNIME, Cortana, or Tableau do not offer means to do a domain-specific analysis or share domain knowledge. They are also not flexible enough to allow creation and extension of domain-specific analysis scripts for domain experts without computer science expertise. The R and Python technologies provide a solid set of analysis scripts and flexibility, but are not accessible enough for most users. Within this project scope, extending these technologies is seen as the most promising approach. ODAS uses Jupyter (Python) technology which offers many benefits and is flexible enough for extension and was used as starting point for development.

The Impromptu project shows a proof of concept to extend the ODAS framework to be reusable and flexible. Impromptu offers a prototype for an easy-to-use environment for analysts to re-use analysis scripts and share domain knowledge. This is achieved through a plugin-architecture with a common interface, an analysis scripts overview, and meta-data storage for domain knowledge that is linked to the original data logs.

\(^1\)Jupyter: Open source, interactive data science and scientific computing across over 40 programming languages. [http://jupyter.org/](http://jupyter.org/)
Within this project, two business cases were explored to show the accessibility and flexibility of the framework and reusability of generic data analysis techniques. A generic anomaly detection technique applied to the signal of a vacuum pump has shown the ability to automatically detect failures and other non-regular behavior. In addition, an attempt was made to perform root cause analysis with correlation techniques, but this investigation did not lead to new insight. In the correlation investigation I was limited by time, lack of preprocessing tools, and domain knowledge and hence did not continue.

The Impromptu design and corresponding prototype have been accepted by a group of machine module designers within the R&D department. These designers have been involved throughout the process and confirmed the validity of the approach. Their continued involvement indicates a positive impact. In the end, the conclusions and corresponding recommendations are:

- Developing and applying domain knowledge and data analysis knowledge in conjunction. Domain knowledge should be shared among machine module designers. I recommend keeping the threshold for sharing domain knowledge as low as possible. Data analysis expertise should be increased for machine module designers and service product specialists. Alternatively, an expert in applying data analysis could cooperate with these parties.
- Facilitating machine module designers with scientific analysis and automation scripts. I recommend focusing on developing generic scripts for comparison and automation, in addition to encapsulating scientific analysis techniques. This is expected to significantly improve the effectiveness of analysis.
- Defining a structure for functional logging from a platform-wide perspective and providing standard data retrieval functions. The largest part of time for analysis is spent in loading and preprocessing data. A platform-wide log definition prevents mismatches in inter-module analysis and saves time by preventing the creation of redundant loading and preprocessing scripts. I recommend:
  - Defining consistent, platform-wide functional log formats and structures as part of designing new products
  - Storing machine behavior logs in a format that allows adding meta-data (planned in ODAS framework)
  - Providing an abstraction layer towards analysts for querying and storing log data (planned in ODAS framework)
- Quantify the value of data analysis in the company context. At this point a significant investment in people and hardware must be made to set up further data analysis. To get the necessary support, I recommend investigating the gain in time and/or money by: 1) performing a predictive maintenance pilot, 2) defining a metric for the time and/or money saved by applying log data consistency, reusable analysis scripts, and domain knowledge sharing.
TABLE OF CONTENTS

Foreword ............................................................................................................................. 4
Preface ................................................................................................................................. 5
Acknowledgements ................................................................................................................ 6
Executive Summary .............................................................................................................. 7
List of Figures ....................................................................................................................... 12
List of Tables ......................................................................................................................... 13

1  Introduction ..................................................................................................................... 15
   1.1 Company context ........................................................................................................ 16
   1.2 Outline ....................................................................................................................... 19

2  Company value of data analysis ..................................................................................... 20
   2.1 Potential company gains .......................................................................................... 20
   2.2 Analysis for High production machines ................................................................... 21
      2.2.1 Overview of potential stakeholders .................................................................. 21
      2.2.2 Stakeholders and their tasks in printer creation process .................................... 22

3  Company analysis tools and limitations ........................................................................ 24
   3.1 Internal analysis investigation .................................................................................. 24
      3.1.1 Current situation ............................................................................................... 24
      3.1.2 Limitations in current situation ......................................................................... 26
   3.2 External analysis investigation .................................................................................. 28

4  Problem statement .......................................................................................................... 29

5  Company analysis tool improvements ........................................................................... 31
   5.1 Actors in the system ............................................................................................... 31
   5.2 Accessibility and reusability .................................................................................... 32
   5.3 Finding root causes of exceptional behavior ............................................................ 36
   5.4 Sharing domain knowledge ..................................................................................... 39
   5.5 Design criteria .......................................................................................................... 40

6  Investigation: Anomaly detection and correlation ........................................................... 43
   6.1 Anomaly detection techniques ............................................................................... 43
   6.2 Correlation techniques ............................................................................................ 44

7  Summary: Effective machine behavior analysis .............................................................. 46

8  Framework basis ............................................................................................................. 47
   8.1 Research of current frameworks .............................................................................. 47
   8.2 Focus of system ........................................................................................................ 49

9  Structure for effective analysis ....................................................................................... 51
   9.1 Plugin structure and interface .................................................................................. 51
LIST OF FIGURES

Figure 1 Product release lifecycle: regular versus early release ............................................................. 15
Figure 2 Océ Colorwave® 700: Can produce about 500 color posters per day on demand ......................... 17
Figure 3 Océ Arizona® flatbed printer: Signage for up to 200 signs of 1.25 x 2.5m per day ............................ 17
Figure 4 VarioPrint® i300: Cutsheet high production system .............................................................. 18
Figure 5 Report outline .......................................................................................................................... 19
Figure 6 Simplified company structure with respect to data analysis ....................................................... 21
Figure 7 Use cases in company structure .............................................................................................. 24
Figure 8 Overview of tools used for analysis ......................................................................................... 25
Figure 9 Typical current analysis cycle ............................................................................................... 26
Figure 10 Goal within the company structure ....................................................................................... 29
Figure 11 Improving current Diagnostic Framework-based analysis cycle ............................................. 30
Figure 12 Main users for system ........................................................................................................... 31
Figure 13 Accessibility and reusability focus for DF workflow (see Figure 11) ......................................... 32
Figure 14 Accessibility and reusability use cases .................................................................................. 33
Figure 15 Accessibility and reusability quality attributes ....................................................................... 35
Figure 16 Finding root cause focus for DF workflow (see Figure 11) .................................................... 36
Figure 17 Anomaly-detection-related use cases ..................................................................................... 37
Figure 18 Sharing domain knowledge focus for DF workflow (see Figure 11) ....................................... 39
Figure 19 Sharing domain knowledge-related use cases ....................................................................... 39
Figure 20 Anomaly detection technique groups ..................................................................................... 43
Figure 21 Overview of correlation technologies and problems in the company ........................................ 45
Figure 22 Example of Jupyter basic functionality .................................................................................. 50
Figure 23 Structure for generic analysis scripts .................................................................................... 52
Figure 24 Integration of Impromptu in company analysis framework .................................................. 53
Figure 25 Loading and executing plugins ............................................................................................... 54
Figure 26 Example of automatically generated overview ....................................................................... 57
Figure 27 Sharing domain knowledge relations .................................................................................... 59
Figure 28 input for tagging data with domain knowledge ....................................................................... 59
Figure 29 Interactive visualization for data enriched with domain knowledge ...................................... 59
Figure 30 Proposed interaction for sharing domain knowledge ............................................................. 60
Figure 31 Impromptu in the current tool environment ............................................................................. 61
Figure 32 Analysis flow for anomaly detection on vacuum pump failure ............................................... 62
Figure 33 Original signal of known failure: Time versus pressure ....................................................... 63
Figure 34 Vacuum pump sensor signal with inverted pressure value ..................................................... 64
Figure 35 Vacuum pump preprocessing flow ......................................................................................... 65
Figure 36 Output of anomalous time finder ........................................................................................... 66
Figure 37 Difference in data sample spreads ......................................................................................... 67
Figure 38 “Anomalies” that are all within the safe threshold .................................................................. 68
Figure 39 Flow for ranking machines by anomaly score ........................................................................ 68
Figure 40 Output of anomalous machine finder .................................................................................... 68
Figure 41 Analysis flow for correlation between functions .................................................................... 70
Figure 42 Example of normalization of machine module sensor signal ............................................... 70
Figure 43 Correlation matrix for climate control module ....................................................................... 71
Figure 44 List of top correlations in dataset ........................................................................................... 72
Figure 45 Highest correlation between climate and paper deformation data ........................................... 73
Figure 46 Sum of rejects binned within one hour time periods ............................................................... 77
Figure 47 Correlations between machine climate and rejects ............................................................... 77
Figure 48 Project stakeholders ............................................................................................................. 88
Figure 49 Milestone trend chart up until 2016-07-26 .............................................................................. 89
Figure 50 Weeks versus percentage of time spent on activities ............................................................. 90
LIST OF TABLES

Table 1 Success metric for usability for basic users ................................................................. 41
Table 2 Success metric for usability for regular users ............................................................... 41
Table 3 Success metric for usability for advanced users ............................................................ 41
Table 4 Success metric for reusability of generic scripts .......................................................... 42
Table 5 Comparison of selection of data analysis frameworks .................................................. 47
Table 6 Overview of features provided in Jupyter ................................................................. 50
Table 7 Overview of features provided and planned in ODAS ................................................ 50
Table 8 Overview of features planned in Impromptu ............................................................... 51
Table 9 Options for storing analysis script information ............................................................ 55
Table 10 Proposed content of plugin information ..................................................................... 56
Table 11 Domain information storage solution alternatives ..................................................... 58
Table 12 Success for usability for basic users ................................................................. 80
Table 13 Success for usability for regular users ................................................................. 80
Table 14 Success for usability for advanced users ................................................................. 81
Table 15 Success for reusability of generic scripts ................................................................. 81
Table 16 Hours spent in the project per activity ................................................................. 90
Table 17 Risks and their solution ......................................................................................... 91
Table 18 Success metric for reflection and critical attitude ..................................................... 94
Table 19 Success metric for reflection and critical attitude ..................................................... 94
1 INTRODUCTION

Océ high-performance production printers are capital goods for their customers. The customers owning these printers are typically in the graphic arts or mass production printing business. The products often come with a contract with an SLA (Service Level Agreement) with KPI’s. The main concerns of the customer are print quality, integration in their environment, flexibility, productivity, and reliability (It should be up when it is needed). Of those points, print quality and reliability are the focus of this project.

Figure 1 shows two potential product release cycles. Products are released to a set of early adopters for verification and validation. An early release enables increasing product maturity by feedback from machines in the field and helps validate customer wishes by learning from their experience. Telemetry helps when used internally and enables the support necessary for an early adopter release. In the end this leads to earlier general market release.

![Product release cycle for VarioPrint i300®](image)

**FIGURE 1 PRODUCT RELEASE LIFECYCLE: REGULAR VERSUS EARLY RELEASE**

When machine quality is sufficient, the product is released to the general market. The VarioPrint® i300 is a high-speed cut sheet printer, which is in the phase of releasing to early adopters. This product follows a release with telemetry: logging machine behavior and sensor values, sending these to an analysis server, and using analysis tools to analyze the behavior. Telemetry speeds up the release by giving insight into machine behavior in the field.

*Machine behavior analysis* in the context of this project is the analysis of behavior and sensor logs to find abnormality root causes. This provides opportunities to reduce the number of defects and increase their predictability. In addition, an image is formed of the regular use of the machine. Océ sees effective machine behavior analysis as an opportunity. This project focuses on the VarioPrint® i300 product, since it is the first product to follow the new strategy of utilizing detailed device behavior. More details on this choice and the project scope are in Chapter 2.
Currently, machine module designers and service product specialists have log data available for each machine module. Depending on the machine module, it can take days to gain insight into the behavior of a single machine. In this context, gaining insight means understanding the behavior of the machine. Until now, the number of machines in the field (tens) has been no issue. Prospective growth to over a hundred machines in the field for the next year makes the current approach to analysis infeasible in the near future.

The current approach has opportunities to improve. First of all, analysis is done reactively instead of pro-actively, i.e., an analysis is done on an incident basis and no automatic analysis is created to predict future similar cases. The main causes for this are a lack of automation capabilities and not being able to effectively share knowledge about what problems exist. Sharing results is limited as well: The location of created analysis scripts and their description is often unknown. In addition, these scripts are not designed to be reusable. So, it is hard to find and re-use them for a domain-specific analysis, without copy-pasting and redundant work. More on the current situation and corresponding limitations is discussed in Chapter 3.

To enable machine module designers to gain more insight into machine behavior within their available time, an analysis framework extension was developed enabling domain experts to develop, share, and execute analysis scripts and domain knowledge. Two business cases in company context show analyst working time saved when using this approach. A number of generic and domain specific scripts were created while developing these cases. These scripts have been integrated in the company analysis environment and end-users have confirmed that time is gained by this setup.

The next section provides background information on the Océ as company and the context of this assignment, followed by the outline of the report.

1.1 COMPANY CONTEXT

The context of this project is Océ, A Canon Company. The material for this context is taken from corporate presentations of Océ and Canon. This project is executed within Océ, A Canon Company, specifically within the R&D department of Océ Venlo, The Netherlands. Since 2010, Océ has been a part of Canon, which has 190,000 employees worldwide. Canon is a leader in consumer and professional imaging, with 2015 revenues of 29 billion euros. Their products and solutions include cameras (compact and interchangeable-lens), desktop printers for home and office, office multifunction devices, digital production printers, video equipment, medical equipment, and semiconductor-manufacturing equipment.

Océ itself is a Canon innovation center, with a revenue of 1.9 billion euros in 2015. Océ has 3,600 employees in Océ innovation centers around the world and 12,000 Canon employees sell Océ products worldwide. Océ is a top 10 R&D investor in the Netherlands and its main products include the following printing technologies: large format, continuous feed, cutsheet printing, and sheetfed printing. In addition, they offer workflow software and business services.

Océ’s key growth areas are graphic arts, business services, and industrial printing. Graphic arts is the printing of text and graphics for applications such as newspapers, books, magazines, banners and signage. Océ products offer customers a number of advantages, such as shorter runs, individualization, customization, and high-quality specialized products. Examples of graphics arts production machines are shown in Figure 2 and Figure 3.
Business services offers complete outsourcing and document management. Applications areas include education, healthcare, legal, and printing.

Industrial printing is the graphics and information printing objects such as security applications, labels, packaging, tiles, textile, and wallpaper. It is also functional printing, such as the application of printing technology in the production of printed circuit boards (PCBs), the printing of 3D objects, and other applications.

Océ Venlo is the Océ innovation and technology center in the Netherlands. It employs over 2,400 people. Venlo is the home of the Océ Strategic Business Units for Wide Format Printing Systems, Sheeted Presses, and Business Services. It is also home to the OIP Production Systems Group Venlo, which specializes in cutsheet and sheetfed printing systems. The main activities are Research and Development and Manufacturing and Logistics of large format, sheetfed, and cutsheet printing systems, as well as supporting consumables and spare parts. A prominent example for cutsheet printing is the VarioPrint® i300, as seen in Figure 4.
As mentioned before, the VarioPrint® i300 is of particular interest, as the VarioPrint® i300 project has an early release strategy. In line with this strategy, these machines produce a large amount (gigabytes per day) of log data useful for behavior analysis. This log data flows from the machines in the field towards Océ. This data is an input for a number of data analysis tools and eventually result in valuable information for both Océ and their customers. This project concerns the data and analysis tools utilized by the Research and Development (R&D), Service, and Manufacturing and Logistics (M&L) departments of the VarioPrint® i300 development.
1.2 OUTLINE

**Figure 5 Report outline**

The general outline of this report can be found in Figure 5. First, the analysis space of this project is covered, which contains information on the context and description of the problem, and the rationale for scoping towards requirements. This part shows my capabilities for requirements elicitation, scoping, stakeholder involvement, and prioritization.

After the requirements, the solution space describes the design aimed to fulfill the most important requirements. Through these chapters my capabilities for design, experimental analysis, and showing product value can be assessed. Finally, the process space describes aspects of the process and constraints within this project. These chapters show my skills in risk and time management, as well as reflections.
ANALYSIS SPACE

2 COMPANY VALUE OF DATA ANALYSIS

Machine behavior analysis provides insight into the behavior of machines in the field. Potential insight benefits include reducing the number of defects and increasing machine and defect predictability. This has a positive impact on release time and quality, which saves costs. Section 2.1 explains in more detail what kind of insight is valuable and how this fits in the company context. Section 2.2 zooms in on the value of behavior analysis of high production machines.

2.1 POTENTIAL COMPANY GAINS

Within the company context, there are multiple opportunities for data analysis. Management, marketing, and sales can use product Key Performance Indicators (KPI’s) and trend analysis to understand the product value to change their strategy and bind customers. Machine (including software) module designers and service personnel can benefit from insight into the machine’s usage and physical performance, as measured by sensors. This can be applied to improve design and service. Finally, printer owners, their print floor managers, and service technicians on site benefit from insight into current defects root causes and potential solutions for those defects. This leads to effective machine use and service strategies and less unplanned downtime.

Gaining insight into machine behavior is the main goal of analysis. These are some examples of insight adding value to the context of a high production machine:

- Understanding weakness in parts or configuration, leading to:
  o Improving design of the machine
  o Reducing the number of defects
- Recognizing machine behavior leading up to failures, leading to:
  o Increasing predictability of defects and reducing the cost of service
  o Increasing customer satisfaction
- Understanding the limits and optimal application area of machines
  o Enabling a more effective marketing and sales approach
  o Extending the application area of products

A reason to consider machine behavior analysis is that other companies in a similar field have shown effective business models. Examples are embedded system developers such as ASML (Rozinat, 2009) and Philips Medical (van Zoest & Luijten, 2014), and research institutes like CERN (Brun, 1997). Furthermore, various analysis technologies are available. Examples are commercial tools like the Microsoft business intelligence stack\(^2\) or Tableau\(^3\). Open sources solutions such as KNIME\(^4\), Rapidminer\(^5\), and Weka\(^6\) exist, in addition to flexible environments such as Jupyter (Python)\(^7\) or R\(^8\). 3) Analysis initiatives exist within the company. Chapter 3 explains this domain in more detail.

\(^3\) [http://www.tableau.com/](http://www.tableau.com/)
\(^4\) [https://www.knime.org/](https://www.knime.org/)
\(^5\) [https://rapidminer.com/](https://rapidminer.com/)
\(^7\) [http://jupyter.org/](http://jupyter.org/)
\(^8\) [https://www.r-project.org/](https://www.r-project.org/)
2.2 ANALYSIS FOR HIGH PRODUCTION MACHINES

Obtaining insight into machine behavior can benefit multiple areas of the company. This section discusses what kind of behavior is considered, who is to perform analysis, and who benefits from insight in behavior. Section 2.2.1 describes the options and choice of focus in the company and Section 2.2.2 describes the stakeholders (and their concerns) in this context.

2.2.1 OVERVIEW OF POTENTIAL STAKEHOLDERS

Multiple departments potentially benefit from insight into the machine behavior. To select stakeholders, a number of these departments were considered. Figure 6 shows the general overview of the beneficiary groups and how they interact in the company context with respect to data analysis. The potential beneficiaries are grouped by the amount of influence this project can have on them and the department/group they belong to.

**Figure 6 Simplified Company Structure with Respect to Data Analysis**

The departments and groups shown in Figure 6 are outlined below.

- **Management and Business Unit:**
  - **Management**: Responsible for company-wide strategy
  - **Business Unit**: Responsible for strategy on initiation of new projects and product groups in a specific market area.

- **Marketing, Sales:**
  - **Marketing and Sales**: Responsible for presenting machines to the market and selling them. Within the project scope, their needs are similar enough to group them.

- **Regional and national sales organizations:**
Regional /National service: Responsible for arranging service visits. They are the first contact of customer for incidents. They also aggregate information on incidents in their region.

Field service technician: Responsible for handling incidents at customer site or as part of a planned visit

Account managers: Responsible for maintain a relation with the customer, selling machines, getting feedback, and giving suggestions.

- R&D (Research and Development): Responsible for designing and developing new products and prototyping product concepts.
  - Machine module designer: Responsible for design and development of a machine module, including all data collection and analysis

- M&L (Manufacturing and Logistics): Responsible for manufacturing and shipping machines to customers.
  - M&L Engineer

- Service: Responsible for finding solutions to escalated incidents.
  - Service product specialist

- Customer: Owner of a machine
  - Print floor manager: Responsible for functioning and configuration of machines

This project focuses on the product development and introduction stage with the machine module designers and service product specialists as main stakeholders. This focus is chosen since 1) it has the highest expected impact in a short time span, as a small step can help an expert make next steps, 2) it has accessible stakeholders, 3) the stakeholders have technical expertise, imposing less restrictions on user interfaces, 4) the stakeholder questions are specific and mostly deterministic, and 5) future projects are planned to have a similar release approach.

They main reasons not to focus on the other processes are:

- Marketing/sales/strategy/planning: Analyzing trends for these parties is complex, data is difficult to obtain: 1) it is unstructured, 2) there are many different sources, 3) it is often not originating from the company, 4) analysis tasks are not regular and hence hard to automate, and 5) data is not hard, i.e., truth on the meaning is subjective. The potential stakeholders are also hard to influence in this project scope.

- Printer owners and their staff, and national and regional service organizations: False positives/negatives must be low in predictions, which is infeasible in the time limitation of this project (9 months). They are also almost impossible to influence in this project scope.

Hence, focusing on machine behavior analysis for the machine module designers and service product specialists has high potential impact and a relatively low risk. Feedback from machine module designers and service product specialists supports this conclusion.

