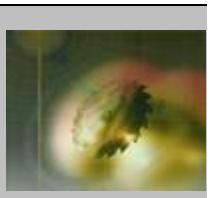
International journal of COMADEM
Vol # No # (January 2017) p.##-##

COMADEM

International Journal of Condition Monitoring and Diagnostic Engineering Management



A generic framework for data quality analytics

Miguel Castaño Arranz[a]*, Anna Gustafson[b], Hussan Al-Chalabi[a]

[a] *Division of Operation and Maintenance, Luleå University of Technology, Luleå, Sweden*
[b] *Division of Mining and Geotechnical Engineering, Luleå University of Technology, Luleå, Sweden*

∗ *Corresponding author. Tel.: +46 920-493988*; *email*: *miguel.castano@ltu.se*

ABSTRACT

The challenge of generalizing Data Quality assessment is hindered by the fact that Data Quality requisites depend on the purpose for which the data will be used and on the subjectivity of the data consumer. The approach proposed in this paper to address this challenge is to employ a semi-automated user-guided Data Quality assessment. This paper introduces a generic framework for data quality analytics which is mainly composed by a set of software units to perform semi-automated Data Quality analytics and a set of Graphical User Interfaces to enable the user to guide the Data Quality assessment. The framework has been implemented and can be customized according to the needs of the purpose and of the consumer. The framework has been instantiated in a case study on Long-hole drill rigs, where several Data Quality issues have been discovered and their root cause investigated.

Keywords: *Data Quality; Maintenance; Long-hole drilling; Mining.*
Article history: *Received ; Published .*

1. Introduction

Efficiency and effectiveness of maintenance strategies can be evaluated through Maintenance Performance Metrics (MPM)[1], which have been extensively reviewed in the literature[2]. The accurate evaluation of MPMs is hindered by Data Quality (DQ) issues[3]. Many organizations have more data than they can use and nevertheless don't have sufficient quality data for adequate decision making[4,5]. This lack of quality data often involves decisions based on personal bias instead of based on fully data-driven technologies[6]. Examples in maintenance literature include the importance of DQ in extended warranty predictions[7].

The characterization of Data Quality (DQ) depends on the purpose which a specific consumer will give to the data. This dependency leads to a subjective definition of DQ[8] on how data is fit for a specific purpose[9,10]. Multiple data consumers may have different requirements on the same dataset. Identification of the data consumers, identification of their needs and translation of their needs into multiple criteria[11] are therefore common steps in DQ assessment. However, it is often that data consumers are not aware of their needs beforehand, and they request a generic DQ assessment of unclear type. This paper targets the identification of such generic assessment and the definition of a generic framework. Many authors have tried to identify a standard set of DQ properties valid for data related to any product[12], whilst other authors describe this task as nearly impossible[13].

All information sources have bias and errors. However, two sources of bias and errors allow to classify the data in either subjective or objective data. In subjective data, bias and errors are introduced by human judgment, whilst in objective data bias and errors are introduced by e.g. sensor or model errors [14].

A usual source of DQ problems in CMMS systems is the fact that they are populated with a great extent of subjective data[15]. Large amount of manual information input in the system has a negative impact on the data since workers enter information of varying quality depending on their level of motivation, their knowledge of the system, their workload, etc.[16]. A possible approach to DQ analytics which is addressed in this paper is to first identify DQ issues by analyzing the data, then investigate the root cause of the issue and finally clean the data if adequate[7].

A framework for determining DQ issues has been developed and is introduced in this paper. A framework is a basic structure underlying a system. In this paper, framework is understood in terms of computer science, where a framework is a general skeleton for an application software that can be customized into a wide variety of specific real applications. A framework describes the components that a complete application will have, but it doesn't include the implementation of these components.

Additionally, this paper goes beyond the conceptualization of the framework and reports an implementation that can be customized to adapt to DQ requirements for specific applications. The framework has been customized and implemented for 2 types of data formats in relation to the maintenance of a Simba long hole drill rig (Reliability format, and Cost format).

From the taxonomy of DQ properties, this study focuses on the intrinsic data properties, which are the most objective[17]. Other authors have stated other properties of data which require access to additional information and cannot be generalized in a generic framework. These properties (such as depreciability and informativeness) are therefore out of the scope of this work.

Depreciability is understood as the loss of value of the data with time, e.g. because significant process changes have been performed and the data does not reflect the current process [18]. Informativeness depends on the inference/investigation technique which will be later applied on the data [19].

This paper is structured as follows. In Section 2, the goals of the framework are identified. In Section 3, the structure of the introduced framework is discussed. Section 4 includes details of the Software Units which form part of the framework. Section 5 includes details of the sub-Graphical User Interfaces which form part of the framework. Section 6 includes a case study using data from the mining industry. Finally, the conclusions are given in Section 7.