2.2.2 STEAKHOLDERS AND THEIR TASKS IN PRINTER CREATION PROCESS

The main stakeholder types are machine module designers and service product specialists. Machine module designers are responsible for designing, prototyping, and testing machine modules. The also provide the logging implementations of the modules they design. They are the domain experts and they can use their knowledge and logs from the logging implementation for analyses. They mainly use analysis to understand the behavior of the machine either in response to an incident or based on a hypothesis or comparison with an existing model of behavior.
Service product specialists handle incidents escalated from regional service organizations and build up service knowledge. They mainly use standard tools and analysis algorithms to find solutions. If they cannot find a solution, they rely on machine module designers of the R&D department for root cause analysis.

Below, some of the analysis benefits described in Chapter 2 are linked to specific tasks for machine module designers and service product specialists. Their place in the company structure can be seen in Figure 7.

- Understanding weakness in parts or configuration, leading to improving design and reducing defects:
  - Case 1: A machine module designer knows of a print quality problem. The hypothesis: A defect in climate control of the machine causes the quality problem. An analysis is done on the ~25 relevant logged variables using an advanced analysis tool. This analysis takes two days and the machine module designer finds that the air pressure for a vacuum pump drops. A physical inspection reveals an air tube defect. This air tube has worn much faster than expected: within three months while it should last the printer lifetime. A different type of air tube is used in future product releases, reducing the defects caused by this quickly wearing part.
  - Case 2: From experimenting with a lab machine, a machine module designer knows of a quality problem which is probably caused by a defect in the print belt. The machine module designer gets information about the machine and time interval in which an incident occurs. He then analyzes the five logs of modules related to this problem, within this time interval. This analysis takes a few hours (for one machine). The sway of the print belt perpendicular to the paper movement is identified as the main problem. Further analysis shows a big factor in the sway is the weld between two parts of this print belt. This knowledge is forwarded to the manufacturing process to improve future product quality.

- Understanding machine behavior leading up to failures and predictability of defects:
  - Case 3: A service product specialist is requested to investigate an incident on print quality. He gets information on the time and machine where the incident occurred. The service product specialist then spends about one hour using a basic analysis tool, running a number of standard analyses (e.g., checking the ink dispensers, the nozzles, the print belt). Depending on complexity of the analysis, there are two flows:
    - 3a: The service product specialist finds out that the print belt performs outside of factory-specified thresholds. He then plans a maintenance visit to fix the problem.
    - 3b: The service product specialist cannot perform the analysis with the tools given and requests a Machine module designer to investigate. Further investigation shows an LED used for measuring paper positions breaks, reducing the quality of prints due to lack of corrections. Analysis shows that the power usage of this LED increases for up to two months before breaking. Now, a predictor can be created to automatically check if LED power usage is significantly growing and signal the need for replacement.
To conclude, insight into machine behavior and status is expected to be valuable. Insight can be attained more effectively by helping machine module designers and service product specialists with their analysis tasks. Hence, it is valuable to let these stakeholders obtain insight using an effective approach. Chapter 3 describes the current way of working of these stakeholders and the corresponding limitations and opportunities. Chapter 4 describes a precise problem statement, followed by a set of corresponding improvements in Chapter 5.

3 COMPANY ANALYSIS TOOLS AND LIMITATIONS

Machine module designers and service product specialists are the main stakeholders. Section 3.1 summarizes the investigation into data analysis for these stakeholders in the company and Section 3.2 summarizes a brief investigation into analysis in general. More detailed investigation results on data and analysis within the company can be found in Appendix B.

3.1 INTERNAL ANALYSIS INVESTIGATION

Section 3.1.1 describes the current situation of analysis for machine module designers and service product specialists in the company. Section 3.1.2 discusses the limitations in this situation.

3.1.1 CURRENT SITUATION

Currently the machine module designers and service product specialists of the VarioPrint i300 product have behavior logs available for analysis, also known as functional logs. Most of the machine behavior logs are sensor output logs. An analysis is executed based on some hypothesis, existing domain model, or incident in the field. An analysis can require hours of work per machine. Even now, with tens of machines, it is difficult to effectively gain insight into machine behavior and status.
behavior. With a prospective growth to over a hundred machines of this type in the next year and expected changes of hardware, manual analysis becomes infeasible.

In addition to a growth in machines, there is also an evolution in data. Data formats change and sensors are added and removed as seen fit. When upgrading machines with new part types, both expected behavior and corresponding data is often influenced as well. Finally, machine or data log settings influence how the log data is built up and how data is preprocessed at the machine.

At this moment machine module designers and service product specialists use Excel, Matlab, the Diagnostic Framework (DF, a tool with standard analysis scripts for time-based analysis), and ODAS (A flexible framework for analysis based on Python). Figure 8 shows the main characteristics of these tools. ODAS is the most promising as it offers the following: automation of analysis, flexibility to tweak existing analyses, and comparison between machines (and different datasets in general).

![Tools Overview](image)

**Figure 8 Overview of tools used for analysis**

The following are details on these tools and examples on how these tools are used:

- For Excel, users develop a local Excel-based script, using a single file, often for a short time period and a small number of variables. These files are often not shared.
- For Matlab, users develop local Matlab scripts. These files are often not set up in a generic way and are not shared.
- For DF, most users choose a particular machine, time period, and predefined Python script to be run. These scripts are shared, but only advanced users can create them. Hence, flexibility is low for regular users. Currently DF is the most commonly used analysis tool amongst machine developers and service product specialists. The typical flow of an experiment for DF is shown in Figure 9.
- For ODAS, users create (basic) Python scripts, re-using parts of examples of other scripts. These scripts are available to one another, but are often not used as it is difficult to search other scripts for reusable parts.
Analysis tasks normally follow as a reaction to some unexpected behavior, e.g., a customer call to service, or by an observation done at R&D or M&L. This input is the basis for a hypothesis. Stakeholder interviews showed a general process to be followed, as seen in Figure 9. After completing a cycle the analyst (hopefully) has gained a new insight or a new idea for an experiment. Currently these processes are either built from scratch or done by reusing existing standard scripts. When only using standard scripts, there is not enough opportunity to tweak analysis to domain-specific needs. Hence, flexibility in the analysis environment is important.

To conclude, analyses on machine behavior normally consist of extracting data, performing exploratory experiments, filtering, and visualizing data. They are executed based on incidents in the field or a hypothesis from machine designers. Analyses easily take a day and the number of machines that are available for analysis is growing steadily. A number of tools, in particular DF and ODAS, are available to assist analysis.

3.1.2 LIMITATIONS IN CURRENT SITUATION
The company analysis situation sketched in Section 3.1.1 has its limitations and hence opportunities. Within DF, there are standard methods for analysis. However, it is difficult to create a new type of script for most users. In addition, it is cumbersome to do an analysis over the fleet, as an analyst must manually redo the experiment for every machine. This will become infeasible with expected growth of the number of machines in the field. This section describes the current limitations in more detail. The scripts within DF are set up as all-in-one solutions and hence hard to reuse.
ODAS provides a solution to the inflexibility of DF by using the Jupyter and Anaconda technologies to allow plugging in new analysis tools and visualizers by others than the DF developer. In addition, it has planned to deploy 1) a standard structure for domain-specific analysis scripts containing reusable parts and 2) an improved file format (HDF5) with standardized extract-transform-load scripts. However, by default the ODAS environment is complex for non-programmers. In addition, re-use of script parts leads to copy pasting, rather than referencing, due to the structure of Jupyter Notebooks. This project has been executed using the released parts of ODAS.

At the time of the project, it is cumbersome to analyze a combination of datasets with ODAS (without the HDF5 data and corresponding scripts). I.e., hours of work could be spent on aggregating data for a small experiment over machines, where many of such small experiments might be needed.

Additionally, domain knowledge, such as the expected and exceptional behavior of a signal, is of great importance to perform analyses and interpret their results. At this moment, however, there is no way to effectively share this domain knowledge, any other meta-data, or annotations. Currently the most common way of sharing is through documents. This is not effective for data analysis, as the distance to the data and scripts used is too far to benefit from the stored knowledge.

Another limitation is that machine module designers often do not know where to start data analysis, even though their data is restricted to a module where their expertise lies. At this moment there is no automatic way to sift through all the data within a specific module. There is a need for pinpointing exceptional behavior and root causes quicker, such that more in-depth experiments can be done on these cases.

Finally, there is a mismatch between analysis tasks and the expertise of machine module designers. Even though machine module designers have the capability to learn basic software development, most do not currently have expertise in it. Even though they have expertise in scientific data analysis, their knowledge on big data and business intelligence is limited. This introduces a requirement on the product in terms of usability: an offered analysis script must be usable without extensive data science and programming skills.

To clarify these limitations, consider these concrete examples of difficulty in doing analysis:

- It is difficult to find and use a fitting scientific analysis techniques for a typical user, e.g., to translate “I want to know if all machines in the fleet behave in this way, or just a few” into an anomaly detection or trend analysis library to use.
- The chosen file format has files split per machine and per time interval in loose files. There is no database management or querying possible on this file structure, which introduces complex data loading scripts for every new analysis.
- In the case of using Matlab, a custom script has to be made for loading the .csv files into Matlab and doing transformations on these files. Scripts are not shared, or re-usable, and hence a lot of redundant work is done.
- When using DF, there is no way to compare between machines. In addition, only adept Python users are capable of creating analysis scripts from scratch.
- When using ODAS:
  - Creation of entirely new analysis scripts is only feasible for adept Python users
  - There is no support for sharing of domain knowledge
To summarize, insight into machine behavior can be gained more effectively by providing accessible, reusable scripts for data loading, preprocessing, general data analysis, and visualization. Within the ODAS project a plan and prototype exist to support data loading and preprocessing. An opportunity exists in improving the accessibility of analysis scripts and the reusability of scripts developed both in the company and by third parties. In addition, assistance in finding root causes of exceptional behavior is valuable. Finally, sharing domain knowledge to be used in analysis should be more effective.

3.2 EXTERNAL ANALYSIS INVESTIGATION
This section provides a summary of third party data analysis solutions. Data analysis in itself is a very broad topic, covering a large number of sub-domains, mainly in the areas of data importing, cleaning, merging/linking, filtering, and visualizing. A short investigation with multiple types of analysis tools, frameworks, and all-inclusive solutions was performed. The techniques to look at were chosen based on being well known, being unique in their approach, and/or being used within the company. The solutions are considered sufficient to gain a broad view of pros and cons in available analytics solutions.

A number of widely used statistical and data mining techniques from open source (ProM9, R10, Jupyter11) and commercial sources (Rapidminer12, WhizzML13, KNIME14, Cortana15, Lavastorm16) are important to consider for potential use and getting insight into good practices. In particular, the company is already familiar with Jupyter, Rapidminer, Lavastorm, and Cortana.

Pros and cons were gathered through research of the analysis tool creator websites and documentation. The dimensions to look at, as specified in Section 3.1, are accessibility of data analysis techniques and the reusability of techniques. To summarize, the relevant positive features identified in data analysis frameworks are:

- Independence of input sources
- Independence of deployment
- The possibility to create and share a new node/library/package/process easily
- The possibility to tweak a node/library/package/process
- Uniform UI towards nodes/library/package/process
- Ease of writing and sharing scripts and adapting existing scripts
- Ease of finding a relevant node/library/package/process

The analysis frameworks and corresponding pros and cons are discussed in Section 8.1 in more detail.

---

9 http://www.processmining.org/prom/start: Open Source framework for process mining algorithms
10 https://www.r-project.org/: A free software environment for statistical computing and graphics
11 http://jupyter.org/: Open source, interactive data science and scientific computing across over 40 programming languages.
12 https://rapidminer.com/: Prep data, create models, and embed in business processes fast
14 https://www.knime.org/: open solution for discovering the potential hidden in data, mining for fresh insights, or predicting new futures
16 http://www.lavastorm.com/: business applications for agile analytics
4 PROBLEM STATEMENT

Behavior analysis for high production machines can be valuable to gain insight into abnormal machine behavior, leading to better design and more effective service. The main stakeholders to help gain insight are machine module designers and service product specialists. The opportunities to improve machine behavior analysis are: 1) Improving the accessibility and reusability of data loading, preprocessing, analysis, and visualization scripts, 2) assisting to find root causes of exceptional behavior, and 3) making analysis accessible and domain knowledge sharing effective.

Figure 10 shows a subset of the flows shown in Figure 6 to focus on: the analysis process of machine module designers and service product specialists. Behavior data flows primarily from machines at customer site. Analysis requests flow from escalated incidents and hypotheses within R&D. The goal is improving the effectiveness of the analysis step.

The envisioned solutions are summarized below. The goal is not to create a holy-grail algorithm that effective performs analysis in a general case, as this is infeasible in a changing environment. Instead, we aim to allow domain experts to add and evolve their own approaches and analysis scripts on demand, as their environment evolves with them. In addition, we aim to stimulate reuse of both analysis scripts and insight in machine behavior. The main desired improvements are summarized below.
1. To improve the accessibility and reusability of data analysis scripts, we should:
   a. First, we need to enhance the current analysis tools in the company with an analysis script structure to support:
      i. Easily adding a new script, i.e., minimal overhead for the developer, but having a sufficient structure to allow finding and re-using for the user
      ii. Easily executing and configuring scripts, i.e., a domain expert without computer science expertise should be able to adapt an existing script to fit their specific needs
   b. In addition, we need the right level of generality in scripts, such they can be re-used, i.e., enough generality to be applicable in more than one analysis, but not so much generality that implementation becomes complex and time-consuming for a non-computer-scientist.

2. To assist in finding root causes of exceptional behavior, we should:
   a. Provide a script to identify exceptional cases from a large group, e.g., in a fleet of machines, over a large timespan, or over different sets of sensor data.
   b. Provide a script to find related effects for interesting behavior, e.g., finding there is erratic climate behavior when many prints have low quality.

3. To reduce redundant work and share insights by providing a structure for effective domain knowledge sharing.

---

**Figure 11 Improving current Diagnostic Framework-based analysis cycle**

To conclude, Figure 11 shows how we want to support the current analysis process. To gain more insight within the available analyst working time, there are three opportunities: 1) To provide a structure for easy creation, use, and sharing of analysis script, 2) To provide scripts for finding exceptional behavior and related causes, and 3) To provide a way to effectively share domain knowledge. The company should also be able to expand this structure and corresponding scripts themselves.
5 COMPANY ANALYSIS TOOL IMPROVEMENTS

This section elaborates on the analysis tool improvement opportunities expressed in the problem statement (Chapter 4). Improvements are expressed as use cases and quality attributes, with related actors. First, the main actors of these use cases are described in Section 5.1. The requirements related to accessibility and reusability can be found in Section 5.2. Next, finding root causes of exceptional behavior is covered in in Section 5.3. Afterwards, the requirements for sharing domain knowledge are shown in Section 5.4. Finally, the project-level design criteria are shown in Section 5.5 and the opportunities are summarized.

5.1 ACTORS IN THE SYSTEM

Machine module designers and service product specialists perform certain actor roles in the desired system. These actors are categorized on the basis of their knowledge and expertise. In particular there are two important kinds of knowledge: 1) the domain knowledge of the log data and the machine under investigation and 2) the knowledge of programming and data analysis. Figure 12 shows the categories based on these knowledge types. Note that Regular users can also execute Basic user tasks and advanced users potentially execute all User type tasks.

Basic users are those with low domain and data science knowledge. They need a standard script to apply to known problems. Their ideal interaction involves simple configuration and pressing a run button.

Regular users can run simple analyses, but have a higher level of domain knowledge and enough data science skills to adapt an existing script to fit their needs, e.g., finding and re-using existing scripts, configuring a filter to take only a certain media type\(^{17}\).

Advanced users have both domain knowledge and data science expertise. They create new generic scripts to use and adapt. A combination of a data science expert and a domain expert can substitute for one advanced user.

![Figure 12 Main users for system](image)

**Figure 12 Main users for system**

The above users are based on the machine module designers and service product specialists introduced in Section 2.2.1. Service product specialists are often basic users, while machine module designers are often regular users. Some machine module designers fit the profile of advanced users and are currently heavily relied on for analysis. This project aims to gain more insight by enabling both advanced and regular users to create and share analysis scripts. The basic user can profit from any standard scripts following from the regular and advanced users’ efforts.

\(^{17}\) Media type refers to the type of material to be printed on, for example Canon Black Label zero 80 gram A4 paper. A media type can also be non-paper material, such as a cardboard or textile.
5.2 ACCESSIBILITY AND REUSABILITY

This section describes the needs for a structure to easily create, find, and re-use analysis scripts (See Figure 13). The main needs for such a structure are hence accessibility and reusability: to have the right level of generality, to easily add new scripts, and to easily execute and configure scripts. First, let’s consider what accessibility and reusability mean in the context.

Accessibility consists of:
- An easy way to find the available libraries, scripts, and methods for machine behavior analysis, e.g., an overview or search method in the analysis environment
- An easy, consistent, way to configure and run those libraries, scripts, and methods (i.e., a machine module designer should be able to translate their domain knowledge to configured scripts)

Reusability consists of:
- An easy way to add new base analysis scripts, e.g., offering a (possibly wrapped) third party library/tool or implementing a script from scratch. This is mainly for advanced users.
- The right level of generality for offered scripts and approaches for each user:
  i) Basic users want to have access to standard scripts in which only simple configuration settings have to be changed, e.g., a machine ID, a time period, or a type of media.
  ii) Regular users want to have access to base analysis scripts and easily apply their domain knowledge, e.g., an anomaly detection script in which they can configure an interesting dataset and filter based on known thresholds.
- Portability: Independence of technology given the dynamic environment, e.g., if Matlab is no longer supported, the Matlab analysis scripts should be easily replaceable by analysis scripts implemented in other technologies.
Figure 14 shows the overview of the relevant use cases and is followed by use case descriptions and priorities. The main goal of these use cases is to verify the flexibility, accessibility, and reusability of the framework to be developed. Priorities are assigned according to the MoSCoW method\textsuperscript{18}. Figure 15 shows the overview of the related quality attributes and is followed by quality attribute metrics.

**Figure 14 Accessibility and reusability use cases**

The use cases of Figure 14 are now instantiated in the domain of the project.

**Use-case 1** Find analysis script with search function [COULD]

- **Step 1:** User enters search query, e.g., “media type filter.”
- **Step 2:** System presents list of relevant analysis scripts.
- **Step 3:** User selects analysis script to look at.
- **Step 4:** System opens analysis script in configuration/run environment.

Goal: Verify ease of finding reusable scripts

**Use-case 2** Analyze machine behavior data [MUST]

- **Pre:** User has opened an analysis script, e.g., user opened anomaly detection script using the search of Use-case 1.
- **Pre:** User has imported data to be used, e.g., a set of climate data for five machines.
- **Step 1:** User edits configuration in script to fit needs.
  - i) Example: the user selects machine with id “id111,” a time period from 1-1-2016 until 1-7-2016, and the number of results to show.

\textsuperscript{18} https://en.wikipedia.org/wiki/MoSCoW_method
Step 2: System updates configuration values.
Step 3: User requests system to run analysis script, e.g., by pressing a “run” button.
Step 4: System runs analysis scripts and shows any output, e.g., showing a table or a graph with analysis results.
   i) Example: The system shows the most likely anomalies in the given dataset in the form of a graph with the anomalies pointed out.

Goal: Verify ability for existing analysis in new framework, verify accessibility of analysis

**Use-case 3** Find generic analysis script in overview [MUST]
Step 1: Regular user requests overview of generic analysis scripts.
Step 2: System presents list of all available analysis scripts, e.g., an anomaly detection script and a filter based on latest machine settings.
Step 3: User observes generic script(s) to be re-used.

Goal: Verify accessibility of reusable, generic, scripts

**Use-case 4** Fit analysis script in machine module domain [MUST]
Pre: Regular user has a problem/an analysis in mind.
   i) Example: testing whether the climate of the machine has a strong effect on paper deformation.
Step 1: User finds a generic script (or scripts) to use in the domain specific problem using Use-case 3.
   i) Example: Regular user finds data loading, correlation, and a data linking scripts.
Step 2: Regular user implements domain-specific script with the general scripts and runs the script.
   i) Example: Regular user enters machine, time, and machine module information for the data loading script; they indicate the linking variable to be sheet id for the loaded datasets; they indicate which variables to focus on for correlation.
Step 3: The system executes the user-specific and generic scripts and provides the script outputs.
   i) Example: The system shows the list of top 10 variables correlated with paper deformation.

Goal: Verify reusability for scripts for multiple, different, problem spaces

**Use-case 5** Add generic analysis script [MUST]
Pre: Advanced user has knowledge of a needed generic script from regular users.
Step 1: Advanced user develops a generic script in their preferred environment.
   i) Example: Advanced user develops a data loading script in PyCharm on their local machine.
Step 2: System provides general interface to conform to.
Step 3: Advanced user adds generic script to the framework adhering to interface.
Step 4: System makes script available for use and overview.
   i) Example: System adds script to be findable in Use-case 1 and Use-case 3 scenarios.

Goal: Verify flexibility of framework w.r.t. changing the available generic scripts

**Use-case 6** Switch to different underlying analysis tool [SHOULD]
Pre: A script exists in a different technology, e.g., an anomaly detection script has been written in Matlab.
Pre: The technology of the existing script is being phased out, e.g., Matlab licenses are not renewed.
Step 1: Advanced user develops generic script as per Step 1 of Use-case 5. They have to fit the same interface as the previous implementation.
   i) Example: Advanced user develops the anomaly detection script in Python with the same functionality, input, and output as the Matlab script.
Step 2: Advanced user adds new implementation to the framework fitting the previous interface.
Step 3: System makes new implementation available for use and overview as per Use-case 5 step 4.

Goal: Verify flexibility of framework w.r.t. different implementations of generic scripts

![Figure 15: Accessibility and Reusability Quality Attributes](image)

**Figure 15: Accessibility and Reusability Quality Attributes**

Figure 15 shows the quality attributes related to the use cases of Figure 14. These quality attributes are described in detail below. Performance is not focused on as a quality attribute, even though it is important to usable analysis. It is acceptable to perform a small experiment fast and then let a large experiment run overnight once this method is proven to work.

**Quality attribute 1 Usability of analysis scripts [MUST]**
Description: Ease with which analysis scripts relevant for the current problem can be found and executed
Example 1: A machine module designer should have an easy overview of the available scripts and their purpose, such that they can identify a relevant script.
Example 2: A machine module designer should spend little time on configuration of an anomaly detection script if they have a relevant dataset as input.
Quality attribute 2  Reusability of analysis script [MUST]
Description: Ease of finding an applicable script to reuse and the ease with which it can be reused
Example 1: A machine module designer looking to do a correlation analysis should be able to find and reuse a correlation analysis script from another module or a generic correlation analysis script.
Example 2: A machine module designer looking to do anomaly detection for the first time should be able to find relevant generic scripts: data loading, data linking, and anomaly detection.

Quality attribute 3  Extendibility with new analysis script [MUST]
Description: Ease of extending the platform with a new analysis script
Example: A machine module designer in the role of advanced user should not spend a large amount of time fitting a new generic script (say data linking) in the existing framework.

5.3 FINDING ROOT CAUSES OF EXCEPTIONAL BEHAVIOR

![Diagram](image_url)

**Figure 16 Finding root cause focus for DF workflow (see Figure 11)**

Providing support for finding root causes of exceptional behavior is the next goal. Figure 16 shows the next how this goal fits in the DF workflow. Consider two focus points: Finding exceptional cases in a large group (e.g., a fleet of machines), and finding related effects for interesting behavior. During this project, a number of interviews and feedback sessions were performed with machine module designers and service product specialists. From these sessions, anomaly detection and correlation came out as the most promising areas of data analysis to consider. Details on the interviews and scoping process can be found in Appendix A.
Given these areas of interest and the context of analyzing machine behavior, these are the goals:

1. Make it easier to find the interesting parts of data to look at and have a starting point for more in depth analysis
2. Find related effects and root causes of problems

For both topics, the purpose is creating a number of generic scripts as described in Use-case 5. These scripts also serve as a validation of the previous use cases. The expected functionality of these generic scripts is described in use cases below and an overview is shown in Figure 17. The goal of these use cases is to verify the applicability of third party analysis techniques in this context and the ability of myself to analyze the data in the company.

**Figure 17 Anomaly-detection-related use cases**

**Use-case 7**  Find most anomalous moments for sensor data [MUST]

Pre: Regular user knows of anomaly detection script, e.g., through Use-case 3.
Step 1: Regular user collects and filters data used for anomaly detection for one sensor.
  i) Example: Regular user collects a dataset for the climate of one machine. The user then filters this dataset based on whether the machine is switched on or not.
Step 2: System retains input data.
Step 3: Regular user configures and runs anomaly detection script.
  i) Example: Regular user provides the climate dataset as input, and configures the anomaly detection script with a window size for samples.
Step 4: System returns a list of the statistically most significant anomalies.
Step 5: Regular user configures and runs anomaly visualizer script.
  i) Example: Regular user provides the climate dataset and list of anomalies as input.
Step 6: System displays top anomalies together with the relevant data points.
Step 7: Regular user judges shown anomalies and acts upon them.
  i) Note: one of these acts can be to filter out particular anomalies and retry.
  ii) Example: Regular user judges some anomalies to be uninteresting startup effects. Other anomalies show unexpected behavior and give insight for a follow-up experiment.

Goal: Verify applicability of anomaly detection in company domain, verify reusability of anomaly detection, and verify flexibility and accessibility of framework.
Use-case 8  Find anomalous machines in a population [MUST]
Pre: Regular user knows of anomaly detection script, e.g., through Use-case 3.
Step 1: Regular user collects and filters data used for anomaly detection over multiple machines.
  i) Example: Regular user collects datasets for the climate of multiple machines in the fleet. They then filter this data based on whether the machine is on or not.
Step 2: System retains input data.
Step 3: Regular user configures and runs multi-set anomaly script.
  i) Example: The same as Use-case 7, but all the datasets are used as input.
Step 4: System returns a list of the statistically most significant anomalous machines.
Step 5: Regular user configures and runs anomaly visualizer script for multiple sets.
  i) Example: The same as Use-case 7, but all the datasets are used as input.
Step 6: System displays top anomalous machines with their behavior.
Step 7: The same as Use-case 7, step 7.

Goal: Verify applicability of anomaly detection in company domain, verify reusability of anomaly detection, and verify flexibility and accessibility of framework.

Use-case 9  Find relations in sensor data [MUST]
Pre: Regular user knows of relation finding script, e.g., through Use-case 3.
Step 1: Regular user collects and filters data for multiple sensors within the same machine.
  i) Example: Regular user collects data on the climate and the print rejects.
Step 2: System retains input data.
Step 3: Regular user configures and runs relation finding script.
  i) Example: User provides the data for multiple sensors and indicates which correlations are irrelevant (e.g., those within the same printer module).
Step 4: System executes script and presents biggest (both negative and positive) correlations between sensor datasets.
Step 5: Regular user judges shown correlations and acts upon them.
  i) Example 1: User deems the correlations obvious and filters out the uninteresting correlations.
  ii) Example 2: User sees an unexpected correlation and defines a new experiment to verify the relation.

Goal: Verify applicability of correlation techniques in company domain and verify flexibility and accessibility of framework.
5.4 SHARING DOMAIN KNOWLEDGE

Effective domain knowledge sharing is another goal. Figure 18 shows how this goal fits in the DF workflow. Sharing refers to the insight about the input and output of experiments. This insight should be shared or stored somehow to prevent redoing the experiment. The goal is providing a structure to share this knowledge easily.

Domain information should be shared in a common place. In addition, the domain information should be relatable to the data. Figure 19 shows the main use cases related to sharing domain knowledge. Once again usability is an important quality attribute here, as a too high barrier stops people from sharing. The concrete use case description is:

| package | Use cases 3 CAs | Sharing domain knowledge use cases |

**Figure 18 Sharing domain knowledge focus for DF workflow (see Figure 11)**

**Figure 19 Sharing domain knowledge-related use cases**
**Use-case 10** Share data-related domain knowledge

**Pre:** Regular user has gained an insight related to the data.

**Step 1:** Regular user requests system to annotate data of particular machine, variable, and time period with a comment.
   i) Example: Regular user annotates climate data for temperature of machine A with the note “failure of heater” at 2016-07-08, 16:20 (fictional case).

**Step 2:** System stores comment with relation to data.

**Step 3:** Basic user requests domain knowledge on data for particular machine, variable, and time period.
   i) Example: Basic user requests climate data for machine A in the month July of 2016.
   ii) Note: According to stakeholder feedback, the user making the comment will be interested to recall it later.

**Step 4:** System provides list of relevant domain knowledge.
   i) Example: The comment about “failure of heater” is retrieved.

**Step 5:** Basic user requests visualization of data with domain knowledge.
   i) Example: Basic user provides the climate data and retrieved domain knowledge as inputs.

**Step 6:** System provides visualization of data with domain knowledge tags.
   i) Example: The data on the climate is enriched with the “failure of heater” note.
   ii) Note: One can also consider steps 3 and 5 together: visualize annotations by default.

**Goal:** Prove concept for sharing domain knowledge by data annotation, gain insight on value of knowledge sharing in company context

### 5.5 DESIGN CRITERIA

To sum up the requirements and quality attributes expressed in Section 5.2 through 5.4, design criteria are defined as the core values and focus of the project. These design criteria were selected based on a standard list of PDEng design criteria. They were then enhanced to fit company context and stakeholder wishes. A process-related design criterion is covered in Chapter 14.

**Design criteria in scope**

From the quality attributes in Figure 15, we have usability, reusability, extendibility, and portability. In this project context, the most important factor for success is making analysis scripts available for machine module designers, and more importantly, allowing them to create and share their own scripts. Hence, usability and reusability are considered the most important and chosen as design criteria.

First we consider usability. The meaning of usable varies for the different stakeholder groups (basic, regular, and advanced users as in Figure 12). Hence, a metric is defined per stakeholder group and described within the company context. For each metric the steps of maturity are indicated with a one to five rating, where the arrows indicate the expected starting point and aimed growth within this project.
The focus of progress in these metrics lies with the basic and regular users. The workload for analysis is currently focused on the advanced users, as they are the only ones capable of creating new analysis approaches in the available analyst working time. To increase analysis potential and go towards automated analysis, the other groups should be helped get access to the advanced user's results. This brings us to the next topic, reusability.

19 Example filter: Consider only the values within the known signal threshold
Reusability concerns sharing analysis scripts between anyone performing analysis, in particularly the machine module designers. It also concerns reuse of (domain) knowledge. The success metric focuses on the ease with which a machine module designer can find and reuse existing scripts, especially generic scripts. A similar success metric can be constructed for basic users with domain-specific scripts, but this is out of scope as the ODAS project has already shown a prototype for sharing these scripts.

**Table 4 Success metric for reusability of generic scripts**

<table>
<thead>
<tr>
<th>Rating</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Regular user can only use local files</td>
<td>A machine module designer works only on local machine</td>
</tr>
<tr>
<td>2</td>
<td>Regular user can access reusable scripts, but cannot easily find them</td>
<td>An advanced user creates a reusable data loading script, but machine module designer does not realize it exists</td>
</tr>
<tr>
<td>3</td>
<td>When advanced users create generic scripts, regular users should be able to find these scripts based on manual search</td>
<td>A machine module designer creates an anomaly detection script; another can scroll through existing scripts and find the applicable script</td>
</tr>
<tr>
<td>4</td>
<td>When advanced users create generic scripts, regular users should be able to automatically search for them</td>
<td>A machine module designers created an anomaly detection script; another can automatically find it based on “anomaly detection” name</td>
</tr>
<tr>
<td>5</td>
<td>When advanced users create generic scripts, regular users should be able to automatically search for aspects of them</td>
<td>A machine module designer creates an anomaly detection script; another can automatically find it based on “root cause analysis” keyword</td>
</tr>
</tbody>
</table>

**Design criteria out of scope**

The design criteria considered least applicable in the standard list are societal impact and economic realizability. Societal impact is not regarded because there is no direct link between this project and general society. Indirectly, society might benefit from the process optimizations achieved in this project, but this is expected to be insignificant. Economic realizability is excluded because of the project focus. This project aims to give value within the company, but is not directly concerned with making a sellable product or business case for machine owners. In addition, precisely estimating the gains of the process optimizations in this project is not feasible. To be specific: The main monetary gain in making analysis more effective is a faster final release of a product (as shown in Figure 1). However, proving such an effect is infeasible in the project influence and time.

To conclude, the main design criteria for this project are usability and reusability for all the basic, regular, and advanced users as they help the effectiveness of analysis and add an analysis opportunity for basic and regular users through sharing. Societal impact and economic realizability are disregarded, as this project cannot measurably contribute to them.
6 INVESTIGATION: ANOMALY DETECTION AND CORRELATION

To get familiar with the potential techniques in anomaly detection and correlation, it is necessary to investigate promising analysis techniques. We try to identify which tools, techniques, or approaches can be used for demonstrators. From the problem statement (Chapter 4), techniques to research should be able to “identify exceptional cases from a large group” and “find related effects for interesting behavior.” The topics closest to these needs are, as argued in Section 5.3, anomaly detection and correlation.

The investigation does not aim to give a full overview of the state-of-the-art in analysis techniques, and neither does it aim to find the best possible technique to apply in this context. The goal is finding a technique good enough to add a valuable analysis method and make it available through a demonstrator relevant to the domain.

This chapter describes the research results of two concrete analysis topics fitting this improvement direction: Anomaly detection (Section 6.1) and correlation (Section 6.2). Investigation on analysis frameworks has been mentioned previously (Section 3.2). Although sharing of knowledge has plenty of research material available, it was deemed more effective to look at the specific situation and needs of the company, rather than the situation in the research world. The company needs are to share domain knowledge and insights directly linked to the machine data and analyses performed on them.

6.1 ANOMALY DETECTION TECHNIQUES

This section gives a summary of investigated anomaly detection techniques. The materials by Chandola (Chandola, 2009) and Idé (Idé, 2014) have been used as main sources to obtain an overview. The former is an often-cited (over 3000) source and the latter is part of the series “Studies in Big Data”20. Appendix C shows the raw research results, which are summarized next.

![Anomaly Detection Technique Groups](http://link.springer.com/bookseries/11970)
Figure 20 gives an overview of the main anomaly detection classes. These are the main technique types:

- Classification-based detection is a supervised approach distinguishing anomalies depending on predicted classes. Example techniques include neural networks, Bayesian networks, support-vector-machines, and rule-based approaches.
- Nearest-neighbor approaches depend on a relation between data density and likelihood of anomaly. Example techniques include $k$th-nearest neighbor, and relative density.
- Cluster-based techniques depend on anomalies belonging to small or sparse clusters, or being far away from cluster centers.
- Statistics-based techniques depend on anomalies being unlikely in the model describing the data. This approach relies heavily on the choice of underlying distribution.
- Information-theoretic-based approaches do not depend on the statistical distribution, but depend on the subset of information used for normal behavior description.
- Spectral techniques have an assumption on a subspace of the data being a good search space for anomalies. This subspace can be hard to define properly.
- Change point detection techniques rely on matching the correct algorithm with a given type of change in data, e.g., a rolling mean for a change in average value.

For each anomaly detection class a number of implementations exist in the field, both as libraries and modules within analysis frameworks. No matter which technique is chosen, the company-domain datasets and signals likely do not satisfy the technique preconditions by default. Using them effectively requires involved preprocessing steps, corresponding with hours of work.

Given the available techniques, the importance of usability, and the lack of need to pick the best performing (in e.g., speed, accuracy, and recall) technique, it is best to focus on an easy to use and already implemented technique. The most promising technique found in the time taken was "A least-squares approach to anomaly detection in static and sequential data" (Quinn & Sugiyama, 2013) as it is a robust, easy-to-use approach for outlier detection. The implementation and use of the library corresponding to this technique is discussed in the solution space.

### 6.2 CORRELATION TECHNIQUES

This section shows an overview of the concept of correlation and commonly used techniques. The most common approach is calculating a linear correlation coefficient for a pair of signals. Signals should be sizable and describe (for the largest part) normal behavior. The most common metrics used are Pearson\(^21\), Spearman\(^22\), and Kendall-Tau\(^23\).

Another important basic concept is whether one deals with single, multiple, or multivariate correlation. Single concerns the relation between a pair of variables $(x, y)$, multiple concerns a linear combination of variables related to another $(x_1, ..., x_n, y)$, and multivariate concerns both multiple inputs and references $(x_1, ..., x_n, y_1, ..., y_n)$. This project started from single correlation, as it is the easiest from a user perspective and is still powerful.

In particular, correlation for time series is interesting, as most data within the company context is time series data. The journal article “An Empirical Evaluation of Similarity Measures for Time

---


22 [https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient](https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient)

Series Classification” (Joan Serrà, 2014) is used as a reference for similarity measures to be used for a correlation analysis. The following types of similarity measures are mentioned:

- Lock-step measures (based on Euclidian distance), which:
  - Assume the temporal space is exactly the same
- Feature-based measures (based on Fourier coefficients), which:
  - Often concern offline processing and live querying
  - Are robust for noise when choosing correct parameters
  - Extract features instead of a time series
- Model-based measures (auto-regressive models), which:
  - Often concern offline processing and live querying
  - Learn models from multiple sets, then compare their features
- Elastic measures, with the following types:
  - Dynamic time warping, which concerns aligning series with the lowest cost; has need of such a cost function, but is robust
  - Edit distance for real sequences, which is an extension of the Levenshtein distance
  - Time-warped edit distance, which is a combination of the previous two measures. It deals well with different sample rates
  - Minimum jump cost dissimilarity, which is a distance-metric based on the principle that small jumps between sequences corresponds to similarity

Figure 21 shows an overview of the technologies mapped to typical problems for machine module designers and service product specialists where correlation can be of assistance.

![Figure 21 Overview of Correlation Technologies and Problems in the Company](image)

From this figure the multiple and multivariate techniques seem the best option, as these allow more complex signal analysis. However, part of the challenge is to find the simplest solution that still provides correlation analysis, such that it can be applied and reused by those without data analysis expertise. Hence, complex solutions or solutions that lack an existing implementation were not chosen regardless of their performance in correlation detection. In particular Dynamic time warping was attempted, but deemed too complex to wrap effectively in the context.

In the end, a univariate Pearson correlation coefficient was chosen as it is easy to use and gives a good initial indication of relations in machine sensor signals from which more in-depth analysis can follow.
7 SUMMARY: EFFECTIVE MACHINE BEHAVIOR ANALYSIS

The previous chapters have discussed effective, accessible, and reusable machine behavior analysis in the context of Océ, A Canon Company. The company is releasing machines to the field to early adopters, in order to gain feedback from the field and achieve product maturity faster. Faster product maturity leads to a faster general market release. Maturity is gained by improving design and making defects more predictable based on behavior analysis of machines in the field. In addition, in the early release devices are less mature, so more defects are expected initially. It is important to quickly analyze defects and propose solutions for the machines in the field.

The group with the most impact to focus on is the personnel designing these machines, the machine module designers. These machine module designers are currently dealing with an evolving analysis platform and spend a long time per analysis, if an analysis is possible with current tools at all. Increasing the effectiveness of these designers and related service product specialists is expected to have the biggest impact on machine design and defect predictability.

Currently, the machine module designers’ process can be improved by offering a way to share generic, reusable, scripts for analysis; this concerns data loading, preprocessing, analysis, and visualization. In addition, designers often do not know where to start an analysis due to the amount of data, even within their module. Also, their expertise does not lie in big data or machine learning. Furthermore, it is difficult to share domain knowledge such that others can benefit from analysis.

Given these limitations, there is a need to enhance the current analysis framework to enable usability and reusability of analysis scripts, based on domain knowledge and scientific techniques. In addition, it should become easier to pinpoint root causes and relations to these causes, which is achieved by providing scripts based on anomaly detection and correlation. Finally, a prototype for sharing domain knowledge is expected to improve reuse.

In the end, the main goal is to provide tools to improve analysis for machine module designers and service product specialists:

- A data analysis framework allowing reuse of generic scripts and analysis ease-of-use for advanced, regular, and basic users.
- An least-squares anomaly detection technique to support analysis
- A Pearson correlation technique to support analysis

Next, the solution space covers the design of the framework extension and demonstrators aimed to improve the current analysis situation.
8 FRAMEWORK BASIS

This chapter and the following chapters discuss how the problem and improvement of Chapter 4 and 5 are satisfied. We discuss a comparison of data analysis frameworks and which is used as a starting point. Then we show how the framework should enable usability and reusability.

8.1 RESEARCH OF CURRENT FRAMEWORKS

From Chapter 5, the important quality attributes for a framework are: 1) usability, 2) reusability for scripts, 3) extendibility for scripts, and 4) portability. The main related functionalities to be supported are: A) Taking an existing script and changing it to fit the domain, B) Allowing for looking over multiple datasets and comparing between these datasets, C) Allowing for automation of analysis, and D) Allowing for sharing of domain knowledge.

In Section 3.2, a list of analysis frameworks under investigation is discussed. The main characteristics of these frameworks are summarized in Table 5. A limited selection of tools from various sources is taken; this is no overview of all tools or all aspects, but a snapshot of potentially useful solutions.