2. Goals of the generic Data Quality Framework

Interviewing data consumers and reviewing literature, the following generic DQ properties have been identified:

- Redundancy, i.e. duplicated information.
- Abnormality, e.g anomalous data which has been erroneously recorded or reflects unusual/unique situations.
- Absence, e.g. missing data values.
- Language issues, such as typos, grammatic errors or notation inconsistency produced by e.g. abbreviations.

Redundancy and absence have been identified as DQ problems in a survey of Australian engineering asset management organizations[12]. Absence[20,21] and typos[21] have also been previously identified. Even if such metrics are targeted to be generic, they require adequate modifications depending on the purpose for which the data is used and the subjective understanding of the data consumer. For example, two identical data entries may in some cases be understood as redundant, but in other cases as entries related to two separate events. Another example is the inconsistency in text due to abbreviations. Such inconsistencies in textual data is by some data consumers considered as irrelevant (they could figure out by themselves the actual meaning) and as very relevant for other data consumers, since it would hinder the application of Natural Language Processing (NLP) technologies. Due to this difference in DQ evaluation which depends on the purpose and on the data consumer, previous authors have concluded that DQ assessment should be automated as much as possible but still be as user-guided as necessary [11].

In the traditional workflow within DQ analysis, two types of actions can be undertaken when low quality data is identified: i) improve historical data, and ii) identify the root cause of the data deficiency and improve the work process[3]. Considering this background, the goal of the framework has been established as twofold:

- Create a generic DQ framework which can be customized for particular purposes whilst leaving the freedom to the data consumer for readjusting the definitions of the DQ properties.
- Providing tools to navigate the data in search for DQ issues.

Posterior investigation of the issues, their root cause and appropriate data cleansing are left out of the framework.

3. Structure of the Generic Data Quality Framework

An assumption made to develop the framework is that the data is accessible as a set of entries structured according to fields (see Figure 1). Each entry will therefore have an associated value for each of the fields (field value). Common data formats of this kind are excel sheets or CSV files.

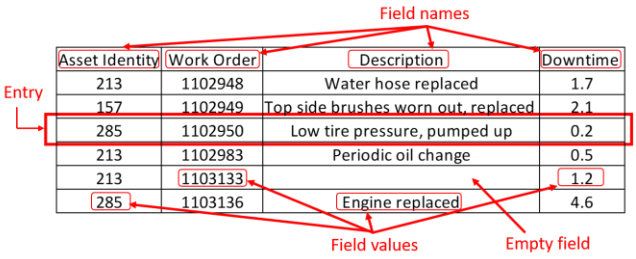


Figure 1. Data structured assumed developing the framework

The following Units for part of the framework:

- Software Units:
  - A customizable Import Software Unit.
  - A Data Save Software Unit.
  - A Report Software Unit.
  - Analysis Software Units.
    - A NLP Software Unit
    - A Missing Data Analysis Software Unit
    - A Redundant Data Analysis Software Unit
    - An Anomaly Analysis Software Unit
- Graphical User Interfaces (GUIs):
  - A set of customizable sub-GUIs to interact with the Analysis Software Units.
    - NLP sub-GUI
    - Missing Data Analysis sub-GUI
    - Redundant Data Analysis sub-GUI
    - Anomaly Analysis sub-GUI
  - A customizable "fit for purpose" GUI which aggregates the needed sub-GUIs for a specific DQ purpose.
- Application Data in a format which can manage and store the original data as well as processed data which is produced by the Analysis Software Units.
- Storage Units:
  - A Temporary Storage Unit which hosts the Application Data and can be accessed by all the Software Units.
  - A Permanent Storage Unit to save processed data.

The interaction between software and storage units is characterized by three types of events: Import Events, Save Events and Analysis Events. Figure 2 depicts the information flow between Units during these events.

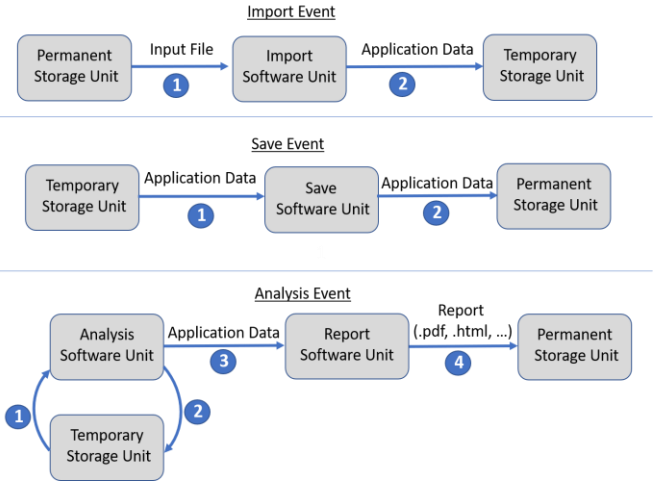


Figure 2. Events with interaction between framework Units

During an Import Event, the Import Software Unit reads data from a file in the Permanent Storage Unit, converts the data into the Application Data format and stores the data in the Temporary Storage Unit. Common file formats to be imported are excel and csv, but also it must be possible to read files previously saved in the Application Data Format.