<table>
<thead>
<tr>
<th>Name</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rapidminer</strong></td>
<td>Database independent + connectors for common databases</td>
<td>Plugins must also be installed on studio (client side)</td>
</tr>
<tr>
<td><strong>Goal: Effective analysis</strong></td>
<td>Possible to install plugins (e.g., to write R scripts)</td>
<td>- Extensions depend on the Rapidminer Marketplace</td>
</tr>
<tr>
<td>workflow</td>
<td>Easy to design an experiment and use local data</td>
<td>- No way to extend with new building blocks</td>
</tr>
<tr>
<td></td>
<td>Processes can be stored, shared, and annotated</td>
<td>- No way to write raw code: low flexibility for analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Lack of visualizers</td>
</tr>
<tr>
<td><strong>ProM</strong></td>
<td>Extendable plugin structure</td>
<td>- Hard to find analysis techniques in list</td>
</tr>
<tr>
<td><strong>Goal: Offer process-</strong></td>
<td>Decoupled visualization, analysis, and data loading</td>
<td>- No flexibility unless you develop a plugin yourself</td>
</tr>
<tr>
<td><strong>mining scripts</strong></td>
<td>Standard tools for UI and loading bars when developing plugins</td>
<td></td>
</tr>
<tr>
<td><strong>as a workflow</strong></td>
<td>Platform-independent</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>WhizzML</strong></td>
<td>Highly extendible and reusable: Scripts can be shared and edited</td>
<td>- Custom DSL is complicated and not applicable outside the tool</td>
</tr>
<tr>
<td><strong>Goal: abstraction</strong></td>
<td>High flexibility in analysis tasks: A script can be edited</td>
<td>- No indication for re-use and modularity: References to other code or even other WhizzML files seem impossible</td>
</tr>
<tr>
<td><strong>and re-use of</strong></td>
<td>Portable in browser</td>
<td></td>
</tr>
<tr>
<td><strong>analysis scripts</strong></td>
<td>Modular internal API’s for analysis methods</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| **KNIME**  
Goal: Effective analysis workflow | - Possible to group a process into a reusable “meta node”  
- Drivers for connecting to most databases (portability)  
- Reusable node concept (similar to plugin) is effective | - Completely new nodes can only be made by people proficient with Eclipse plugins and Java  
- Not possible to change nodes ad-hoc: need for Eclipse |
|---|---|---|
| **R**  
Goal: Flexibility and accessible statistics libraries | - Package-based download of valuable libraries  
- Extremely flexible for analysis tasks  
- Portable | - Hard to find the scripts to re-use  
- Creating and sharing packages difficult for non-programmer |
| **Python/ Jupyter**  
Goal: Flexibility and accessible libraries | - Package-based download of valuable libraries  
- Easy for programmer to create own library  
- Extremely flexible for analysis tasks  
- Portable | - Hard to find the scripts to re-use due to a large and difficult to search set of available libraries |
| **Cortana**  
Goal: All-inclusive solution for analysis in the cloud | - It is scalable for being a cloud service, it is modular in its type of services.  
- Parts can be deployed as web services, which makes integration with external tools easier  
- It supports development in common analysis languages (Python, R) and Notebooks | - Sharing locally can only be done by publicly available link  
- Re-use of modules (not as a web-service) and making a small change for context is limited  
- Re-use and integration seem to rely on making a web service |
| **Lavastorm**  
Goal: Effective data preprocessing | - Node-based modularity  
- Possible to create own nodes (extendibility)  
- Ad-hoc visualization possible | - Making a tweak to a node is difficult  
- Custom DSL for expressing logic is far from users experience |
| **Tableau**  
Goal: Effective analysis workflow | - Portable  
- Independent of database/data storage used | - No way to create custom modules  
- No way to adapt existing nodes |

---


25 Notebook is a web-based structure to write text and code, and execute pieces of code. It is explained in more detail at Section 8.2
The common qualities of these tools and their priority within this project:

- The possibility to create a new analysis script\textsuperscript{26} [MUST, Extendibility]
- The possibility to adapt an existing analysis script [MUST, Reusability]
- The ability to annotate data with domain knowledge [MUST, Reusability]
- The ease of automating an existing analysis over multiple datasets [MUST, Usability]
- A uniform UI towards analysis scripts [SHOULD, Usability]
- The ease of finding existing analysis scripts [SHOULD, Usability]
- The ease with which to create or extend an analysis script [SHOULD, Usability]
- Independence from input sources and deployment [SHOULD, Portability]

Regardless of these qualities, no off-the-shelf tool can solve the most important challenge of this project: The application and sharing of domain knowledge to effectively use these scripts.

8.2 **FOCUS OF SYSTEM**

This section discusses the decision for Jupyter and ODAS as a starting point for this project. Given the research results of Sections 3.2 and 8.1, a comparison is made between starting from an analysis framework and starting from scratch.

It is important to enable easily adding and adapting scripts. Adapting should be possible for a regular user (a domain expert with no programming expertise), as they are the ones able to fully interpret data. Given this requirement, the following tools are considered unfit because of the complexity of adding and changing analysis scripts:

1. Rapidminer is not extendible and changeable for regular users.
2. ProM is extendible, but not changeable for regular users.
3. WhizzML is extendible and changeable, but too complicated for any user.
4. KNIME is extendible, but too complicated to extend for any user.
5. Cortana experiment environment is not properly extendible and changeable.
6. Lavastorm is extendible, but tweaking is too complicated for regular users.
7. Tableau is not extendible and changeable.

R and Python. These are programming-heavy and deemed too complex for regular users to effectively tweak experiments. However, Python and R differ from the above tools in flexibility: It is possible to build a structure on top to facilitate ease of use. Examples here are Tinn-R for R and Jupyter for Python. This leaves R and Python as the most promising contenders of the current investigation. There are no clear pros and cons between R and Python when considering the available libraries or way of working. At the point of writing, a Jupyter environment exists within the company as part of the ODAS framework with a user base. Hence, Jupyter and ODAS were chosen as a basis for development. The framework created in this project is deployed as a library within the ODAS Jupyter Python kernel.

Jupyter is a technology exposing Python to users by using web-based "Notebooks." Notebooks are structured in Cells, where each Cell can either be a piece of marked-up text or a piece of code. Code can be executed in the browser environment and results are visualized at the Cell location (See Figure 22 for an example). In combination with a scientific Python package, such as Anaconda, a large number of data analysis scripts and usability requirements are met.

\textsuperscript{26} An analysis technique is represented as e.g., a node, library, package, or process in these tools
Plain Jupyter offers too much freedom, i.e., there are no guiding principles or structures. Previously, similar environments within the company have shown this degree of freedom results in low reusability (redundant work, no standards on input and output). On the other hand, fully standardized tools are not flexible enough to work in a dynamic environment. So, there is need for a compromise where reusability is enabled, but a flexible approach stays possible.

Jupyter does not provide all the needed functionalities out of the box, but does provide all the flexibility needed to expand. Hence, it is a good basis for a solution. Table 6, Table 7, and Table 8 show the current coverage, what is planned to be added for ODAS, and what will be added within this project respectively.

**Table 6 Overview of features provided in Jupyter**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>The possibility to create a new analysis script</td>
<td>Provided by library/import concepts for Python</td>
</tr>
<tr>
<td>The possibility to create and adapt an existing analysis script</td>
<td>Provided by Jupyter Notebooks</td>
</tr>
<tr>
<td>The ease of automating an existing analysis over multiple datasets</td>
<td>Enabled by Jupyter</td>
</tr>
</tbody>
</table>

**Table 7 Overview of features provided and planned in ODAS**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A uniform UI for analysis scripts</td>
<td>Planned with ODAS notebook structure</td>
</tr>
<tr>
<td>The ease of finding existing analysis scripts</td>
<td>Prototype shown for notebook-based search</td>
</tr>
<tr>
<td>Independence from input sources and deployment</td>
<td>Planned with the PyBricks concept: implementation-independent scripts</td>
</tr>
</tbody>
</table>
To conclude, a small set of analysis tools and frameworks was considered to assess pros and cons of analysis frameworks, and to see if one can be used as a starting point. Most of the tools and frameworks were deemed unfit because they are not extendable, do not support reuse, or are too complex for a regular user to deal with. From the final selection of R and Python, Python was chosen as a Jupyter Python analysis framework already exists at the company. This framework was used as a starting point.

9 STRUCTURE FOR EFFECTIVE ANALYSIS
This chapter explains the improvements that were made on top of the ODAS platform. Section 9.1 explains the framework core concept, which is based on a plugin-structure. It discusses creating and extending scripts, a uniform UI, and ease of automation. Afterwards, Section 9.2 describes how adapting scripts is enabled by a common overview of existing scripts. Finally, Section 9.3 discusses how sharing of domain knowledge is facilitated.

9.1 PLUGIN STRUCTURE AND INTERFACE
A structure is needed to support the analysis process, but this structure should not hinder flexibility. In particular, it should support these improvements named in Section 8.2:

- Creating and extending scripts
- A Uniform UI for analysis scripts
- Independence from input sources and deployment

We also discuss how scripts can be created, loaded, and executed in the proposed structure.

Creating and extending scripts
This section discusses how domain-specific and generic scripts fit in the framework. The domain-specific scripts are those re-used for multiple machines or time periods. The generic scripts to be re-used in multiple machine-module domains. For example, using an anomaly detection script in both climate control and print belt. The former is handled by the Jupyter environment (Figure 22), while the structure for the latter is described in Figure 23.

A common practice in the data analysis frameworks reviewed in Section 3.2 is introducing a general concept for a piece of executable code, with a predefined input and output. They are commonly called nodes, plugins, packages, and processes. The pros of this approach are a common interface for the users and a common way to write reusable code. In this project, a plugin-based structure was used for the same reasons. Hence, the proposed framework uses a plugin structure for generic analysis scripts.
This plugin structure allows creating and extending generic scripts. Consider the structure in Figure 23. First, creating and extending scripts is twofold: partly handled by Jupyter and partly handled by this structure. The scripts handled by this structure are generic scripts created by advanced users. Plugins are implemented as Python library classes and follow a standard format: A class with two basic functions:

- `getInformation()` should return the specification of the function in terms of input, output, and main functionality. The creator should make sure this function returns an up-to-date description of the plugin, such that it can be used effectively.
- `runPlugin(parameters: Dictionary)` should run the plugin, i.e., extract the input from the Dictionary structure, perform the logic, and return the result as specified in the `getInformation()` function. The Dictionary structure is a simple key-value dictionary, making it a flexible input format.

![Diagram of plugin structure](image)

**Figure 23 Structure for generic analysis scripts**

A generic script needs to fit the plugin structure and be loaded in the Impromptu library. This library is deployed at the company server, as shown in Figure 24. Impromptu and any plugin in Impromptu are part of a library attached to the Jupyter Python kernel. These plugins can be called by writing appropriate code within the Jupyter Notebook Cells. Generic script plugins (Least squares anomaly detection, correlation, and load data) have instances of the `AbstractPlugin` and `PluginImplementation` classes in Figure 23.
In addition to creating generic scripts, consider the creation of domain-specific scripts which use these generic scripts. This is typically done by regular users in the Jupyter environment. Figure 24 shows an example of a Notebook called “Fixation vacuum outliers.” This is considered a domain-specific script. In this case it uses the generic script for least-squares anomaly detection.

The creation and extension of domain specific scripts, in the form of Notebooks, is covered by the design of ODAS. The Impromptu library functions are an added value here. Regular users can use these libraries and Notebooks of other users to gain inspiration and complete their own analysis. One way to extend generic plugins is with filters for runtime-assessment, e.g., telling an anomaly detection algorithm that all values within the threshold can be ignored. Currently this is implemented as a loop over all filters. A decorator pattern\(^\text{27}\) with filters as decorators is a more structured alternative.

**A uniform UI**

Next, consider a uniform UI to enable regular users to more easily access generic scripts. Once again, the Jupyter platform offers part of the solution as it uses a standardized GUI for finding domain-specific scripts, writing code, and executing code. However, there is no standard structure in Jupyter to run plugins (generic scripts), and hence this has to be provided. From a usability perspective, there is need for a simple interface, i.e., one should need to write only a few lines of code to run a script, and one should be able to use a common interface.

This common interface is provided by the "data analysis framework" class as seen in Figure 23. Consider the runPlugin (pluginName: String, parameters: Dictionary) function of the data analysis framework. The interface to call a plugin is always the same: a name of the plugin and a set of parameters in a Dictionary format. Inside the dictionary can be any values. For this moment, assume these names and corresponding input parameters are known to the user; Section 9.2 explains how this is achieved. Given that the way of calling functions is always the same and the Jupyter GUI is consistent, a uniform UI is achieved.

**Loading and executing plugins**

Finally, given the concept of a plugin, the Notebook environment, and a uniform UI, we discuss how the plugins are loaded and executed. Consider Figure 25. When the framework library is imported, it scans its local plugin folder. It finds all valid AbstractPlugin implementations and corresponding PluginImplementation implementations. Only plugins with both an AbstractPlugin and PluginImplementation file are loaded. References to these files are kept within the framework. These are the AbstractPluginIndex and PluginImplementationIndex objects in Figure 23.

Assuming the plugins are all properly loaded, a user can call it with the runPlugin() function. The framework then finds the correct plugin from the plugins list, instantiates it, runs it, and returns the result. The Jupyter Notebook handles the result depending on the user script, e.g., by providing visualization or a snapshot of data.

**Figure 25 Loading and executing plugins**
Conclusion

A plugin-based structure is used to satisfy easy creation (through a non-restrictive plugin interface), extension (through the plugins and Notebooks), a common UI (though the plugin interface), and easy automation given the combination of Notebooks and generic plugins. In addition, portability is satisfied both by the Jupyter server and a separation of plugin name and function from its implementation. This satisfies a large portion of the functionality and quality attributes named in Section 5.2.

The following steps would improve the current framework:

- Allow advanced users to automatically publish their plugins from an IDE (e.g., PyCharm or WingIDE). At the moment advanced users have to manually paste their code files in the library format.
- Develop examples of wrapping other tools, e.g., Matlab, in a Python plugin to use as a reference for advanced users.
- Make someone responsible for the organization and the approval of plugins when the user base grows. Some form of governance on these plugins and the Notebook scripts is necessary to stimulate re-use and avoid proliferation.

9.2 COMMON OVERVIEW

To stimulate re-use of scripts, users need to have a good overview of the available scripts. To find relevant Notebooks we rely on the structure given by ODAS, as seen in Figure 24. Creating this structure, maintaining it, and searching it for interesting Notebooks is out of scope as prototypes have been created and developments are planned within ODAS. This project will add an overview for the generic non-notebook scripts.

Storing generic script descriptions

In the current situation, even though the structure is in place, it is still difficult for the user to find out about generic scripts in a convenient way. To generate an overview of what plugins exist and what their function is, meta-data about the plugins should be stored somewhere and made available. An overview of the domain-based scripts (in the form of Jupyter Notebooks) is considered out of scope.

Table 9 shows the alternative solutions for storing plugin meta-data. Storing scripts descriptions in the script is a superior option, as it is the most likely location a developer will consistently update descriptions. It also couples the information and code, reducing the risk of losing information when moving files.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>In a known location within the script</td>
<td>- Index/summary can always be generated from scripts themselves.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Developer does not have to edit in separate place.</td>
<td>- Overview must be collected from all script files.</td>
</tr>
<tr>
<td>In a central store referring to the scripts</td>
<td>- No multi-file processing needed for overview.</td>
<td>- Inconsistencies occur when moving files.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Not obvious that developer should edit this file + ownership of the file is vague.</td>
</tr>
</tbody>
</table>
As text in an example Notebook
- Already visualized in nice format.
- Easy to set example format.
- Notebook has to be automatically shared.
- Notebook has to be maintained, introducing read/write/share issues.

In a local store near the script
- Easy to access meta-data through file explorer.
- Inconsistencies occur when moving files.
- Overview must be collected from all script files, hence: location of all scripts must be known (also, might be slow).
- Not obvious that developer should create this file.

Automatically generated from the source code
- Index/summary can always be generated from scripts themselves.
- Developer does not have to edit in separate place.
- Infeasible within project scope.
- Overview must be collected from all script files.

We chose to embed analysis script information in a known location within the script, as it has the lowest risk of inconsistencies. This is implemented by the value of the function “getInformation()” of the class AbstractPlugin (as seen in Figure 23). This function is used to get the stored information. Stakeholders indicate it is fair to expect that authors of generic scripts spend a short time on describing at least the guideline information in See Table 10. In the end the developer is responsible for filling it in.

<table>
<thead>
<tr>
<th>Proposed info</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Short description of the purpose and function of the analysis script</td>
</tr>
<tr>
<td>Input</td>
<td>Precise description of the input format and any assumptions made</td>
</tr>
<tr>
<td>Output</td>
<td>Precise description of the output format and any assumptions made</td>
</tr>
</tbody>
</table>

### Overview of scripts
Assuming the information on plugins is stored, the next challenge is making it available to the user. The process of creating an overview should be automatic to increase the chance of reuse. A logical solution is automatically scanning through all plugins in a given folder and reading the stored information from the getInformation() function. This is similar to creating the plugin index in Figure 25. As this information is collected, it should be presented in a convenient way.

Since the user is already using the Jupyter Notebooks, a logical step is providing the overview of scripts in a Notebook as well (See one example in Figure 26). The current design uses the Jupyter Notebook generation API28. Alternatively one could use an external file or web-server, but these solutions are both more complex and less usable. The current solution provides a list, with no additional search or grouping functionality.

---

This concludes the design for finding generic and domain-specific scripts, satisfying the remaining requirements of Section 5.2. Although the current solution gives a basic overview, there are some improvement points for usability:

- A plain list is acceptable for a small number of plugins, but as the number of generic scripts grows, there will be a need for searching and categorizing them. Ways to improve include (in order of complexity):
  - Categorize the plugins based on a folder structure, e.g., group the data preprocessing plugins together based on their folder
  - Allow for tagging of scripts with keywords to enable searching and implement a basic search function
  - Develop or integrate an indexing/search function based on the script contents
- For all the above holds: there should be someone responsible for governance on the plugins, the structure, and the keywords used to make search effective.

### 9.3 SHARING DOMAIN KNOWLEDGE

The final feature concerns sharing data-related domain knowledge. For example: “This is a startup effect” at a peak in sensor value. Other examples are “Part X was replaced at time Y” and “This drop in the variable value means the hardware has broken.” This knowledge needs to be input, stored, and retrieved on demand.

Preferably one would store domain information with the data itself, following the reasoning shown in Table 9 of Section 9.2. However, the current data format is a folder tree with .csv files containing the data in the leaves. These .csv files have no proper way of storing metadata. Hence, some other way to store domain knowledge metadata and refer to these .csv files is needed. As there are a number of ways to store metadata with data, a comparison is made. Table 11 shows the pros and cons.
### Table 11 Domain information storage solution alternatives

<table>
<thead>
<tr>
<th>Solution</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Structured database (e.g., SQL)</strong></td>
<td>Easy to extend with new metadata sources</td>
<td>Concurrency and transactions are handled</td>
</tr>
<tr>
<td><strong>Semi-structured database (e.g., NoSQL)</strong></td>
<td>Easy to extend with new metadata sources</td>
<td>Concurrency and transactions are handled</td>
</tr>
<tr>
<td><strong>Structured flat file (e.g., .csv)</strong></td>
<td>Easy to develop</td>
<td>Hard to extend with new information</td>
</tr>
<tr>
<td><strong>Semi-structured flat file (e.g., XML)</strong></td>
<td>Easy to develop, Easy to extend with information within file</td>
<td>Concurrent use needs development, Needs editor to properly maintain</td>
</tr>
<tr>
<td><strong>Freeform flat file (e.g., .txt)</strong></td>
<td>Easy to develop</td>
<td>Hard to extend with new information</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Concurrent use needs development, Hard to maintain</td>
</tr>
</tbody>
</table>

A semi-structured database approach is valid, as complex domain relations can be stored. A structured database already exists on the company analysis server. Using this database is preferable at this point, as this setup gives all benefits and it is a small investment.

The project Reflexion was executed in parallel with this project. Reflexion concerns, amongst other data analysis improvements, gathering and reasoning about domain information. This project cooperates with Reflexion in terms of having a common storage and retrieval for domain information. Both this project and Reflexion integrates with the database on the company analysis server, as it fits the needs and the choice for the technology used for a structured database is not critical. More on this database and the schema used can be found in Appendix D.

Currently, a proof of concept has been created that allows users to add tags to sensor data. To refer to the data, the following identifiers are used: A reference to the machine, a reference to the particular subfolder in the logging, a reference to the variable used, and references to the start and end times of the event. The author and their comment are stored as well.

The current design (including a snapshot of part of the Reflexion project) can be seen in Figure 27. Note how the Notebook, Cell, and library concept in the top right corner correspond with the Notebook and library concepts in Figure 24. Reflexion and the framework of this project are seen as libraries within ODAS. There are two steps in the process of sharing domain knowledge:

1. **Adding domain knowledge**: Add a domain knowledge tag to the Domain knowledge database using a common TimeTagger class. The function saveTag() accepts a set of identification parameters, an author name, and a comment name. It saves this information in the database. (See Figure 28) Later, a visual solution should work as an abstraction on top of this function.

2. **Retrieving domain knowledge**: The getTagsInRange() function accepts a set of identification parameters and returns all relevant tags as a ”Tag data” artifact. This data can then be used to enrich data visualizations. (See Figure 29)
Currently users can view and store their comments through a graphical interface, of which the inner workings are shown in Figure 30. Users directly add comments within a results graph rather than adding them as an afterthought in code. Retrieving domain knowledge is still independent from visualization.
To conclude sharing domain knowledge: a proof of concept has been implemented to show the feasibility of enriching data with domain knowledge, satisfying the requirements of Section 5.4. However, there are plenty of possible improvements here:

- First, storing domain information directly with the data should be facilitated. This way it is harder for metadata to get lost and easier to filter data based on meta-information later (e.g., give all data for which there is no “startup effect” tag)
- Furthermore, tagging thresholds or expected values for variables should be possible.
- Next, integrating with the Reflexion project should continue as it matures.
- Then, searching for existing tags is a valuable addition according to stakeholder interviews. Example use cases include finding which tags were written by a particular user (themselves) or tags of a specific topic (e.g., “maintenance”)
- Finally, automatically proposing tags from user-defined tags is valuable. E.g., if a regular user tags startup effects 20 times, the system should propose other potential startup effect moments or tag them automatically.

![Figure 30 Proposed Interaction for Sharing Domain Knowledge](image)

**Figure 30 Proposed Interaction for Sharing Domain Knowledge**

**Conclusion**

In this chapter three main aspects of the proposed Impromptu framework are shown. First, a plugin structure is proposed to tackle easy creation and extension of analysis scripts, a common UI, easy automation, and portability. Then, a common overview is shown to enable reuse and ease of use for regular users. Finally, a prototype is shown for tagging data as a form of sharing data-related domain knowledge. With these improvements, a significant step is made towards a framework with usability, reusability for scripts, extendibility for scripts, and portability. Figure 31 shows how Impromptu fits in the tool environment presented in Figure 8.
With this the main quality attributes and functionalities defined in Chapter 5 are satisfied. The next chapters discuss satisfying the requirements for “Finding root causes of exceptional behavior” of Section 5.3.
10 ANOMALY DETECTION DEMONSTRATOR

This demonstrator concerns applying anomaly detection techniques within the context of the company. In particular it shows the potential of applying generic data analysis techniques to the domain, and verifies the defined framework fits the needs. This chapter discusses two of the three use cases shown in Section 5.3: Finding anomalous moments for sensor data and finding anomalous machines in a population.

This demonstrator starts from a business case and is then translated into specific data mining problems. Plugins to solve these problems were developed according to the framework in Chapter 9. Throughout this case study, stakeholders were included in demo sessions to gain feedback on the effectiveness of the methods and for domain expertise. The next sections follow the same structure as the flow shown in Figure 9. Figure 32 shows the generic flow for this experiment. In the practical case, the module concerned is the fixation module and the variables are vacuum pump pressure and time.