During a Save Event, the Data Save Software Unit accesses the Application Data (allocated in the Temporary Storage Unit) and stores it in the Permanent Storage Unit. The goal of this event is to reduce future computational effort by saving processed data for future reuse.

An Analysis Event is triggered by a specific analysis request. First the corresponding Analysis Software Unit processes the Application Data according to the request. Partial or complete processing can be avoided depending on the availability of processed data saved from previous Analysis Events. Secondly, Application Data may be stored in the Temporary Storage Unit in order to maintain processed data which required large computational effort for future reuse. Finally, the Report Software Unit will receive Application Data and report the analysis result which is saved on the Permanent Storage Unit.

These events may be triggered by either the interaction of the user with the *fit for purpose* GUI or during the initialization of the sub-GUIs. More explicitly, Import and Save Events can be triggered from buttons in the menu bar from the *fit-for-purpose* GUI. Different Analysis Events can be triggered using the sub-GUIs. During the initialization of the sub-GUIs, analysis events are triggered to reconfigure the information displayed at the sub-GUIs.

An instantiation of the framework refers to the particularization of the framework for specific data, a specific application and for a specific data consumer. During the instantiation, a *fit-for-purpose* GUI is created by deploying the adequate sub-GUIs over an empty canvas as illustrated in Figure 3.

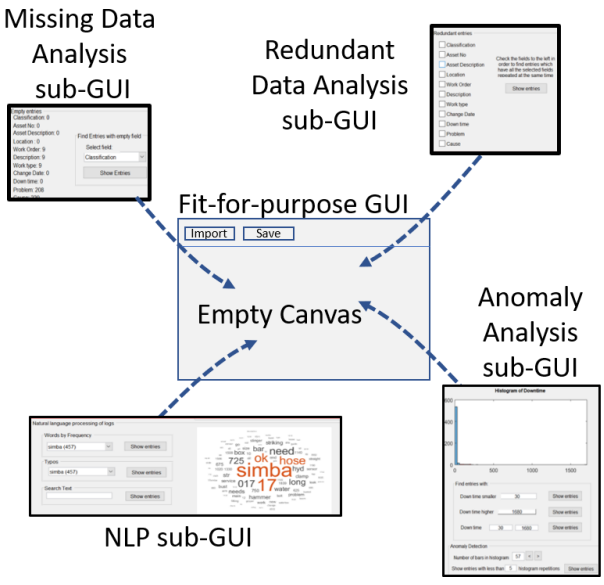


Figure 3. Deploying sub-GUIs during framework instantiation

4. Software Units

This section describes the Import, Save and Report Software Units as well as all the types of Analysis Software Units. Only the Import and Report Units may be customized, the rest of the Units are generic and can be used in any instantiation of the framework.

4.1. Import Software Unit

Functionality: When an import event is triggered by the *fit-for-purpose* GUI, this unit receives input files with the data and converts it to the Application Data Structure.

Customization: This unit has to be reprogrammed depending of the format of the input file (.xlsx, .csv, …) and the format of its contents. For example, in some cases the first row in data files is used to store metadata (such as the field names) and is some cases multiple rows are used to store metadata.

4.2. Save Software Unit

Functionality: When a save event is triggered by the *fit-for-purpose* GUI, this unit receives the Application Data and stores it in the Permanent Storage Unit.

4.3. Report Software Unit

Functionality: This unit receives instructions from the Analysis Software Units and generates a readable report (e.g. pdft, html, …) with the analysis results. These reports include data entries which are potentially problematic as well as the reason why they are problematic.

Customization: Customization of this Unit is not strictly necessary. However, some organizations or data consumer may want e.g. the reports to have a specific file format, include the name of the worker who generated the reports, include logos, include watermarks, and so on.

4.4. Natural Language Processing Software Unit

Natural Language Processing (NLP) is of relevance in the presence of a vast amount of text resulting from manual inputs. NLP has previously been used to process maintenance unstructured text logs in order to e.g. analyze the frequency of word appearances[22], in order to gain insight diagnostic fault trees[23], or in order to classify scheduled and unscheduled actions using clustering[24].

The NLP Unit can reduce words to their stem based on the semantic context. Namely, the following steps are performed: tokenizing, semantic annotation, normalization, removal of stop words, substitution of capital letter for low case letters. The result is that a word such as “building” will be wither reduced to the stem “build” from the verb “to build”, or to the noun “building” which refers to an architectural construction. This reduction will depend on the semantic context determined by the sentence.