**Hypothesis:** It is possible to automatically identify anomalous behavior in a module

1. Load data for machine based on ID, module, relevant variable,
   and time interval of the last year (subsampled)
2. Explore the two weeks around a known anomaly and see what
   pattern occurs. Then select a technique that finds this pattern.
3. Apply the anomaly detection script on the whole (1 year)
   dataset and see if the anomaly is found.
4. Observe the anomalies found for the given machine

**Figure 32 Analysis flow for anomaly detection on vacuum pump failure**

10.1 BUSINESS CASE: AUTOMATIC DETECTION OF AIR TUBE FAILURE

A stakeholder from the machine climate control module presented the following case: After a period of time customers indicate a drastic decrease in print quality. An analysis of the sensor data for climate control has shown a strong decline in vacuum pump pressure. A follow-up inspection at the machine in the field showed a rupture in one of the air tubes controlling the printing drum climate. Given this case, the questions posed by the stakeholder are:

- Can we automatically find the moment at which the air tube fails?
- Can we automatically find other machines that show this behavior?

This demonstrator aims to show a more effective way of analyzing this case and enabling the comparison of machines in the fleet to see if a global problem occurs or a local problem stands out. In terms of monetary value, understanding the behavior of air tube failures can either lead to the use of different air tubes (increasing machine quality, and hence saving service costs), or make their failure predictable (hence saving service costs by making a visit predictable). In addition, this gives printer owners predictable downtime, saving their costs.
First, consider the question of “finding the moment at which an air tube fails automatically.” The stakeholder relayed the following domain-based facts on the analysis throughout demo sessions:

- As an input, use data of the pressure sensor of the vacuum pump of one machine.
- As an output, find the point(s) in time where anomalous behavior of the vacuum pump occurs, where anomalous behavior can be recognized by:
  - A sharp relative change in value given the values in a local context (a timeframe)
  - An absolute change in value exceeding known threshold values (2500 – 3500 pascal for absolute pressure in the pump)
- Only consider moments when the machine is running (i.e., in a state of printing)

The particular signal to be found is shown in Figure 33: The point where pressure drops from around 3000 towards 1500 is interesting.

![Figure 33](image_url)

**Figure 33** Original Signal of Known Failure: Time versus Pressure.

The blue line includes samples from any state, the red line only has samples from a running state.

**10.2 Preprocessing Vacuum Pump Sensor Data for One Machine**

The first step is loading and preprocessing the vacuum pump data to be used. The data relevant for this experiment is time series data of the vacuum pump pressure in Pascal. The data is sampled on average five times per second and is stored in a tabular structure. One year of data for this sensor is around 70 million rows and 40 gigabytes.

As all data is currently in .csv tabular structure files, a generic way to load sensor data is preferred. However, loading all data for the lifetime of a machine is ineffective performance-wise: it takes hours to load into memory (if it fits at all) and it may take a long time to get even a basic overview. It is also not necessary for insight into a general trend or tipping point to have five data samples every second. Hence, subsampling is to be applied to this sensor data.
To allow loading of multiple .csv data files and subsample them, a plugin called “PreloadCSVData” was developed. The input of the plugin is a dictionary with the following parameters:

- **Source**: A string indicating the source folder to load data from
- **Target**: A string indicating the target folder to store data
- **Sparsity**: An integer indicating the rate of subsampling. A sparsity of K implies one in K samples are taken from the data
- **StartTime**, **EndTime**, and **TimeStampCol**: **StartTime** and **EndTime** are datetime objects indicating the start and end period for which data should be loaded. **TimeStampCol** is an integer to identify the timestamp in the source set.
- **ReturnData**: A Boolean to indicate whether data should be stored on disk or returned and kept in memory

The output of the plugin is the subsampled data either in memory or on disk in a temporary storage. Such temporary storage becomes relevant when subsampling takes a relatively long time. In this case, preprocessing takes 15 minutes, while an experiment on this data takes around three to five minutes, making an intermediate storage worthwhile for multiple experiments.

As data is stored in an intermediate storage, there should be a standard way to retrieve it. A plugin called “LoadIntermediateData” was developed. The only input parameter is the path to the stored object. The returned value is the stored object, in this case the subsampled data for the vacuum pump.

Two preprocessing steps are applied. First, the absolute value of the vacuum pressure signal was taken. Dependent on the configuration of the machine and the timespan used, some of the vacuum pressure data is positive, while other is negative (example shown in Figure 34). From a stakeholder interview, I learned that the vacuum pump values cannot become negative and the absolute value can be taken to not detect such a settings switch as an anomaly for machine behavior. Secondly, a plugin called “LastStateFilter” is used to filter the vacuum pump data based on when the printer is running, i.e., to filter red line from the blue line in Figure 33.

![Figure 34 Vacuum Pump Sensor Signal with Inverted Pressure Value](image)

**Figure 34 Vacuum Pump Sensor Signal with Inverted Pressure Value**
10.3 FINDING ANOMALOUS MOMENTS IN VACUUM PUMP BEHAVIOR
The next step is finding the moment when an air tube fails. To make the solution applicable to multiple situations, it is generalized to finding similar anomalous moments, i.e., the moment where there is a significant change from the local values. Note that just filtering out the data between the 2500 and the 3500 interval for a running machine will give a good indication of failures. Anomaly detection is a relevant improvement as it can distinguish a big jump from small peaks outside of the range.

To find anomalies we need an analysis script that takes the vacuum pump data (initially of a short period around an anomaly) as input and provides a set of candidate anomalies as output. At this moment, there is a lack of understanding of both data and domain. So, rather than judging which values are anomalies, it is more effective to present the most significant candidate anomalies to domain experts and let them learn from the output and/or improve the experiment. So, a plugin was defined that, given a dataset of one dimension, provides the user with a list of times at which the data is most anomalous.

In the end two techniques were applied. Both were chosen because they are easy to use and effective enough. Both algorithms loop over all samples in the data set and assign an anomaly score. The times for which the highest anomaly scores occur are reported.

1. A Standard-deviation technique developed by me. For each sample the anomaly score is defined by comparing the sample's standard deviation to a rolling average of a window of samples around it. An anomaly score is assigned to all times based on how many standard deviations the sample is away minus one standard deviation. I.e., all values within one standard deviation of the window are considered regular.
2. A least-squares algorithm based on a technique by (Quinn & Sugiyama, 2013). This approach was valuable because a library was available and the approach is robust to input. An anomaly score between zero and one is assigned to all times.

The following techniques were also briefly investigated:
- The change point method IKA-SST by (Idé, 2014). This attempt was dropped as implementing the technique correctly from a paper with no implementation directions was very time consuming. In addition, the input for this method was deemed too complex to wrap effectively for regular users.
- A local outlier factor implementation was considered, but was dropped due to very low performance (13 seconds per value to check or 402 rows of data with 2 columns)
- Finally, an implementation the scientific method LoOP (Kriegel, Kröger, Schubert, & Zimek, 2009) was a candidate, but was stopped as I could not find a robust way to configure the tool such that absolute changes score higher than relative changes.

Figure 35 is an overview of the steps taken.

![Figure 35 Vacuum Pump Preprocessing Flow](https://github.com/damjankuznar/pylof)

29 Damjan Kuznar - https://github.com/damjankuznar/pylof
After applying an analysis algorithm, it is necessary to visualize the anomalies with the original data to get a view on what behavior could be anomalous. A plugin was developed to take the original dataset and the prioritized list of anomalous times and visualize them. Figure 36 shows an example output. The top image shows the overview of the top 10 anomalous points for the vacuum pump dataset over one year. The (red) vertical lines indicate the anomalous moments.

The bottom images show a zoomed-in figure of the first and second biggest anomalies. The point indicated in the initial problem statement is considered the second biggest anomaly. A new effect was discovered: somewhere in the very first days of logging data for the machine an unexpected behavior occurred.

In general, the stakeholders have expressed that this script is very valuable to them, as it saves them a lot of manual work in generating and analyzing images. This approach can be applied to a multiple sensor values within the printer. At this moment there is no need to restrict the algorithm with the threshold values 2500 and 3500. The next sections show this is just luck.

**Figure 36 Output of Anomalous Time Finder**

The top image shows the overview for the entire dataset; the bottom row shows two zoomed in anomaly points.
10.4 TOWARDS INSIGHT INTO THE WHOLE FLEET

The above results for anomaly detection were shown to stakeholders, who gave positive feedback. However, their second question remains: “Can we automatically find other machines which show this behavior?” to find out whether this is a one-time occurrence or a global problem. The same approach for one machine is now applied to multiple machines. Once again it is impossible at this moment to automatically judge which machine is behaving anomalously and which is not. Hence, a ranking of the machines are presented in order of how anomalous they are perceived to be.

This presents a challenge: How does one compare the anomalous behavior of machines? From the previous experiment, there are two approaches to work with: the rolling standard deviation and the least-squares based approach. Alternatively a solution can be built from scratch. The least squares-based approach cannot be used for comparison as the outcomes always have a similar distribution, both for very regular and very abnormal machines. The anomalousness of the rolling standard deviation is not distributed regularly and hence can be used for comparison.

In this case, the mean over all anomalies for a machine was used for comparison as it suffices to gain an initial overview, and starting from scratch is too time-consuming. This metric is not necessarily a strong indicator of the air tube failure signal, but generally of a jump in relative standard deviation.

Data is not fully comparable. Even though the time period used is the same for all machines and the metric is the same, the number of samples and spread of these samples is not the same. Some machines print more than others and have different timing for printing. See Figure 37 for some examples. This aspect was not focused on for this experiment, as significant patterns were picked up correctly.

![Figure 37 Difference in data sample spreads](image)

**FIGURE 37** DIFFERENCE IN DATA SAMPLE SPREADS

**The left dataset only has samples in the start of April (around 1000), while the right dataset has evenly distributed samples (around 20000) and behaves regularly**

There is need for using thresholds to indicate safe zones (the pressure values 2500 until 3500). Some machines were initially ranked as highly anomalous because they had a high difference in standard deviation (See Figure 38). However, all their values were neatly between 2800 and 3000. It is not possible to remove these values in a preprocessing step as they are the regular behavior to find anomalies in. Hence, the anomaly detection algorithm was updated to allow for a filter input. In practice this filter is a function given as a parameter to the anomaly detection plugin. The filter indicates all values between 2500 and 3500 are considered exempt from being an anomaly.

67
**Figure 38** "Anomalies" that are all within the safe threshold.

These values are not interesting for this analysis.

**Figure 39** Flow for ranking machines by anomaly score

In the end, the mean of the standard-deviation-based anomaly scores are used to compare machines. The values within the threshold are ignored. The total flow is as seen in Figure 39. Preprocessing of data is the same as for a single machine.

As an outcome, we get ranked visualizations of the anomalies for the most anomalous machines as seen in Figure 40. We found three main kinds of patterns: The top left image shows an effect attributable to installing the machine. These machines are hard to compare as they do not have the same amount and spread of samples. I.e., the right figure has 30000 samples, the middle picture has around 25000 samples, while the left picture has around 5000 samples and is sparse.

Despite an imperfect ranking, stakeholders still indicate this is a valuable result. Previously, it was necessary to either write a complex script or manually run the same analysis for every machine to gain this insight. The second image shows a change in settings or fix applied to a machine. The third shows the behavior we initially started from: A failure in the air tube leading to lower vacuum pump pressure.

**Figure 40** Output of anomalous machine finder

All images show the overview of anomalies within the machine. The machines are in order of mean anomaly score.
10.5 WRAP UP AND LESSONS LEARNED

To conclude, anomalous behavior in the vacuum pump of the print drum can be automatically found, in addition to other anomalous effects. Through feedback from the stakeholders and the development process, the following lessons were learned.

First, automatic scripts for preprocessing and visualizing are seen as very valuable by the stakeholders, regardless of the analysis techniques used in between. To automatically generate a list of graphs for a particular sensor (e.g., vacuum pump sensor data) and visualize the behavior was already a big step compared to manual analysis. At this point stakeholders indicated that scrolling through one picture per machine is an acceptable situation. As the number of machines in the field grows, ranking of anomalies is expected to become more important.

Second, this solution does not completely solve the business problem. Although testing the hypothesis of “quality has decreased due to the print drum climate” is now possible, there is no functionality to automatically find out if the vacuum pump data could be relevant to a decrease in print quality.

Next, this analysis reinforces the need for sharing domain knowledge. Once a domain expert has seen and assessed the pictures in Figure 40, it is very valuable to annotate an air tube failure, a setting change, or a startup effect occurred at a certain time. In a follow-up experiment, this observation can become obvious without analysis.

Finally, and most importantly: There is always need for domain knowledge to perform an analysis on the machine sensor data. To make a hypothesis for analysis, one needs to understand the relations between printer modules, e.g., understand which climate sensors could have data relevant to print quality decrease. Domain expertise is also needed to interpret the changes in signal values, e.g., to understand that a dip in vacuum pump could imply a broken air tube, and to understand that a short spike could mean a startup effect.
11 CORRELATION DEMONSTRATOR

According to stakeholder interviews and brainstorm sessions, the second most promising topic is relating datasets to find root causes via correlation analysis. The goal is finding the relations between signals and behavior of models in the machine as seen in the flow of Figure 41. Correlation techniques are taken as the scientific background for finding relations. With this chapter the final use case of Section 5.3 is covered.

**FIGURE 41 ANALYSIS FLOW FOR CORRELATION BETWEEN FUNCTIONS**

11.1 CORRELATION WITHIN A MACHINE MODULE

The starting point for correlation was a standard Pearson correlation technique, which is part of the Pandas Python library. At first, the basic premise of this technique needed to be verified with the company data. Two types of experiments were executed: One for data within the same machine module and one for data between functions. Correlations between machines are not considered, as current machines are known to be independent.

For the first experiment, data from the printer climate control (also used for the anomaly detection case) was used. Climate data includes humidity, temperature, and pressure for various modules of the machine. Data was loaded and preprocessed as described in Section 10.2. To get a fair comparison between signals in the module it is necessary to normalize the data. A plugin called “DataFrameNormalizer” was created, which, given a DataFrame and the name of the time variable, normalizes all variables in the DataFrame, except time. The plugin uses a Python normalize function from the Scikit-learn preprocessing library. Figure 42 shows an example.

**FIGURE 42 EXAMPLE OF NORMALIZATION OF MACHINE MODULE SENSOR SIGNAL**

The left figure shows the raw signals, the right figure shows the normalized signals.
Next, a standard Pandas library correlation method was used to generate a matrix of pairwise correlations. The results of this method were visualized using a matrix where correlation strength is related to a color map. Figure 43 shows an example for the climate control module.

**Figure 43** Correlation matrix for climate control module

The colors indicate the strength of correlation. Red (1) implies a positive correlation, where blue (-1) indicates a negative correlation. Grey (0) implies no correlation.

For a domain expert, not all correlations in Figure 43 will be interesting. For example a strong correlation between `actCIR1[0.1%]` and `actCIR2[0.1%]`, which are two sensors near one another, does not give a new insight. However, the user would still like to see the strongest correlations without needing to check every pair of variables. For this reason the Pandas correlation method was wrapped in a plugin called “PairwiseCorrelation”.

The input of the “PairwiseCorrelation” plugin consists of a dataset, the number of correlations to show, and a list of variable pairs that can be ignored. For example [“LocalTime[us]”,“*”] to ignore any correlation with time. It returns a list ordered by the absolute correlation value, excluding any of the pairs to be ignored. The output is a list of Dictionaries as the following text:
As this list is not very intuitive, a visualizer called "CorrelationsViewer" was created. The visualizer needs the following input:

- The list of correlations in the format seen above
- The raw data (before normalization) for all variables
- The normalized data for all variables
- The data to be used for the x-axis, in most cases the time

The plugin outputs the top correlations as signal graphs with the corresponding correlation values. The signal graphs have both the normalized and the original signal. Figure 44 shows an example of such an output.

![Figure 44 List of top correlations in dataset](image-url)
From this experiment, the following was concluded:

- The proposed framework can support multiple types of data analysis.
- The correlation method is applicable on some of the data in this context. However, the relations found do not provide new insight in the machine behavior.
- Not all related signals can be related by taking the raw datasets. Preprocessing is needed for fair comparison.
- Many strong correlations do not give new insight in machine behavior as they are obvious before analysis, e.g., a relation between a heater and a heat sensor.
- Reuse of the Python analysis libraries is effective.

11.2 CORRELATION BETWEEN MACHINE MODULES

Correlations within a machine module are often known by domain experts, as they work together based on the machine modules. Relations between the modules are not as obvious. The next step was to find out if correlations occur between machine modules.

The data was loaded as described in Section 10.2. The data was subsampled, using one in every few thousand values to reduce computation time. All datasets were normalized using the normalization plugin presented in the previous section. This normalization plugin was extended to interpolate values based on one set of timestamps. This was done to ensure that the signals are all aligned on the x-axis. Linear interpolation was used, which should suffice given the large amount of samples.

The correlation plugin of Section 11.1 was used to find all pairwise correlations. This dataset imposed a more complex requirement on the correlations to be ignored, namely: Only correlations between modules should stay. To facilitate this, all dataset variables were prepended with an identifier string. All pairs with the same identifier string were ignored. In addition, all values of time, sheet identifier, and other meta-data were excluded, as they are uninteresting and occur often between modules.

In the end, none of the signals in different modules show a strong correlation. The strongest correlation can be seen in Figure 45 and is around 0.166. No conclusions about relations can be drawn. Proper preprocessing might expose a correlation, but to obtain the knowledge to properly preprocess all data for a fair comparison is infeasible given the spread of domain knowledge. This reaffirms the need to share domain knowledge, also between designers of different modules.

![Figure 45 Highest Correlation between Climate and Paper Deformation Data](image-url)
The following is concluded after this experiment:

- Even with subsampling, it is cumbersome and slow to load all data for different modules and link this data. Good data loading and preprocessing plugins are needed, but currently out of scope.
- Domain knowledge is necessary in multiple steps of the analysis:
  - For preprocessing the data correctly
  - For excluding uninteresting pairs of variables
  - For assessing the output correlations and defining a follow-up experiment

When the results of this exploration phase were shown to stakeholders, it was confirmed that signals need to be preprocessed to truly assess correlation. Even with powerful analysis machines and tools, domain knowledge is necessary for this preprocessing steps. For example, it is impossible to correlate paper rejects with high humidity when taking raw signals. Knowing the signals, one could preprocess data to gain time-based averages of the paper rejects and peak values of humidity going over a threshold.

### 11.3 BUSINESS CASE: EXPLORATORY ANALYSIS FOR REJECTS

Printers need to remove media sheets when sheets deform. The machine in this context automatically rejects sheets based on a threshold, also known as the “rejectLevel." These sheets are rejected because keeping them in the system leads to paper jams and potential damage to print heads. Sheet deformation is measured in the x, y, and z directions, and for every sheet a combined metric of these deformations (HighestDefectHeight) is used to decide whether a sheet should be rejected or not. As rejected sheets mean waste, it is important to prevent this.

The question was: What is the root cause of the deformation of the paper? The initial hypothesis was: The climate of the paper input module influences the deformation. The questions posed were:

- How can we find which media type produces the most waste?
- Are there any obvious correlations of signals with the amount of waste?

Where the previous case started from a known type of signal to be detected, this case started from exploration. There is no interesting known relation to be verified. Four experiments were performed in an attempt to identify an unknown relation.

The first experiment was a black box experiment. Data from a multitude of climate logs and the rejects log are used as an input. Once again, the same preloading plugin is used. Data is sampled such that around one sample of every five minutes is taken. Data within a period of six months is considered. Loading this data takes around three hours, which is too slow for exploration.

In addition, an inconsistency in the log files was found. For the paper input module log data, the timestamps are taken as microseconds since the 1st of January 2016, while all other datasets have microseconds since the 1st of January 1970. This inconsistency forced me to update the “PreloadCSVData” plugin to account for it. I strongly suggest ensuring consistency in these logs, to prevent extra difficulties when loading and comparing datasets.
After loading the data, we tried to filter out data corresponding to one media type, to see if there is a common behavior. As joining large datasets together is a slow process, a plugin called “LastStateFilter” was developed. This plugin has as input:

- A dataset that is to be filtered, which is called `setToFilter`
- A dataset that consists of data based on which a filter can be applied, which is called `filterSet`
- A filter function to indicate which data should be filtered
- A name for the column to merge the two datasets on

The plugin creates a log from the `filterSet` dataset based on the filter function. The log only consists of the change points. It is expected that regular users can create such filter functions based on their domain knowledge. An example `filterSet` and filter function with corresponding are:

<table>
<thead>
<tr>
<th>SheetID#</th>
<th>MediaName$</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>PaperType1</td>
</tr>
<tr>
<td>13</td>
<td>PaperType3</td>
</tr>
<tr>
<td>14</td>
<td>PaperType2</td>
</tr>
<tr>
<td>15</td>
<td>PaperType1</td>
</tr>
</tbody>
</table>

`filterFunction`: return `filterSet["MediaName$"]` == "PaperType1"

<table>
<thead>
<tr>
<th>MergeValue</th>
<th>KeepValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>True</td>
</tr>
<tr>
<td>13</td>
<td>False</td>
</tr>
<tr>
<td>15</td>
<td>True</td>
</tr>
</tbody>
</table>

Then, the plugin filters the `setToFilter` with the log based on the latest `KeepValue`. Given the current example, that means:

<table>
<thead>
<tr>
<th>SheetID#</th>
<th>HighestDefectHeight</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>0.5</td>
</tr>
<tr>
<td>13</td>
<td>0.1</td>
</tr>
<tr>
<td>14</td>
<td>0.2</td>
</tr>
<tr>
<td>15</td>
<td>0.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SheetID#</th>
<th>HighestDefectHeight</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>0.5</td>
</tr>
<tr>
<td>15</td>
<td>0.4</td>
</tr>
</tbody>
</table>

This two-step approach prevents a potentially quadratic size join of datasets. After extracting the data for the correct media type, the approach described in 11.2 was used. Both correlation matrix and list of correlations did not show any new insights. This experiment was not pursued further as stakeholders indicated it is not sufficient to take a black box view on the media types.