Functionality:

Word cloud chart: generates a visually appealing map which represent the most frequent words in the text. Such maps are used in several contexts to provide an overview which facilitates text understanding [25]. Several studies have investigated the effectiveness and perception of word clouds by varying different visual properties [26], such as font size [27] or word position.

Find words by frequency: generates a list with the words ordered by how many times they are repeated in the text.

Find typos: finds words which are not in the dictionary and listing them by number of repetitions.

Find free text: find the entries with a specific character sequence.

4.5. Redundant Data Analysis Software Unit

Functionality: This Unit finds redundant data entries on request according to a specification of redundancy in terms of which identical fields must redundant entries have.

4.6. Missing Data Analysis Software Unit

Functionality: This unit finds data entries with specific empty fields.

4.7. Anomaly Analysis Software Unit

Functionality:

Find thresholds for abnormally low and high numerical values.

Determine an adequate number of histogram bars for numeric data visualization.

Find entries including numeric values either within specified intervals, above a specified threshold or below a specify threshold.

5. Sub-Graphical User Interfaces

The interfaces to trigger the import and save events are generic and can be simply invoked through menu options in the *fit-to-purpose* GUI.

The following subsections describe de sub-GUIs which allow the user to interact with the Analysis Units. Each subsection includes the purpose of the sub-GUI, its functionalities, its customization and its initialization. Customization refers to the tasks which have to be done in order to deploy the sub-GUI to a new DQ project. Initialization is the automatic reconfiguration in the sub-GUI which occur after an import event.

5.1. Natural Language Processing sub-GUI

Purpose: The sub-GUI is designed to navigate text fields in search for potential DQ problems (see Figure 4).

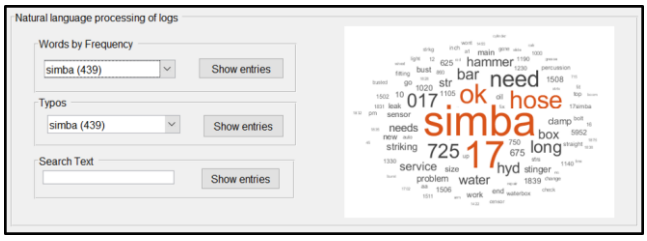


Figure 4. Natural Language Processing sub-GUI

Functionalities:

Visualization of a word cloud chart

Searching words by frequency. A drop-down list of words ordered by frequency is displayed in a drop-down menu. Selecting a word of the list and clicking on the *Show entries* button initiates an analysis event which reports the entries including the selected word.

Searching entries with typos. a drop-down list of typos ordered by frequency is displayed in a drop-down menu. Selecting a word of the list and clicking on the *Show entries* button initiates an analysis event which reports the entries including the selected word.

Searching free text. Introducing free text and clicking on the *Show entries* button initiates an analysis event which reports the entries including the introduced free text.

Customization: For each text field of interest, a sub-GUI has to be deployed in the *fit-to-purpose* GUI and linked to the field.

Initialization: When new data is loaded, the NLP Software Unit analyses the data and automatically sends to the sub-GUI the following items for appropriate display: a map of words, a list of words by frequency and a list of typos.

5.2. Redundant Data Analysis sub-GUI

Purpose: This sub-GUI is designed to navigate the data in search for redundant entries (see Figure 5).

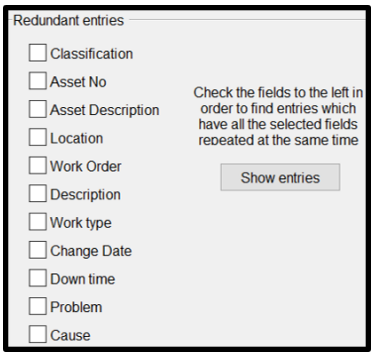


Figure 5. Redundant Data Analysis sub-GUI

Functionality: This sub-GUI allows the user to define what is understood as redundant data by selecting a number of fieldnames using check boxes. An Analysis Event is invoked when pushing the *Show Entries* button. In this Analysis Event, the Redundant Data Software Unit finds the entries which share identical values in all the selected fieldnames simultaneously. The Redundant Data Analysis Software Unit passes sets of redundant entries to the Report Software Unit and a report is generated.

Customization: The sub-GUI is simply added to the *fit-for-purpose* GUI and requires no further customization.

Initialization: When new data is loaded, the Redundant Data Analysis Software Unit analyses the data and sends a list of field names to the sub-GUI for appropriate display with a check box for each field. The drop-down-list with field names is also deployed automatically.

5.3. Missing Data Analysis sub-GUI

Purpose: The sub-GUI is designed to find missing field values (see Figure 6).

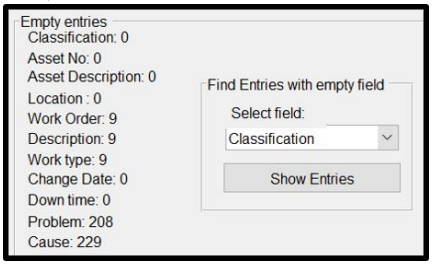


Figure 6. Missing Data Analysis sub-GUI

Functionality: The sub-GUI lists all the field names and the number of entries with the corresponding field missing.

Customization: The sub-GUI is added to the *fit-for-purpose* GUI and requires no further customization.

Initialization: When new data is loaded, the Missing Data Analysis Software Unit analyses the data and sends to the sub-GUI a list of field names with the number of times that each field has an empty value. These field names and their associated number of empty entries are automatically displayed on the sub-GUI. The drop-down-list with field names is also deployed automatically.

5.4. Anomaly Analysis sub-GUI

Purpose: The sub-GUI is designed to find unusual numeric values based on histogram navigation (see Figure 7).

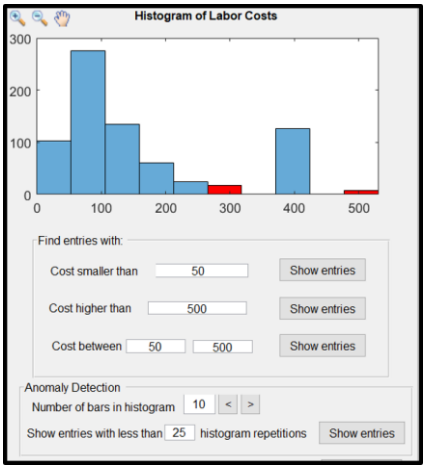


Figure 7. Anomaly Analysis sub-GUI

Functionalities:

The sub-GUI must allow navigation of the histograms by:

- Zooming in and out as well as panning.
- Choosing the number of bars to split the histogram.
- Choosing a minimum number of entries under which a histogram bar is considered to be abnormal and is consequently highlighted.

The indexes related to abnormal values found navigating the histogram are sent to the Report Software Unit when the user clicks on the corresponding *Show entries* button. Such values may include: values larger than a selected threshold, values smaller than a selected threshold, values within selected intervals, and values with low number of repetitions related to the highlighted histogram bars.

Customization: For each numerical field of interest, a sub-GUI has to be deployed in the *fit-to-purpose* GUI and linked to the field.

Initialization: When new data is loaded, the Anomaly Analysis Software Unit analyses the data and automatically chooses default values for the thresholds and the number of histogram bars.

6. Case study, Boliden's Tara mine

Boliden's underground mine in Tara, Ireland was chosen for the case study. It is one of the largest Zink-mines in the world. Mining started in 1977 and annual production is around 2.6 million tons of ore. In Tara, Epirocs Simba Long hole drill rigs are being used for production drilling.

Maintenance Work Orders (MWO) has been collected from one Simba drill rig (see Figure 8) and used as input data. The drill rig was first taken into service in April 2017. MWOs record semi-structured information regarding maintenance activities and their analysis can provide insight on e.g. reliability, maintenance, and planning[28]. MWO data has been previously used in literature to extract knowledge to predict maintenance[29].

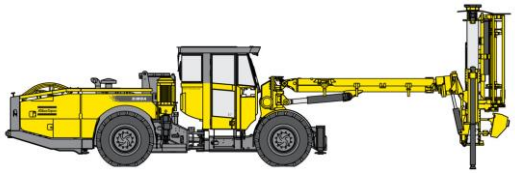


Figure 8. Simba, Long hole drill rig (Courtesy Epiroc)

The data has been exported from MAXIMO for the period April 2017-September 2019. It was collected in two different formats for different purposes in order to test the customization and instantiation of the framework in two fit-to-purpose scenarios. These two formats are:

- Maintenance Cost Format (MCF): oriented to be used for maintenance cost calculations.
- Reliability Analysis Format (RAF): oriented to be used for reliability analysis.

6.1. Instantiation of the framework

An instance of the framework has been created for each of the data formats.

Figure 9 is a screenshot of the *fit-for-purpose* GUI resulting for instantiating the framework on the MCF. The purpose is to use the data for LifeCycle Cost (LCC) calculations. Three Anomaly Analysis sub-GUIs have been deployed and liked to the numerical fields "Material Cost", "Labor Cost" and "Service Cost". No Redundant Data Analysis sub-GUI has been deployed, since two identical entries can reflect separate costs and are not considered as redundant (e.g. two separate workers reporting their labor cost at the end of the shift). A Missing Data Analysis sub-GUI has not been deployed, since entries with missing costs will not have any impact when costs are summed during LCC calculations.

Figure 10 is a screenshot of the *fit-for-purpose* GUI resulting for instantiating the framework on the RAF. One analysis sub-GUI of each kind has been deployed. The Anomaly Analysis sub-GUI has been deployed on the *Downtime* field which stores numeric values representing the downtime of the asset (in hours). The NLP sub-GUI has been deployed on the *Description* field, which is a free text field where workers can input a description of the maintenance action.