This led to follow-up experiments, where a stakeholder cooperated in some sessions to provide domain knowledge. Between sessions I set up experiments to confront the stakeholder with. The stakeholder indicated there might be a link between paper input module climate and the amount of rejects. He suggested finding out which media type has many rejects, where many was defined by him as over two percent.
For this experiment three datasets were used: that of sheet deformation, that of media type, and that of the paper input module climate. Sheet deformation and media type information was linked using a different plugin, as data for all media types was needed. A plugin called "IntervalLinker" was developed. This plugin essentially performs a join operation on two datasets, as one could do in an SQL database or with Pandas DataFrame's. The difference is that the timestamps cannot be matched as signals are logged independently. Instead, data is matches depending on the latest timestamp.

Through stakeholder discussion it became obvious that this approach does not suffice. The "IntervalLinker" creates duplicates of certain paper sheets, which skews results. In addition, the subsampling used for loading data skews data as well. Hence, this experiment was stopped. A new experiment was started using all data to prevent this skewing effect and a better focus.

11.4 BUSINESS CASE: HIGH WASTE IN LAB MODEL PRINTER
The previous experiments had an exploratory nature and did not provide any conclusive results. To make the case more concrete, a stakeholder indicated that a relatively large number of rejects occurred particularly with duplex jobs within a specific lab machine. The assumption is still that a paper input module could be the reason. The main questions are:

- Is there a particular paper input module corresponding to the rejected sheets?
- Can we detect if there are relations between the paper input climate and the number of rejects? How about other paper input module behavior?

Like the previous experiments, sheet deformation, media type, and paper input module climate datasets are used. Instead of subsampling data, a more involved filter was needed to efficiently load all data concerning media type. The plugin "PreloadFilterCSVData" was developed to allow more interesting filters during preprocessing. A list of filters can be input to specify whether a row of tabular data should be loaded. In this case the filter was that the “plexity” field should indicate duplex.

After loading media type data, a list of media types with duplex jobs is obtained. This list is used as an input to load sheet deformation data. A filter based on these media types is used for the "PreloadFilterCSVData" plugin. When trying to link the media type data with sheet deformation data, a memory limit was reached, blocking progress. An experiment with a smaller dataset is described below. Obtaining a valid indication of reject rates over a longer time period within a week is infeasible with current tools. At this point in the project, there was not enough time to develop parallel processing approaches or wait for more effective data loading scripts. Hence, this experiment was stopped as well.

In the case that this data loading obstacle is overcome, for example with out-of-memory processing, finding out the most commonly rejected media type is straightforward. One can use standard grouping and counting functions from the Pandas Python library to show which percentage of sheets is rejected for each media type.

Finally, an experiment on the same data for a two week interval was performed to explore the potential of correlating a reject signal. In this case climate data was correlated with reject data. Reject data is in the form of a "yes" or "no" string for the Rejected field. To get a time-based signal from this data, the sum of "yes" values was taken. These sums were then for periods of one hour. An example of the reject signal can be seen in Figure 46. The correlations found with this signal can be seen in Figure 47. The "Sum of Rejects" row shows reject correlation.
Even though the basic correlation technique works, no strong correlation between one of the sensor signals and the time-binned rejects is found; the strongest is 0.166, where 0.667 would
be significant. There could be no signal that correlates with rejects, but no conclusion can be drawn. The following factors prevent a conclusion to be drawn:

- The data with a small time interval cannot be used to find the longer term effect.
- The sensor signals might not have been preprocessed properly; Domain knowledge is needed to verify this.
- The data necessary to find a signal with rejects is not logged at all

11.5 WRAP UP AND LESSONS LEARNED
This experiment does not provide a new insight, but confirms the existing insights of intramodule relations. It successfully shows that:

- The framework structure can support multiple types of analysis.
- Third party correlation techniques can be applied in this context. In general, I recommend wrapping existing analysis algorithms in the framework plugin structure when possible.
- Domain knowledge is necessary for analysis.

Throughout these experiments a number of limitations in both the current status quo at Océ and the techniques created by me became apparent.

First, consider the data. There is not enough consistency in the data. An example is the timestamp of the paper input module differing from the timestamp of other datasets. Another example is that, even though datatypes are indicated, they do not follow the same semantics. In addition, linking data can be cumbersome. For example, to link a particular paper sheet to the job this sheet is part of, there are two identifiers that must be matched in different files. I recommend storing the relations between jobs and sheets explicitly.

Then, consider the file infrastructure. With the current file format and file loading tools it is impossible to efficiently link two large datasets. I recommend looking into these technologies and integrating one of them in ODAS. In addition, the current setup allows for analysis of big datasets in memory, but does not allow for saving big intermediate results. Any database or advanced file storage solution, such as HDF5\(^\text{30}\) or HDFS\(^\text{31}\), would solve many efficiency, data saving, and data loading problems. At the moment, ODAS has prototyped an HDF5-based solution and plans to engineer and deploy this. Plugins such as “LastStateFilter” and “PreloadFilterCSVData” are not necessary when such a file structure is used.

Next, consider the available tools. Regardless of file infrastructure, it should be easy to load and preprocess data. Currently there is no effective way to filter while loading data, such that less processing power and memory are wasted. Developing flexible loading and preprocessing plugins is needed to enable effective analysis. Even with efficient programming and good preprocessing plugins, out-of-memory techniques need to be supported when memory limits are reached. At the moment of writing, ODAS plans to provide these.

Finally, consider domain knowledge. No out-of-the-box correlation technique can find correlations without preprocessing the signals. In addition, correlation does not imply causation. Even if a correlation is found between two signals in the machine, a follow-up experiment by a domain expert is needed to verify this relation.

\(^{30}\) https://www.hdfgroup.org/HDF5/
\(^{31}\) https://hadoop.apache.org/docs/r1.2.1/hdfs_user_guide.html
12 CURRENT IMPACT AND RECOMMENDED EXPANSION

This chapter discusses the impact of the solution in the company, gives a recap of lessons learned, and provides the main recommendations to improve effective analysis.

12.1 IMPACT IN THE COMPANY

Throughout this project stakeholders have been kept in the loop to get feedback. This feedback concerned the usefulness of the proposed solutions, getting domain expertise as input, and obtaining business cases. This section describes the indicators of having a positive impact in the company and serves as a validation of the focus of this project.

First and foremost, machine module designers consistently were willing to schedule meetings with me for demo’s, feedback, or even paired analysis. Even though the schedules of machine module designers are very busy, they indicated they see the value of my approach and were willing to consider and give feedback on proposed working methods.

Secondly, the machine module designers notified me and asked help with current analysis cases. In total, five relevant cases were presented to me. In addition, they made time to explain domain-related facts and walk through an analysis with me.

Next, stakeholder feedback was positive on the usefulness of this project. I got positive feedback on the capabilities to 1) compare between machines in a fleet, 2) automatically point out the anomalous points in data, and 3) the capability of storing and retrieving domain knowledge with data.

There was no opportunity to do a full analysis case, such as correlation between climate and paper rejects, together with a machine module designer in this framework or a user test with multiple users. Hence, the effectiveness of the Impromptu results is not measurable until it can be deployed for a larger audience. In addition, there was no way to measure monetary gains. The biggest potential impact in the long term is an earlier release of the product. There are too many factors influencing release of a product to objectively state the impact of this project.

Finally, the main designer and developer of ODAS kept in touch to allow integration. Anomaly detection and sharing domain knowledge were confirmed to be valuable additions. In the end, anomaly detection, domain knowledge sharing, and the basic plugin structure were transferred to expectedly become part of ODAS.

With the feedback from the stakeholders and the implementation and deployment of design, it also became clear what the impact is in terms of design criteria defined in Section 5.5. First, let’s consider usability for the three user groups: basic, regular, and advanced (See Section 5.1 for reference). The (blue) top arrows show the planned improvement and the bottom (green) arrows show the perceived progress.
Currently, basic users benefit from the all-inclusive analysis scripts (in the form of Notebooks) provided by ODAS. Stakeholders have confirmed that adapting simple settings, such as machine ID, time period, or sensor value to look at are fine in the Jupyter environment. This project has only indirectly helped basic users. It provides help to regular users, which in turn create scripts for basic users. However, no step was made in helping the script development process of basic users when comparing to Diagnostic Framework and ODAS.

**Table 12 Success for usability for basic users**

<table>
<thead>
<tr>
<th>Rating</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Basic users cannot get analysis results at all.</td>
<td>Service product specialists have no way to get analysis results</td>
</tr>
<tr>
<td>2</td>
<td>Basic users need regular / advanced user to analyze.</td>
<td>Service product specialists ask machine module designer for analysis every time.</td>
</tr>
<tr>
<td>3</td>
<td>Basic users need to spend a long time developing analysis.</td>
<td>Service product specialists spend over a day to implement simple analysis and run it.</td>
</tr>
<tr>
<td>4</td>
<td>Basic users can start an analysis with some configuration and gain results.</td>
<td>Service product specialists spend a few hours configuring an analysis script and gets results.</td>
</tr>
<tr>
<td>5</td>
<td>Basic users can start an analysis with minimal configuration and gain results.</td>
<td>Service product specialists can run a pre-made analysis script for their case and get interesting insight.</td>
</tr>
</tbody>
</table>

Regular users benefit from the generic analysis scripts in the form of plugins. The standard interface is deemed acceptable and the current overview of scripts is sufficient to find and use these generic scripts. In addition, the Jupyter technology offers enough flexibility, whereas example Notebooks (e.g., air tube failures) and generic scripts (e.g., anomaly detection and visualization) show the feasibility of creating and running a full analysis. There is, however, not enough proof to conclude that machine module designers can effectively develop own filters and enhancements for these scripts.

**Table 13 Success for usability for regular users**

<table>
<thead>
<tr>
<th>Rating</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Regular users cannot get analysis results at all.</td>
<td>Machine module designers have no way to analyze what they want with current tool (e.g., DF).</td>
</tr>
<tr>
<td>2</td>
<td>Regular users need advanced user to analyze.</td>
<td>Machine module designers ask expert in their team to do the analysis or help them on the way with an algorithm.</td>
</tr>
<tr>
<td>3</td>
<td>Regular users can find and use analysis scripts and tweak them to their needs.</td>
<td>Machine module designers ask expert once for an algorithm and run own analysis afterwards.</td>
</tr>
<tr>
<td>4</td>
<td>Regular users can extend analysis script with own filters and enhance analysis.</td>
<td>Machine module designers find scripts and extends them with a domain-based filter(^\text{32}).</td>
</tr>
<tr>
<td>5</td>
<td>Regular users can easily customize and configure existing scripts to do analysis.</td>
<td>Machine module designers find scripts and extends them with existing material based on domain knowledge.</td>
</tr>
</tbody>
</table>

For advanced users, the use of Python and Jupyter supports creating Python scripts and plugging them into the Impromptu and ODAS framework. At this point deployment consists of conforming to the framework structure (See Section 9.1) and copying the script into the plugins folder. No automatic deployment or wrapping for plugins exists at this moment. For users of other languages than Python (e.g., Matlab), the rating is still considered three, as there are no example scripts for wrapping code and transferring data effectively.

\(^{32}\) Example filter: Consider only the values within the known signal threshold.
Table 14 Success for usability for advanced users

<table>
<thead>
<tr>
<th>Rating</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Advanced users cannot perform analysis given the current tools and data available.</td>
<td>Machine module designers can only use DF or does not have the necessary data.</td>
</tr>
<tr>
<td>2</td>
<td>Advanced users can work with existing tools and is limited by them.</td>
<td>Machine module designers can only use existing tools to create (generic) analysis scripts, e.g., only DF is available.</td>
</tr>
<tr>
<td>3</td>
<td>Advanced users can work in preferred environment, but needs significant effort to run analysis.</td>
<td>Machine module designers can work in their preferred IDE, but need to manually upload and configure library function to use on server.</td>
</tr>
<tr>
<td>4</td>
<td>Advanced users can work in preferred environment and deploy and run an analysis.</td>
<td>Machine module designers can work in their preferred IDE, and can plug in library function to use on server.</td>
</tr>
<tr>
<td>5</td>
<td>Advanced users can work in preferred environment and instantly deploy and run analysis.</td>
<td>Machine module designers develop generic and specific scripts in preferred IDE, then instantly publishes and runs on analysis server.</td>
</tr>
</tbody>
</table>

Table 15 shows the impact of this project on the reusability design criterion for generic scripts. Through the generic script overview, the criterion for reusability has been satisfied well. However, search functions for scripts should be added when the number of users grows.

Table 15 Success for reusability of generic scripts

<table>
<thead>
<tr>
<th>Rating</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Regular user can only use local files</td>
<td>A machine module designer works only on local machine</td>
</tr>
<tr>
<td>2</td>
<td>Regular user can access reusable scripts, but cannot easily find them</td>
<td>An advanced user creates a reusable data loading script, but machine module designer does not realize it exists</td>
</tr>
<tr>
<td>3</td>
<td>When advanced users create generic scripts, regular users should be able to find these scripts based on manual search</td>
<td>A machine module designer creates an anomaly detection script; another can scroll through existing scripts and find the applicable script</td>
</tr>
<tr>
<td>4</td>
<td>When advanced users create generic scripts, regular users should be able to automatically search for them</td>
<td>A machine module designers created an anomaly detection script; another can automatically find it based on “anomaly detection” name</td>
</tr>
<tr>
<td>5</td>
<td>When advanced users create generic scripts, regular users should be able to automatically search for aspects of them</td>
<td>A machine module designer creates an anomaly detection script; another can automatically find it based on “root cause analysis” keyword</td>
</tr>
</tbody>
</table>

To summarize this section, this project has had a positive impact according to stakeholder feedback and involvement. In addition, the design criteria for usability and reusability are deemed sufficiently covered. However, a larger scale user experiment is needed to measure the effectiveness of the approach.
12.2 LESSONS LEARNED

This section summarizes the lessons learned throughout development of the framework and performing experiments. These findings are generally expected, but were confirmed in the specific company context.

First of all, domain knowledge is paramount in analysis of machine behavior, especially within the high tech domain. In the current situation, it is necessary in the following contexts:

- When exploring the current log data, domain knowledge is needed to understand which part of the logging relates to which module of the machine. One also needs to have basic understanding of the machine module and the meaning of log variables to explore data.
- Domain knowledge is needed for preprocessing of signal data. For example, the scores for paper deformation leading to rejects need to be averaged over a window and filtered for a particular paper type and a type of job (duplex) to have a useful set for analysis.
- Interpretation of analysis results (and raw signals) requires domain knowledge. A peak in the heater signal found by anomaly detection could mean a regular startup effect or a defect. Only a domain expert can tell the difference.

Following from this: the analysis solution in the form of algorithm, library, or tool to be used is not the most impactful decision. The connection of an analysis solution to the domain experts and the potential to translate domain expertise into input and output for such a solution is much more important, i.e.: From my observations during this project, I think it is more fruitful to enable domain experts to use data analysis tools than to let data science experts explore the whole domain.

On the other side, I observed potential improvement of data analysis knowledge and tools for domain experts. Generic data analysis tools and techniques should be offered to domain experts. In this project the advanced user (See Section 5.1) fulfills part of this task using the generic plugins provided by Impromptu. I see a data analysis expert having a supporting role by 1) providing generic data analysis techniques, 2) performing a system-wide analysis with a group of domain experts, and 3) selecting the best fitting technique for the domain-specific problem.

A complex process might be needed to gain insight into the behavior leading towards a defect, but his does not mean an equally complex process is needed to predict this defect afterwards. In a particular case of analyzing LED’s the process of finding the root cause signal is complex, but the prediction of an error can be done by checking if values exceed a threshold. In the company context, using thresholds specified at design and the sensor values for these thresholds is a good way to find abnormalities, both within a machine and between machines.

Furthermore, preprocessing data and exploratory data analysis are as important as reusable analysis scripts for varying tasks. Examples of very simple, yet valuable approaches: To combine data over machines in the fleet, to filter data based on a media type or time, or to find the biggest anomalies in a sensor signal. In addition, visualization for understanding signals and exploring data is very valuable. In these situations, the same technology can be reused not only with similar tasks, but also with different types of analysis tasks.

Finally, a large number of readily available scientific analysis, preprocessing, and data manipulation techniques can be applied. Within Jupyter, prominent examples are Scipy, Numpy, Pandas and Scikit-learn. These libraries have a lot of potentially reusable functions which are easy to miss.
12.3 RECOMMENDATIONS
The recommendations are split up in two categories: Those confirming the strategy of ODAS, and those that propose an addition to ODAS. The recommendations are based on the experiences gained in doing analysis, developing the framework, and interviews with stakeholders.

12.3.1 FOLLOW THE GOOD PRACTICES PROPOSED IN ODAS AND IMPROMPTU
First, consider the recommendations matching with the ODAS plans:

- A lot of redundant work is done in redesigning and implementing data loading, preprocessing, automation (e.g., look over the fleet of machines), and visualization functions.
  - These functions take upwards from 50% of the total analysis time, and hence are worth making reusable. In addition, the development of a normalization, data linking, and data loading script has cost over a day respectively. To reuse these scripts only takes half an hour. Hence, to provide a centrally accessible set of such scripts saves a lot of time.
  - This recommendation matches with the plans for ODAS to introduce a standard notebook format and providing standard functions.
- I confirm it is wise to change to a more robust and flexible file format for analysis. This matches with the plans and prototype of HDF5 for ODAS.
  - Consider a query language to extract data (e.g., select fixation/airheat/controller data from machines 103,104,105 in the last five weeks).
  - Allowing domain knowledge to be stored as meta-data with data prevents redundant experiments. E.g., enabling to indicate a startup effect prevents redoing an experiment for this effect.
  - Making the log format standardized and robust to column changes. Current analysis is made difficult due to columns being inserted at unknown times in the same type of data sets. However, columns change due to the dynamic environment of analysis; it is unpredictable which data will be required in the future. In this instance, the .csv format is too free.
- Consistent with ODAS, I recommend maintaining and governing the overview and structures of both logs and scripts. In practice, I propose 1) to look for someone willing to be responsible for governing and keeping overview of the logs and scripts and 2) to maintain the scripts and log design in a versioned storage.

12.3.2 INVEST IN DATA OVERVIEW, REUSABLE SCRIPTS, AND KNOWLEDGE SHARING
Next, consider the proposals as an addition to the current plans. I recommend designing the log overview as a platform-wide initiative. Lack of this overview leads to a week of data exploration for every new inter-module analysis.

The benefits of such a design are:

- No mismatches between timestamps and other identifiers.
- Safeguard completeness of a log from a diagnostics standpoint, i.e., to be able to answer questions on data from the whole machine, instead of one module.
- Have an overview of the log, to enable data scientists and machine diagnostics analysis to start up faster, on top of module-based analysis.
The aspects for a project-wide log I recommend are:

- Uniqueness of identifiers and start/end moments for common concepts, such as jobs and sheets. In addition, an easy ways to link data based on these identifiers, e.g., by having job id matched with sheet id by default.
- Enforce the same semantics for the already standardized data formats. E.g., a Boolean value is always True or False, not yes or 0.
- Share the domain knowledge of all modules and variables and store it in a central place, e.g., by providing a meta-data description in a database.

I also recommend sharing domain knowledge between functions to enable inter-function analysis, such as a correlation analysis to find the root cause for paper rejects. Given that domain knowledge is stored with the data, I recommend implanting functionality be possible to filter and search based on this domain knowledge to prevent redundant work, e.g., to filter data to exclude the startup effects or to alert on data exceeding thresholds.

Next, I recommend investing in the creation of generic analysis technique scripts at the start of machine development projects. I spent around 40 hours experimenting with anomaly detection and correlation techniques; repeating this step is wasteful. In addition, good generic scripts enable creation of high quality domain-specific scripts during development. When a product is mature, the domain-specific scripts can then be offered to service product specialists such that remote help is more likely to succeed, saving service costs.

12.3.3 Quantify the Value

Finally, I recommend quantifying the value of this approach, and data analysis in general, explicitly. Stakeholders have indicated that there is a clear value in both time and money.

Examples for data analysis in general, with Diagnostic Framework and remote service include:

- Saving an expected six hours and estimated costs around multiple thousands Euro by finding a disconnected plug; Alternatively, a cooling module would have been swapped.
- Saving expectedly one day by applying remote service while looking at functional logs when solving a conditioning incident.
- Saving expectedly four to eight hours and a lot of costs by finding a software fault through log analysis; the alternative would be to experimentally swap all parts.

The value of the previous examples was not explicitly measured. As the data analysis initiative scales up in both machines in the field and complexity in development, costs for hardware and human resources will increase. Generally, people highly expect this investment to pay off, but to get support it is necessary to quantify it.