6.2. Analysis of Reliability Analysis Format

Navigating the data with the software tool, the following DQ issues were found as potential problems. The project team together with the project partners investigated the DQ issues and tried to identify the root cause of the DQ problems.

DQ Issue: Using the Anomaly Analysis it was found that there are entries with downtime which seem unreasonably low. In cases, the registered downtime of the rig was only a few seconds.

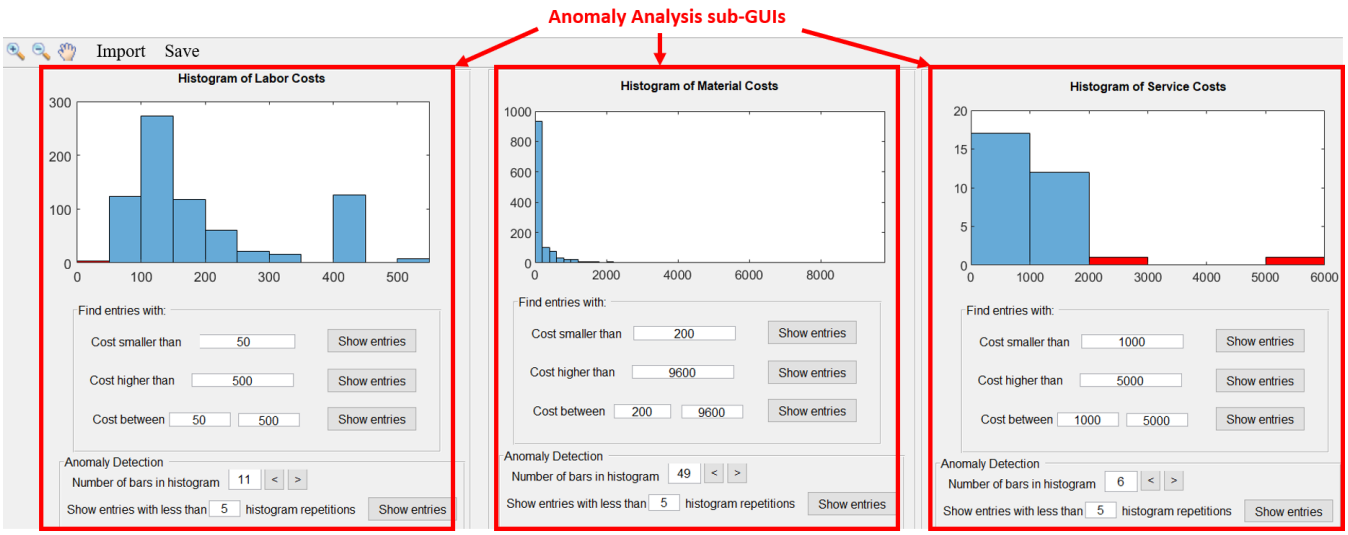


Figure 9. Fit-for-purpose GUI for the Maintenance Cost Format

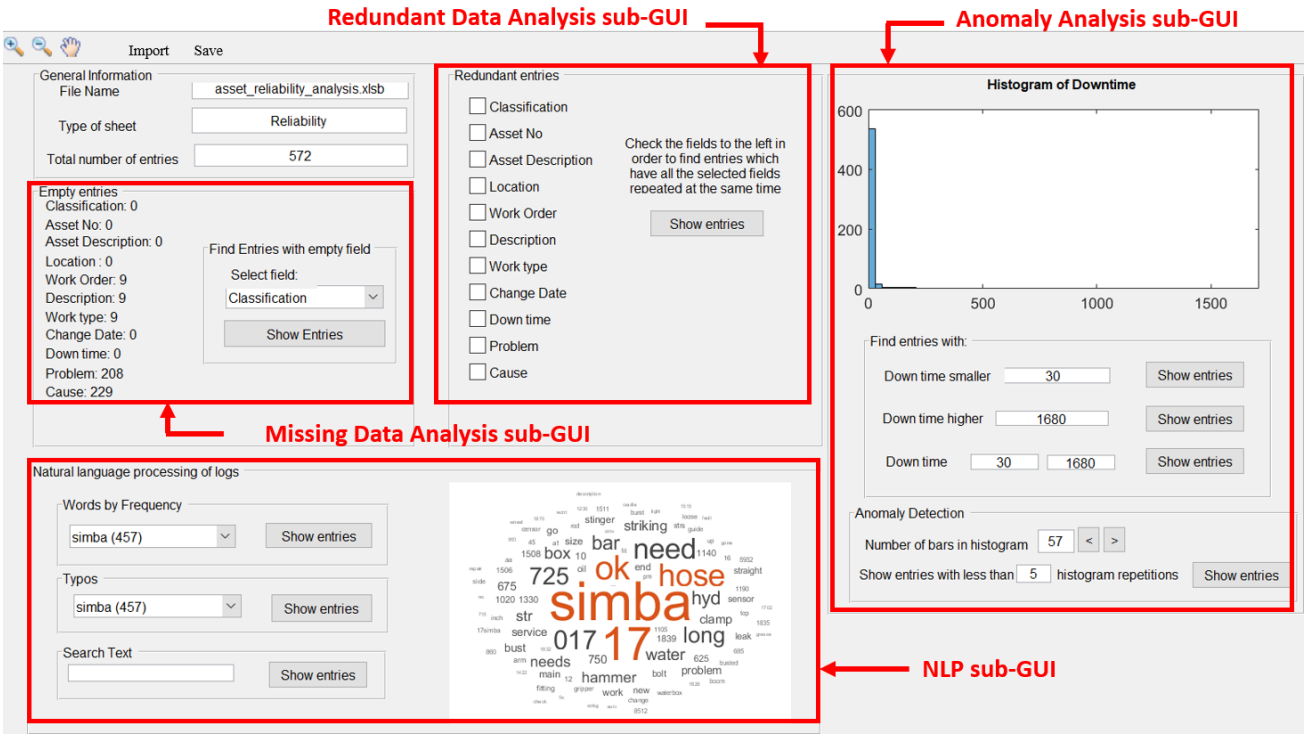


Figure 10. Fit-for-purpose GUI for the Reliability Analysis Format

Investigation: These entries doesn't normally imply that the machine is down. They are the consequence of reporting a Work Order for something which has to be repaired but doesn't prevent the machine from operating. The reparation will normally be performed during a future maintenance stop. When reporting the Work Order the asset is automatically marked down and the worker swiftly marks the asset up again, resulting in a short downtime registered in the system.

DQ Issue: Using the Anomaly Analysis, it was found that there is an entry with more than 2 months downtime.

Investigation: Whilst the root cause of why this entry was generated is unknown, the entry is erroneous because it is from a date before the rig even started operation.

DQ Issue: From 572 entries in the asset reliability spreadsheet, the *Problem* field is missing 208 (36%) times and the *Cause* field is missing 229 (40%) times.

Investigation: The fields *Problem* and *Cause* are structured text where the worker chooses one of many text descriptions. These fields are perceived by the mine staff as very important for reliability analysis. However, the fields are often left empty by the workers. The list of problem/causes has changed recently and therefore there is a data with the old convention and the new convention. Sometimes the problem/cause is unknown. Sometimes workers don't report the problem/cause field because it is not mandatory. Sometimes the problem/cause is not reflected by any of the allowed sections in the list. In order to change the list of problems/causes, company guidelines could be changed. However, this is nearly impossible challenge, since new guidelines/convention has to be approved at a global level for all the sites of the enterprise.

DQ Issue: Using the Missing Data Analysis, it was found that there are 9 entries with missing work order number.