I recommend quantifying the value of a predictive maintenance case, as opposed to preventive or reactive maintenance. For such an experiment one would need to consider a machine part showing wear and tear within multiple machines. Then use the current analysis tools to verify if this defect can be predicted. Depending on the prediction, the service schedule can be changed to either 1) replace a part on a planned service visit, as it is expected to break soon and replacing it now is cheaper, or 2) to plan an extra service visit to prevent high costs of a predicted defect. The costs of a service visit must be balanced against the predicted costs of system failures.
In addition, I recommend verifying the effectiveness of ODAS, including Impromptu. I recommend piloting ODAS using multiple people of multiple machine module designer teams:

- The time saved by reusing generic scripts
- The time and money saved by remotely finding a problem's root cause
- The time and money saved by automatically running this analysis on the whole fleet
CONCLUSION

Customers of Océ consider printers to be capital goods and highly value print quality and reliability. To ensure quality and reliability, it is important to understand printer behavior very well. Currently, Océ employs a strategy where machines are released to a set of early adopters before general market availability. Within this early adopter stage, it is very valuable to quickly learn from the machines in the field. Insight into the machine behavior enables improving machine design and predicting defects more accurately, which improves product and service quality. In addition, it helps solve potential issues of these less mature devices faster for the early adopter customers.

Currently, machine behavior is logged in datasets per machine. These logs are used by the Research and Development, Service, and Manufacturing and Logistics departments of Océ. Insight and machine maturity can be gained by having a more effective analysis approach. The approach of machine module designers and service product specialists was focused on, because these stakeholders have the necessary domain knowledge to do an analysis and the influence to improve design and service.

Current analysis tools and data formats should be improved to increase the analysis potential. In particular, this project aimed to provide for machine module designers and service product specialists. The main improvement areas are 1) enabling easy access to a flexible analysis environment, 2) enabling more effective re-use of both generic and domain-specific analysis scripts for various analysis tasks, and 3) enabling sharing of data-related domain knowledge.

The Impromptu project results in a flexible analysis framework to improve the effectiveness of analysis. Impromptu shows a prototype for an easy-to-use environment for analysts to re-use analysis scripts and share domain knowledge. This was achieved through a plugin architecture with a common interface, an analysis scripts overview, and a meta-data storage for domain knowledge that is linked to the original data logs.

Within this project, two business cases were explored to show framework value and the applicability of generic data analysis techniques. A generic anomaly detection technique applied to the signal of a vacuum pump has shown the ability to automatically detect failures and other non-regular behavior. In addition, a correlation experiments has shown the need for effective file storage, flexible loading and preprocessing methods, and domain knowledge.

Usability and reusability are stimulated through easy-to-use analysis scripts in a framework structure, in addition to the benefits of the ODAS analysis framework and underlying Jupyter technology. An agile feedback-based approach has been used throughout the project to keep stakeholders involved and to validate product design. The main conclusions and recommendations for this project are:

- Developing and applying domain knowledge and data analysis knowledge in conjunction. Domain knowledge should be shared among machine module designers. I recommend keeping the threshold for sharing domain knowledge as low as possible. Data analysis expertise should be increased for machine module designers and service product specialists. Alternatively, an expert in applying data analysis could cooperate with these parties.
- Facilitating machine module designers with scientific analysis and automation scripts. I recommend focusing on developing generic scripts for comparison and automation, in addition to encapsulating scientific analysis techniques. This is expected to significantly improve the effectiveness of analysis.

- Defining a structure for functional logging from a platform-wide perspective and providing standard data retrieval functions. The largest part of time for analysis is spent in loading and preprocessing data. A platform-wide log definition prevents mismatches in inter-module analysis and saves time by preventing the creation of redundant loading and preprocessing scripts. I recommend:
  - Defining consistent, platform-wide functional log formats and structures as part of designing new products
  - Storing machine behavior logs in a format that allows adding meta-data (planned in ODAS framework)
  - Providing an abstraction layer towards analysts for querying and storing log data (planned in ODAS framework)

- Quantify the value of data analysis in the company context. At this point a significant investment in people and hardware must be made to set up further data analysis. To get the necessary support, I recommend investigating the gain in time and/or money by: 1) performing a predictive maintenance pilot, 2) defining a metric for the time and/or money saved by applying log data consistency, reusable analysis scripts, and domain knowledge sharing.
14 PROJECT MANAGEMENT

This project has a number of attributes which influence the process and design decisions. The main attributes are:

- A time limitation of nine man months and one resource, equaling ¾ FTE
- An influence space limited to R&D and service (HQ and R&D)
- A lack of initial domain knowledge.

The available time limits the features that can be developed before the project ends. The main challenge of having one resource executing the process within this time is a lack of broad perspective and a lack of domain knowledge. A natural solution is obtaining feedback. Hence, an agile approach was chosen with a focus on getting feedback to validate concepts. The main people giving feedback are the project supervisors and the main stakeholders (Machine module designers). Point 2) has mostly had an influence on scoping the project, as seen in Chapter 2.2.

This chapter describes the high level processes used to complete this project. First, the project-related stakeholders are explained in Section 14.1. Next, the management processes and tools are described in Section 14.2. Then, the specific company analysis and scoping process is discussed in Section 14.3. Afterwards, the process used for architecture and design is described in Section 14.4. Section 14.5 wraps up the project management chapter and discuss the final design criterion: reflection and critical attitude.

14.1 PROJECT-RELATED STAKEHOLDERS

This section describes the stakeholders related to the project organization. Project related stakeholders are associated with both Océ and Eindhoven University of Technology (TU/e). Figure 48 shows an overview of the main project stakeholders, which is explained below.

Figure 48 Project stakeholders

The main stakeholder was me, the PDEng Trainee. I was responsible for executing the project. The trainee was assisted by supervisors from both TU/e and Océ. For TU/e this was Mykola Pechenizkiy and for Océ these were Edy Klomp and Rob Kersemakers. Edy Klomp was the main responsible contact. In addition to the supervisor, Ad Aerts was the manager responsible from the side of TU/e, while Ronald Fabel was responsible from the side of Océ.
14.2 PROJECT MANAGEMENT PROCESSES
The following high level management processes were executed to validate the project direction, reduce risks, and communicate/store the tasks and goals to be achieved.

Tasks
Purpose: Keep track of both long and short term tasks through milestones and product backlog

Two main methods were used: milestones and a task backlog. The milestone tracking contains a list of milestones with their description, planned completion, a completion marker and date shifts. Every time a planned completion changed, a reason was written down for accountability. In this way I kept track of the shifting milestones. A milestone trend graph was extracted and used for communication (See Figure 49 as an example). Appendix E shows an overview of milestone changes and their reasons.

![Figure 49 Milestone trend chart up until 2016-07-26](https://trello.com/)

A task backlog was used as part of an agile approach of this project. The tasks were allocated in weekly sprints. The main purpose was obtaining quick feedback, performing prioritization internally, and preventing management overhead during the week. Trello partially supported this process by providing a list of current tasks, the task backlog and the achieved results. Tasks in any list contain the name, output, maximum time I want to spend on it, and time actually spent. The comparison of time spent and planned time were used to improve time estimations.

Tracking hours
The hours to be spent were estimated at the beginning of the project based on type of activity. The estimated and actual time (as of 2016-07-26) can be seen in Table 16. These hours were updated weekly and a snapshot was saved including visualizations of the work done.

33 [https://trello.com/](https://trello.com/)
approach was used as a reflection method and to see whether particular activities take an unexpected amount of time.

**TABLE 16 HOURS SPENT IN THE PROJECT PER ACTIVITY**

<table>
<thead>
<tr>
<th>Activity</th>
<th>Expected hours</th>
<th>Actual hours spent</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain analysis</td>
<td>60</td>
<td>90.75</td>
<td>151.25%</td>
</tr>
<tr>
<td>Stakeholder analysis</td>
<td>80</td>
<td>139.75</td>
<td>174.69%</td>
</tr>
<tr>
<td>Management</td>
<td>140</td>
<td>87.25</td>
<td>62.32%</td>
</tr>
<tr>
<td>Architecture/Design</td>
<td>260</td>
<td>57</td>
<td>21.92%</td>
</tr>
<tr>
<td>Implementation</td>
<td>280</td>
<td>183</td>
<td>65.36%</td>
</tr>
<tr>
<td>Report</td>
<td>115</td>
<td>173.75</td>
<td>151.09%</td>
</tr>
<tr>
<td>Quality/Testing</td>
<td>80</td>
<td>3.5</td>
<td>4.38%</td>
</tr>
<tr>
<td>Progress meetings/Demo's</td>
<td>130</td>
<td>127.25</td>
<td>97.88%</td>
</tr>
<tr>
<td>Buffer</td>
<td>100</td>
<td>71.5</td>
<td>71.50%</td>
</tr>
<tr>
<td><strong>Sum</strong></td>
<td><strong>1245</strong></td>
<td><strong>933.75</strong></td>
<td><strong>75.00%</strong></td>
</tr>
</tbody>
</table>

There are a few observations from Table 16: Domain and stakeholder investigation took longer than expected due to the freedom provided throughout this assignment. Architecture and design, implementation, and quality and testing took less time due to the heavy focus on understanding the problem and reporting. The report took a longer time due to underestimating the complexity of the report task and an early start. Since only a small part of the code-base (the framework core) is critical, Quality/Testing took a relatively short time. In addition, a lot of quality-related refactoring tasks were executed as part of Implementation.

**FIGURE 50 WEEKS VERSUS PERCENTAGE OF TIME SPENT ON ACTIVITIES**

The activity distribution graph at the moment of writing (2016-07-26) can be seen in Figure 50. Interesting observations are: Domain analysis related tasks keep occurring throughout the project, as doing experiments properly requires gathering domain knowledge. Also, the report was started very early as this was a known weakness. Chapter 15 discusses a reflection on the difference between expected time and time spent.

**Risk management**

Purpose: keep track of risks and their mitigation and contingency strategies
Lists risks with possible results, impact, probability, triggers, mitigation, and contingency. Risks are added and changed during the project on a biweekly to weekly basis. The main risks (with high probability and impact) in this project are shown in Table 17.

**Table 17 Risks and their solution**

<table>
<thead>
<tr>
<th>Risk</th>
<th>How risk was handled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Framework scripts cannot be used interactively</td>
<td>Design allows for experiments with small datasets to verify approach. Bigger analysis is acceptable to run overnight.</td>
</tr>
<tr>
<td>Final report: confidentiality and quality</td>
<td>Keep in touch with the Intellectual property department for confidentiality. Start early and plan plenty of reviews for quality.</td>
</tr>
<tr>
<td>Too broad scope</td>
<td>Forced scoping to a number of concrete cases for the Niagara product.</td>
</tr>
<tr>
<td>Framework and scripts not accessible/reusable for stakeholders</td>
<td>Confront stakeholder with demonstrations regularly. Focus design on accessibility for non-programmers</td>
</tr>
</tbody>
</table>

**Social network**

Purpose: Keep track of the various people and their location/relation

In the beginning of the project, people were ordered by team/location and documented. Special relationships (e.g., referral by someone) or traits were noted per person. This document was used to track the stakeholders spoken to throughout the project. As the project progressed, relations with stakeholders became stronger and this document became largely obsolete.

**Retrospective**

Purpose: Improve processes

Approximately bi-weekly I held a retrospective and noted the main positive and improvement points. The results of this effort can be seen in Chapter 15.

**Keeping stakeholders up to date**

It is paramount to the success of the project that the stakeholders stay involved and I can connect my work to their processes. Hence I kept in touch with my stakeholders using:

- Presentations on critical points in the process. In practice this included two presentations on the concepts of anomaly detection, linking data sources, and the general framework.
- On demand meetings with stakeholders to present ideas and obtain their feedback. The two main types of stakeholders involved:
  o Potential future users (a set of machine module designers)
  o Those involved in data analysis initiatives
Keeping supervisors up to date

The Océ supervisors have been met weekly or biweekly and were asked to review process and content. The TU/e supervisor is consulted after a few review steps. All supervisors are kept up to date with on demand updates of status and content, and PSG meetings with monthly content updates. In particular, PSG meetings gave powerful feedback on this project.

14.3 COMPANY ANALYSIS

Throughout the first months of the project, the company context was investigated to find the main project goal. The main reason for this approach is getting prompt feedback (countering the lack of domain knowledge and the time limit imposed). The steps taken in this process are:

- Interviews with various stakeholders from various departments to gain insight into the status quo, opportunities and limitations
- Aggregation of the wishes from the interviews and a summary of the general trend. Commonly asked questions were gathered and a direction for research was formulated.
- The following topics were researched briefly:
  - Domain of data analysis to find opportunities in data analysis
  - Domain of high production printing to gain insight into data analysis in the company context, mainly through interviews. This lead to challenges to be addressed and input for scoping.
- The current concepts were scoped to more concrete problems
  - First, I generated concept solutions as an input for validation ideas with stakeholders
  - Then I got feedback from stakeholders on these concepts in sparring/brainstorm sessions to find direction for research and scoping
  - Afterwards, I drilled down to research the feasibility and expected impact of concepts
  - Next, I analyzed the opportunities within context of Machine module designers. I researched the specific opportunities in chosen context: anomaly detection and correlation
  - Finally, I investigated the overlap and potential synergy with other initiatives for this scope, to reduce risks and save work.
- Finally, I attempted to comprehensively write down results of the above processes to effectively communicate project argumentation

14.4 ARCHITECTURE AND DESIGN PROCESS

In parallel with analyzing the problem, a design and solution were devised. A number of architecture proposals and prototypes were used to confront supervisors and stakeholders. The steps taken in this process are as follows:

- Initially, design was based on the criteria for portability, reusability, and accessibility. This resulted in a first plugin-based architecture with a common interface. An abstract factory pattern was used for instantiation of plugins.
- Through feedback sessions with my supervisors this initial design was improved and a basic prototype was implemented. In addition, the position of this project within the company server (as part of ODAS) was made explicit.
Afterwards an overview of the potentially relevant features became clear, of which the selection in Chapter 5 remained. The following promising features were also identified, but (initially) not focused on:

- Providing data loading and preprocessing plugins.
  - No focus due to the data format currently being redefined and loading and preprocessing plugins already being developed. In addition, at this point it was deemed more interesting to verify the application of scientific techniques, rather than standard data manipulation.
- Providing an overview of all log files and meanings
  - Deemed too large a task for the potential benefit as 1) this knowledge is spread over a large number of people and 2) domain expertise is needed to properly describe the machine modules and log variables.
- Providing prediction analysis scripts for predictive maintenance
  - Understanding of the data and the behavior at this point is too low to properly do prediction. Especially within the scope of predictive maintenance, there are often not enough events to statistically conclude a pattern corresponds with a defect.

A focus was also put on reusability, accessibility, and portability. Performance and security were kept out of scope as they are of lower priority. In addition, preprocessing and data loading scripts were kept out of scope as the data format was being developed in parallel with this project. These steps were taken:

- Within the basic prototype an anomaly detection case was set up with multiple plugins. While developing plugins I noticed the initial choice for using a configuration file to load plugins was inconvenient; these files will likely not be updated by developers. In the next iteration an automatic scan of folders was implemented and configuration files were removed.
- The anomaly detection and basic framework functionality was shown to a set of stakeholders. Feedback showed automatic analysis over the fleet was very valuable. The anomaly detection plugins were (partially) rewritten to fit in the new framework.

In the 5th PSGM the following suggestions were made: 1) Focus more on sharing domain information, as it is a key point, and 2) Focus more on preprocessing and data loading. I also realized taking Pandas DataFrame objects as the standard data format was a good choice, as it offers a large number of standard functions and is consistent. From this point onwards I took the following steps:

- Defined and executed test cases for the framework core
- Designed a prototype to save and retrieve domain information linked to machine data.
- Developed a number of plugins for data loading and preprocessing using Pandas DataFrame’s.
- Developed a prototype for correlation of data sources using Pearson correlation.
- I demonstrated the above features to stakeholders and got positive feedback on the plugins and the sharing of domain information.

The core framework was fitting for the correlation case, but plugins had to be updated and refined based on generality needs and limitations due to the .csv file format. From this point onward the main design and features remained stable. The final feature to focus on were plugins and domain-specific scripts for a correlation-related experiment. The plan, at time of
writing, is increasing general robustness and consistency before final release. In addition, easier sharing of domain knowledge through GUI was investigated.

14.5 REFLECTION AND CRITICAL ATTITUDE

This section wraps up the project management processes. Most of the processes aim to support the final design criterion: reflection and critical attitude. The criteria of Usability and Reusability were approached with a reflection and critical attitude approach. Table 18 shows the success metric, as taken from the standard PDEng metric list. My current perceived level at the start of the project is 3 and the target level is 4.

**Table 18 Success metric for reflection and critical attitude**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Takes everything for granted</td>
<td>Lacks reflective thinking; sees flaws only when pointed at</td>
<td>Occasionally shows reflective thinking; reacts to errors and flaws when pointed at</td>
<td>Often shows reflective thinking; tends to seek errors and flaws</td>
<td>Consistently shows reflective thinking and calls attention to flaws</td>
</tr>
</tbody>
</table>

Throughout the last sections, reflection and critical attitude were applied in:

- Planning: I have regularly updated planning at own accord when looking at deadlines and priorities of existing tasks. I have, however, not been critical enough on the introduction of new tasks and needed guidance from the PSG.
- Risks: I have regularly assessed project risks and acted accordingly. The main risks being the creation of an unusable tool or a tool not supporting reuse.
- Stakeholder interaction: I have set up multiple sparring, demo, and “pair analysis” sessions with stakeholder to gain feedback and steer business cases.
- Supervisor interaction: I have interacted with my Océ supervisor well for feedback on content and process. However, within PSGM’s my focus has been too much on process to gain effective feedback on the content of the project.
- Design and architecture: Design and architecture has been reevaluated on both supervisor and stakeholder impact. However, some key focuses (domain knowledge sharing and preprocessing scripts) were not identified. Again, reassessing project focus is not completely successful.
- Scoping and concretizing has been a relatively difficult process. In hindsight I noticed that I spent a long time dwelling on which focus to give the project and gaining concrete examples of analysis. There should have been more focus on gathering such examples.

In the end, I perceive my development to have grown to rating four, but I should improve in 1) critically reassessing project focus and 2) using experts (particularly the PSG) effectively, and 3) effectively searching for concrete examples supporting the general problem and solution.

**Table 19 Success metric for reflection and critical attitude**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Takes everything for granted</td>
<td>Lacks reflective thinking; sees flaws only when pointed at</td>
<td>Occasionally shows reflective thinking; reacts to errors and flaws when pointed at</td>
<td>Often shows reflective thinking; tends to seek errors and flaws</td>
<td>Consistently shows reflective thinking and calls attention to flaws</td>
</tr>
</tbody>
</table>
Finally, this chapter summarizes the main lessons learned and other retrospectives. Throughout the project an (on average) biweekly retrospective was documented. Here I summarize the biggest learning moments. I also summarize what I would do differently in hindsight.

**Effective use of experts and resources**

First of all, I have had mixed success using the resources at my disposal. First, consider the coaches available. On one hand I started out with coaching session on project management, but throughout the project I did not continue these sessions. In addition, I should have used the expertise of my TU/e supervisor in data analysis more effectively by providing him with concrete analysis cases.

Secondly, I effectively shared my process, but had trouble effectively sharing my content throughout the project. At first I did not share live documents as much as I should have. In later stages I failed to add content to PSGM’s in an effective way. In particular, I could not present my content specifically enough in a summary form of e.g., a slide set. After some point I did not include content in the PSGM, which is wasteful.

In future projects I will consider more carefully: 1) who I can ask for feedback, 2) what their expertise is, and 3) how I should present the information, and when is the critical moment to ask for feedback.

**Lack of quality verification methods**

Throughout this project little quality, verification, and testing tools were used. The following factors held me back:

- Apart from the Impromptu Framework core, the code written is mostly a temporary experiment.
- Verifying a plugin using a library file or scientific technique is complex and depends heavily on the library. The complexity of writing a test scales with the complexity of the library

Nevertheless, the lesson is: quality and testing should have been approached more structurally.

**Integration with related initiatives**

An important part of this project was integration with related initiatives. Throughout the project I have kept in touch with the ODAS, Reflexion, Predictive maintenance, and the ODAS-related PDEng ST generation 2015 project. Staying connected has given me the following benefits: 1) Giving me extra insight and confirmation in machine module designer needs, 2) Introduced me to new stakeholders, 3) allowed me to re-use database and visualization tools, and 4) provided better context perspective of my project. Approach confirmed: Keeping good contact takes time, but has a good return on investment.

In addition, it was important to become part of the team. Initially, the Remote Service team seemed quite loosely coupled and it was hard to find a connection point work-wise. This is inherent to the PDEng graduation assignment, as it is not allowed to be on the critical path. Eventually we developed more interaction, for example concerning installation of Python libraries. The lesson learned: Connect with the team you are in, even if the work items are disconnected.
Binding with stakeholders

To follow up on the last topic, binding with stakeholders (including the related initiatives) was very fruitful. I experimented by asking for their feedback, showing them demo's to raise enthusiasm, and then tried to arrange for input to help them with practical cases. In the end my efforts paid off, as stakeholders stayed engaged and are willing to experiment with my solution. Approach confirmed: keeping good communication with stakeholders positively impacts the reception of the product.

Scoping towards the problem

Within this project, one of the main challenges was scoping towards a concrete problem. On a positive side, interviews and sparring sessions provided good information on stakeholder needs. On the other hand, moving from this information to a concrete problem was cumbersome. The lessons learned (and to be tried in future projects) are: Take concrete examples as much as possible, distinguish real musts and nice-to-have's as soon as possible, and try to communicate abstract ideas with concrete examples.

Writing as a weak point

At the start of the project, I identified writing a final report as a challenge for me. I started as soon as possible and planned reviews early to improve. This helped a lot, but also consumes a lot of time. The lessons I learned in the process to be more effective:

- Write to support my own argumentation, rather than seeing it as a deliverable
- Similarly with scoping: Give concrete examples as soon as possible
- Do not obfuscate concepts due to confidentiality
- Write to describe the reasoning more so than the results. Reasoning is (in general) more interesting.