Investigation: Workers can mark assets down and up without the need to create a work order in order to record downtime. These entries are not erroneous.

DQ Issue: The *Description* field in the maintenance logs is populated by abbreviations, jargon, and semantic issues such as fragmented sentences and inconsistencies when referring to the same thing.

Investigation: The quality of maintenance logs is perceived in this study to be of significant importance for Boliden, but not critical. When mine staff were asked about what would make a quality text log, the answer was richness, length and semantic quality. The interviewed staff considered that it is not of interest to limit the ability of the workers to express themselves (e.g. using jargon) as they think adequate in order to describe a work order accurately.

DQ Issue: A number of typos have been found using NLP analysis.

Investigation: The interviewed mine staff don't consider typos a problem, since as data consumers they are going to read the descriptions. It is therefore trivial for humans to understand the correct interpretation of the typo.

DQ Issue: There are numerous abbreviations, with ambiguous meaning in some cases. An example is that "str" is used as abbreviation both for "straight" and "striking". Another notable example is referring to water boxes, which is performed at least in the following ways: "w/box", "box", "waterbox" and "water box.

Investigation: The interviewed staff considered abbreviations as acceptable since texts are going to be interpreted by humans and there is no interest in limiting the ability of the workers to express their actions freely. The mine staff considered however that it is of interest to be aware of such inconsistencies.

6.3. Analysis of the Maintenance Cost Format Sheet

During the analysis of the maintenance cost the following DQ problems were identified:

DQ Issue: The *labor cost* field is always the same constant value of cost/hour multiplied by the time it took to repair. This indicates that the cost/hour rate is always the same.

Investigation: When an operator reports the hours spent, the labor cost is calculated according to a *labor cost/hour* ratio. The used value "*labor cost/hour" in the system* is the same for all the operators regardless of their actual salary and it does not change with time to reflect e.g. yearly salary raises. This calculation is a source of inaccuracy for the labor cost and will influence e.g. Lifecycle Cost models, but such approximation is perceived as "good enough" by the mine staff.