In hindsight, what I would do differently

To wrap up this chapter, I would do the following differently:

- I now realized sharing domain knowledge is a more interesting and less explored aspect of an analysis framework. I would have spent more time on this concept from the start and spent less on scientific analysis techniques.
- I now realized preprocessing and data loading scripts are very valuable. I also confirmed these scripts were already being developed in the ODAS framework. Even though I saved some redundant work, I would have focused earlier on creating some convenient loading and preprocessing scripts.
- I would spend more time with coaches, e.g., for architecture, stakeholder involvement, and project management
- I would try to be more concrete in my written results
- I would present concrete results to the PSG to get better feedback on project content
APPENDIX A  AREAS OF INTEREST FOR DEMONSTRATOR

This appendix shows a number of cases in the company context, for different domains/departments in the company. During stakeholder analysis, a large number of stakeholders were interviewed. This information, in addition to a brainstorming session, were used to generate the following set of potential opportunity areas:

- **Data science areas:**
  - Automatic anomaly detection: automatically find odd-behaving machines
  - Error correlation: find root causes and relations between failures
  - Predicting failures: predict future failures based on known behavior

- **Enhancing the machine-owner communication and process:**
  - Machine owner-behavior analysis: analyze machine owner usage and propose improvements
  - Dynamic service technician manual: enhance service technician information with live status data and scoping potential failures
  - Context data for operator: helping operator do simple service operations with live status data and service suggestions

- **Predictive maintenance:** optimize service visit strategy based on recent information

- **Shared data analytics:** support sharing and re-use of data analytics results and methods

After defining a number of promising areas, sparring sessions were held and the areas were investigated further. The final ranking can be seen in A.1. The criteria for selection are:

- Expected buy-in of stakeholders based on their reactions
- Expected value based on questions to and from stakeholders
- Technical opportunities based on research on existing data
- Estimated feasibility and personal affinity

<table>
<thead>
<tr>
<th>#</th>
<th>Case</th>
<th>Main Pros</th>
<th>Main Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data science: Anomaly detection</td>
<td>- Valuable even when partially completed</td>
<td>- Challenge in finding right focus within the data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Reusability likely</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Value in speeding up process R&amp;D/Service</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Data science: Error correlation</td>
<td>- Supports step in Predicting failures and enhancing service manual</td>
<td>- Probable that information is missing or localized in domain experts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Root cause knowledge valuable for multiple stakeholders</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Technical solutions exist</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Shared data analytics</td>
<td>- Value in preventing redundant work</td>
<td>- From stakeholder enthusiasm: more likely to be ignored afterwards</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Stimulate cooperation</td>
<td>- Unsure about technological possibilities</td>
</tr>
<tr>
<td>4</td>
<td>Data science: Predicting failures</td>
<td>- Supports predictive maintenance</td>
<td>- Real failure data is missing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Value in time saved service/R&amp;D</td>
<td>- Can be difficult to find root causes to begin with</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Technical solutions exist</td>
<td>- High false positives and negatives rates are a danger</td>
</tr>
</tbody>
</table>
### A.1 AREAS OF INTEREST WITHIN THE PROJECT WITH THEIR PROS AND CONS

In the end, machine owner-related cases were deemed risky to achieve due to little influence and high security needs, in particular privacy. Data science related cases seemed promising given the company need and the availability of data and domain experts. In the end, the areas of automatic anomaly detection, error correlation, and shared data analytics came out as most promising. Given the generic nature of “shared data analytics,” I chose to focus on anomaly detection and error correlation with concrete cases, while keeping the shared data analytics concept as quality attributes supporting these cases. These quality attributes are usability and reusability.
APPENDIX B  OCÉ STATUS QUO

This appendix concerns an overview of the information flow within Océ. There are two main data suppliers for Océ: Machines in the field of Océ and business and machine data from Canon sources. B.1 shows an overview of the information streams within Océ. The list below summarizes these sources.

B.1 ABSTRACT OCÉ INFORMATION FLOW

OCÉ “RAW” SOURCES
- Machine functional logging (most of all Niagara): Large number of sensor and software logs about machine
- Machine error logging: Error reports and stack traces
- Océ printer business logs:
  o OSI (service), OMSI (marketing), OLI (logistics), OFI (financial)

CANON “RAW” SOURCES
- MIF (Machines in field)
  o Contract and placement information about machines
  o Information dirty: differences between multiple databases and interpretations
- UGW (Usage admin portal): Counters for usage of various kind of paper, ink, etc.
- MDS: Canon management document services, consists of printing statistics
- Totalfleet: Third party devices (non-MPS)
- Uniflow: Third party application managing fleet reports
- Financial info: Bills, overview of costs and savings
- Global/Corporate IT: Dashboard and back office data
- Service data: Faults/Incidents/Service requests
- Salesforce: Machines-In-Field-based contract data
APPENDIX C  ANOMALY DETECTION RESEARCH RESULTS

Anomaly detection research results are summarized in Section 6.1. This appendix contains the raw list of techniques with their key points, advantages, and disadvantages. First the main sources and terminology are discussed. Then, two types of techniques are summarized: Context-based and point-based. Finally, the chapter is wrapped up with conclusions and a reality check.

1) SOURCES
Google Scholar and the TU/e library were used with search terms “anomaly detection,” “outlier detection,” and “change detection.” The most promising looking papers were downloaded, of which two gave an overview:

- The paper by Chandola, Banerjee, and Kumar (Anomaly Detection: A survey). Chapter 4 – 11 are used as a focus. It describes groups of anomaly detection techniques.
- The book Data mining for service, Chapter 13 (Change detection from heterogeneous data sources) is also used. It discusses change point detection.

2) TERMINOLOGY
Class-based analysis:

- Multi-class: Assumption that labeled data belong to multiple normal classes
- One-class: Assumption that there is only one normal and one anomaly class

Univariate vs Multivariate

- Univariate = Involving one variable to be analyzed
- Multivariate = Involving multiple variables to be analyzed at the same time

Generative vs Discriminative

- Generative: probabilistic model of all variables, more flexible
- Discriminative: probabilistic model of interesting variables

3) CONTEXT-BASED TECHNIQUES
Data needs to have:

- Contextual/environmental attributes
  - Spatial: spatial data-based neighborhood is considered
  - Graphs: edges define the neighborhood
  - Sequential: The position in the sequence defines the neighborhood
    - Also event data with timestamps
  - Profile: Using multiple attributes to define a profile
- Behavioral/Indicator attributes
- Two main categories of context-based techniques:
  - Those that reduce to point anomaly detection
    - Identify the context for a test instance, then compute an anomaly score within the context
    - Generic reduction exists for non-clear contexts
  - Those that utilize the data structure
- Advantage: more fitting for real-life applications
- Disadvantage: applicable only if context can be defined
4) OVERVIEW OF POINT-BASED TECHNIQUES
Assumption for all techniques: point-based detection, no context-based detection.

Classification based techniques

Key aspects:
- Labeled training data needed
- Uses regular classification approaches, with anomalies not fitting in the normal classes
- Assumption that distinguishing between normal and anomaly can be learned practically

Advantages:
- Fast testing phase
- Possible to use powerful existing algorithms

Disadvantages:
- Availability of accurate labels is necessary
- Labels can be worse than scores

Neural-networks-based

Key aspects:
- Both multi- and one-class.
- Training/testing set approach with accepting/rejecting by a trained network
- Summary of papers and techniques in Table XIII
- Particular interest: Replicator neural networks for one-class anomaly detection

Bayesian-networks-based

Key aspects:
- Multi-class
- Univariate: Choose most likely label based on training data
- Multivariate: Aggregate the variables and apply Univariate
- Assumes independent attributes

Support-vector-machines-based

Key aspects:
- One class
- Learn region for training data. Inside = normal, outside = anomaly
- Interesting one: RSVM (Robust support vector machines)

Rule-based

- Both one- and multi-class
- Learn rules from training data. Not fitting any rule = anomaly
  - Learning e.g., RIPPER, decision tree
  - Not fitting: find the most likely rule. Anomaly core is inverse of best rule found
Nearest-neighbor-based anomaly detection

Key points:

- Assumption normal is a dense neighborhood; anomaly is far from the closest neighbor
- Uses a distance/similarity measure for all the points
  - For multivariate, often computed per dimension, then aggregated
- Two main categories:
  - Using the distance to k-th nearest neighbor as anomaly score
  - Using the relative density of an instance as anomaly score
- Often quadratic computationally complexity
- Often unsupervised

Advantages:

- Unsupervised is possible
  - Semisupervised gives better results
- No assumptions about generative distribution of data
- Easy to apply different data type, as long as distance measure can be defined

Disadvantages:

- In unsupervised: high dependence on dense neighborhood assumption
- In semi supervised: high false positive rate if not enough similar normal instances
- High computational complexity
- Very high dependency on the distance metric used

K-th nearest neighbor

Key points:

- Uses distance to kth-nearest neighbor as metric, then a threshold needs to be defined to see whether this score implies anomaly or not
- Extensions:
  - Other anomaly scoring, e.g., counting the number of nodes less than distance d or sum of distances to k nearest neighbors
  - Hypergraph-based methods for non-continuous attributes
  - Efficiency-techniques, e.g., pruning search space, taking random samples

Based on relative density

Key points:

- Purpose is estimating density for each instance
  - Low density = anomalous
- Performs poorly if the data has varying density regions
  - Solution: Local outlier Factor (LOF)
    - Variation: Connective-based Outlier Factor (COF)
    - Simple variation: Outlier detection using In-degree Number (ODIN)
    - Variation: Multi-granularity Deviation Factor (MDEF)
    - And more specific variations exist as well
**Clustering-based anomaly detection**

Key points:

- Unsupervised primarily
  - Semisupervised also explored
- Three main categories (based on assumptions):
  - Normal instances belong to a cluster, anomalies do not
  - Normal instances lie close to closest centroid, anomalies are far away from closest centroid
    - Can also work semi-supervised for clustering
  - Normal instances belong to large and dense clusters, anomalies belong to small and/or sparse clusters
- A number of optimizations and approximations exist
- Different from nearest-neighbor: they rely on the cluster instead of a local neighborhood
- Complexity: Fast test, training can be slow depending on pairwise comparisons

Advantages:

- Unsupervised possible
- Can be adapted to complex data time by adding a fit clustering algorithm
- Testing phase is in general fast

Disadvantages:

- Performance depends heavily on the effectiveness of clustering algorithm used
- Often byproduct of clustering, not focused on anomaly detection
- Often instances are forced to be part of cluster, even if they are anomaly
- Only effective when anomalies do not form a cluster
- High computational complexity

**Statistical anomaly detection techniques**

Key points:

- Assumption: Anomaly is identified by not (likely) being generated by the statistical model generating the normal behavior
  - Hence, normal instances are within high probability regions of the model, while anomalies occur in low probability regions
- Fit a statistical model to the normal behavior, then use inference test to see if odd instance occurs (i.e., in the case of low likelihood).
- Both parametric and nonparametric
- Computational complexity depends on underlying statistical model. Simple distributions are often linear; more complex ones are potentially quadratic

Advantages:

- If the underlying statistics are valid, the approach is easy to be proven for validity
- Anomaly score has confidence interval, which is useful for reasoning how to act (is it likely an anomaly, or is it very unsure)
- Can be unsupervised if distribution estimation is robust to anomalies
Disadvantages:
- Assumption of data being generated from distribution is often not true
- Choosing the right statistic to use is hard
- Histogram-based techniques cannot see the anomalousness of a multivariate (combination of variable) case

**Parametric techniques**

Key points:
- Main methods: distance from instance to estimated mean for anomaly score
  - Gaussian model-based
    - Grubb’s test (univariate)
    - For multivariate
    - For graph structured data
    - For OLAP (Online Analytical Processing) cubes
  - Student-t test has been used
  - As well as Ghi-squared
  - Regression-based: Uses residual of test instances for anomaly score
    - For time-series data
    - Akaike information Content (AIC): detect anomalies in set
    - Autoregressive Integrated Moving Average (ARIMA): robust approach

**Nonparametric techniques**

Key points:
- Use nonparametric model, so model structure not defined on query, but obtained from data
- Often fewer assumptions on data compared to parametric
- Main approaches:
  - Histogram-based
    - Often used in intrusion detection
    - Steps: 1) make histogram based on training, 2) test if instance falls in one of the buckets, if not: anomaly
  - Kernel function-based
    - Use kernel function to approximate density. Otherwise similar to parametric approaches

**Information theoretic anomaly detection techniques**

Key points:
- Based on analyzing information content using theoretic measures, such as Kolomogrov complexity or entropy.
- Assumption: anomalies in the data induce irregularities in the information content
- Exponential time complexity

Advantage:
- Works unsupervised
- No assumptions on underlying statistical distribution

Disadvantage:
- Performance highly dependent on information choice, often exponential
- Dependency on size of substructure used for defining normal cases
- Hard to define anomaly scores

**Spectral anomaly detection techniques**

Key points:
- Approximate the data using a combination of attributes capturing the biggest variability.
  - Determining these subspaces is the main challenge
- Assumption: There exists a subspace of the data in which normal instances and anomalies are distinguishable
- Can be used for time series
- Typically quadratic in number of dimensions

Advantages:
- Automatic dimension reduction
- Unsupervised is possible

Disadvantages:
- Assumption is big
- High computational complexity in general

**Change point detection**

Key points:
- Detecting structural changes in data generation mechanism behind observed data
- Typical types of change points: cusps, steps, changes in frequency, changes in amplitude
- CUSUM (Cumulative summation) is a well-known baseline method
- Number of references in 2.2 (in folder Data Mining for Service references)
- SST algorithm is proposed

5) **WRAP UP AND REALITY CHECK**

In the end, a number of anomaly detection technique types were investigated, but only a few were fit for use. In particular, a requirement was to have either a clear implementation description or an actual implementation of algorithms. An implementation that performs well enough was chosen in the end, as can be seen in Section 6.1. This implementation was "A least-squares approach to anomaly detection in static and sequential data" (Quinn & Sugiyama, 2013), which falls into the statistical anomaly detection category.
APPENDIX D  STORING DOMAIN INFORMATION

The database used is a PostGreSQL\textsuperscript{34} database. The Python library “psycopg2”\textsuperscript{35} as interface to the database. The database format in D.1 describes both the database format and the format used inside the Python environment.

The fields \textit{machine}, \textit{datatreelink}, and \textit{variable} are used to identify the machine, module of the machine, and variable within that module respectively. The \textit{starttime} and \textit{endtime} fields identify the start and end of the relevant time for the comment. The \textit{author} and \textit{comment} fields are dependent on the user input.

<table>
<thead>
<tr>
<th>name</th>
<th>id</th>
<th>datatreelink</th>
<th>machine</th>
<th>variable</th>
<th>starttime</th>
<th>endtime</th>
<th>author</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Py type</td>
<td>Serial primary key Auto generated</td>
<td>String</td>
<td>String</td>
<td>String</td>
<td>Long</td>
<td>Long</td>
<td>String</td>
<td>String</td>
</tr>
<tr>
<td>db type</td>
<td>Text</td>
<td>Text</td>
<td>Text</td>
<td>Bigint</td>
<td>Bigint</td>
<td>Text</td>
<td>Text</td>
<td></td>
</tr>
</tbody>
</table>

\textsuperscript{34} \url{https://www.postgresql.org/}
\textsuperscript{35} \url{https://pypi.python.org/pypi/psycopg2}
APPENDIX E  MILESTONE TREND CHANGES

This appendix summarizes the changes made in milestones over the course of the project. It is meant to clarify the choices of focus and priorities.

<table>
<thead>
<tr>
<th>Change reason</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stakeholder presentation; insight to get more in depth before committing. Also considered TU/e as presentation point</td>
<td>16-feb</td>
</tr>
<tr>
<td>Report on analysis is delayed for two weeks since scoping took longer than expected. Impact is not expected to be large enough to counter effect by reducing 3 cases to 2</td>
<td>16-feb</td>
</tr>
<tr>
<td>Case 3 was canceled for now, and case 2 was shifted to create more breathing room.</td>
<td>16-feb</td>
</tr>
<tr>
<td>Framework: Hello world was canceled due to another approach: First define case, then expand from case, rather than start from general framework</td>
<td>16-feb</td>
</tr>
<tr>
<td>Process split up in two parts, since impossible to complete process before second demonstrator is done</td>
<td>24-mrt</td>
</tr>
<tr>
<td>Stakeholder presentation idea: insight has increased, but interaction with stakeholders was postponed, which means later verification, later presentation. This might impact other presentation in form of cancellation and replacing with another topic, but delay is not expected</td>
<td>08-apr</td>
</tr>
<tr>
<td>Finishing prototype anomaly detection is pushed forward, because including more algorithms and stakeholders is deemed more valuable than finishing quickly. This also impacts Framework: Anomaly detection and Report: Anomaly detection. There will be an impact on the second case, which will be assessed later.</td>
<td>08-apr</td>
</tr>
<tr>
<td>Added milestones with respect to more user interaction, i.e., having joint workshops in which the product is validated/used. The time spent on these workshops is estimated to be max half a week, so no big impact in planning is expected</td>
<td>17-mei</td>
</tr>
<tr>
<td>Shifting Report, Architecture, and implementation of prototype for anomaly detection part. Reason: Redoing of presentation of argumentation and redo of architecture was necessary. Expected extra work from now is optimally one week, so a two week interval is taken (towards June 6th). Interval until next milestone (prototype next part) is then still a month, which would mean complete the entirety of correlation in that time. <strong>This is infeasible. See next point</strong></td>
<td>17-mei</td>
</tr>
<tr>
<td>Error correlation development is shifted forward as a result of the previous point. It is estimated that around 1.5-2 months are needed to complete this task. There is a time constraint for internal review of the report until 15-07, since there is an important review at the start of august. The demo of the system is not shifted, as it would be too late to get proper feedback. Finishing the system is shifted, such that that feedback can be utilized (to the start of august)</td>
<td>17-mei</td>
</tr>
<tr>
<td>Workshops were canceled due to high priority correlation case for architecture</td>
<td>27-mei</td>
</tr>
<tr>
<td>Correlation cases was moved forward as progress was slower and more dependent on domain expert than expected. Impact is expected to be low, as report can still be made in parallel</td>
<td>04-jun</td>
</tr>
<tr>
<td>Report deadline for Judy is put forward as they require the report well beforehand. This was an oversight in planning, but has no impact on further tasks, as a version of the report had to be ready anyway</td>
<td>04-jun</td>
</tr>
<tr>
<td>Put deadlines of report and correlation examples at the real deadline moments.</td>
<td>11-jul</td>
</tr>
<tr>
<td>Framework no need to be changed based on correlation, plugins can be added and the structure is sufficient</td>
<td>11-jul</td>
</tr>
<tr>
<td>Major shift for report deadline: IP is shifted back, new (more realistic) review moments are planned based on current progress</td>
<td>22-jul</td>
</tr>
<tr>
<td>Correlation case and demo are moved forward as no satisfying result has been found.</td>
<td>22-jul</td>
</tr>
<tr>
<td>Final presentation date is moved as the date is now set</td>
<td>22-jul</td>
</tr>
</tbody>
</table>
### GLOSSARY

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis script</td>
<td>Executable program with some data as input and data (potentially with visualization) as output.</td>
</tr>
<tr>
<td>Canon</td>
<td>Canon, Inc.</td>
</tr>
<tr>
<td>Data</td>
<td>Values in some standardized format, not implying readability.</td>
</tr>
<tr>
<td>Data analysis / Data science</td>
<td>Techniques, technologies, and processes to extract information from data.</td>
</tr>
<tr>
<td>Data analysis framework</td>
<td>A software solution providing the possibility to implement or link data analysis software to it and support the data flow between data analysis software.</td>
</tr>
<tr>
<td>Demonstrator</td>
<td>Practical use case showing the value of the developed system.</td>
</tr>
<tr>
<td>Final Project Evaluation (FPE) Committee</td>
<td>Group responsible for the final evaluation of the project this report concerns. This group consists of the supervisors at Océ and TU/e, ST Program Director (or delegate), and optionally the program director from one of the other PDEng programs.</td>
</tr>
<tr>
<td>Information</td>
<td>Data which is interpretable to answer some question.</td>
</tr>
<tr>
<td>KPI</td>
<td>Key Performance Indicator: metric with corresponding values which indicate rate of success.</td>
</tr>
<tr>
<td>Media (type)</td>
<td>Type of material used for printing, often denoted by a media name and size.</td>
</tr>
<tr>
<td>Project Steering Group (PSG)</td>
<td>Group which steers and gives feedback to the project this report concerns. This group consists of the supervisors at Océ and TU/e and the author.</td>
</tr>
<tr>
<td>The company, Océ</td>
<td>Océ Venlo (Part of Océ, A Canon Company).</td>
</tr>
<tr>
<td>The university, TU/e</td>
<td>Eindhoven University of Technology.</td>
</tr>
<tr>
<td>Tool</td>
<td>A piece of software which obtains, transforms, or visualizes data.</td>
</tr>
</tbody>
</table>
BIBLIOGRAPHY


ABOUT THE AUTHORS

Luc de Smet completed his bachelor’s in Computer Science and his master’s in Computer Science and Engineering at Eindhoven University of Technology, graduating in 2014. His master's thesis concerned combining process mining and queue analysis to provide operational support for business processes. His main topics of algorithms, modeling, and in general software design and architecture. He has some work experience in both educational and industrial settings and expanded his knowledge and experience in the Software Technology program of the Stan Ackermans Institute at Eindhoven University of Technology.
4TU. School for Technological Design, Stan Akkermans Institute offers two-year postgraduate technological designer programmes. This institute is a joint initiative of the four technological universities of the Netherlands: Delft University of Technology, Eindhoven University of Technology, University of Twente and Wageningen University. For more information please visit: www.4tu.nl/sai