DQ Issue: 13 entries have no associated cost.

Investigation: One of the entries was identified as a human error where the worker reported 0 hours by mistake. The other 12 entries are considered erroneous with unidentified root cause.

DQ Issue: 881 (43%) cost entries have been found that are not connected to a work order. Most of the 881 entries with no associated work orders are related to Material Costs.

Investigation: Two root causes have been identified: i) Workers are allowed to order material and charge the cost to the asset without specifying a work order. This is not considered erroneous data, ii) Service costs with no work order value are associated to the salary payed to the representative from Epiroc which is on-site working on the Simbas. This cost is inaccurate if the data is used for LCC estimation of the asset, since the representative works with the complete fleet and it is unclear when his/her salary is charged to one asset (Simba) or another.

DQ Issue: An entry with abnormally low service cost of 0.01 EUR has been found.

Investigation: It is probably erroneous information. The investigations could not find why this entry was created.

DQ issue: There are two entries with abnormally high service cost (few thousand EUR) and these entries have no associated work order.

Investigation: These two entries are associated to the salary of the Epiroc representative as discussed above.

7. Concluding remarks

Data Quality assessment is of subjective nature and depends on the needs of the data consumer. However, data consumers are not always aware of their needs and have difficulties in formulating requirements. There has however been attempts in literature to define generic DQ properties.

In this paper a generic framework for DQ analysis has been developed based on intrinsic DQ properties. Fit-for purpose software tools result from the instantiation of the framework for a particular purpose. The purpose of the software is to assist the user to investigate DQ issues in a semi-automated way. Once that the DQ issues are identified, posterior investigations have to be undertaken to find evaluate the impact of the issue and find the root cause if necessary.

During a case study regarding the maintenance of Long Hole Drill rigs, the framework has been instantiated in two software tools for processing two different datasets. A generic DQ analysis of both datasets with support from the tool has been performed. The analysis revealed a number of data issues which have been reported in this paper. These data issues were discussed with staff from Tara mine in order to find their root cause and determine their relevance. The tool has enabled a way for the staff of the mine to analyze their maintenance data in order to improve the data quality.

Acknowledgments

The authors acknowledge Boliden Mineral AB for its financial and infrastructural support. The authors are also grateful for valuable input and support from the staff and management of the Tara mine. Boliden Mineral AB and EPIROC AB are acknowledged for their valuable input to the project. Vinnova, the Swedish Energy Agency and Formas are acknowledged for financing this project through the SIP-STRIM programme. The eMaintenanceLAB at Luleå University of Technology is acknowledged for providing the cloud services to deploy the software tool. Additionally, we would like to acknowledge the eMaintenance research team which has provided a conceptual model (i.e. 'AI Factory') that can be used to materialise solutions based in Industrial AI.

8. References

1. Parida A, Kumar U. Maintenance Performance

Measurement Methods, Tools and Application. *Maintworld*. 2009;(1):50-53.

2. Kumar U, Galar D, Parida A, Stenström C, Berges L. Maintenance performance metrics: A state-of-the-art review. *J Qual Maint Eng*. 2013;19(3):233-277. doi:10.1108/JQME-05-2013-0029

3. Lukens S, Naik M, Saetia K, Hu X. Best Practices Framework for Improving Maintenance Data Quality to Enable Asset Performance Analytics. :1-13.

4. Fox C, Levitin A, Redman T. The notion of data and its quality dimensions. *Inf Process Manag*. 1994;30(1):9-19. doi:10.1016/0306-4573(94)90020-5

5. Koronios A, Lin S, Gao J. A data quality model for asset management in engineering organisations. In: *ICIQ*. ; 2005.

6. Mahlamäki K, Niemi A, Jokinen J, Borgman J. Importance of maintenance data quality in extended warranty simulation. *Int J COMADEM*. 2016;19(1):3-10.

7. Strong DM, Lee YW, Wang RY. 10 Potholes in the road to information quality. *Computer (Long Beach Calif)*. 1997;30(8):38-46. doi:10.1109/2.607057

8. Leitheiser RL. Data quality in health care data warehouse environments. *Proc Hawaii Int Conf Syst Sci*. 2001;00(c):152. doi:10.1109/HICSS.2001.926576

9. Brundage MP, Sexton T, Hodkiewicz M, et al. Where do we start? guidance for technology implementation in maintenance management for manufacturing. *ASME 2019 14th Int Manuf Sci Eng Conf MSEC 2019*. 2019;1. doi:10.1115/MSEC2019-2921

10. Naumann F, Rolker C. Assessment Methods for Information Quality Criteria. *Int Conf Inf Qual*. 2000;(June):148-162.

11. Lin S, Gao J, Koronios A, Chanana V. Developing a data quality framework for asset management in engineering organisations. *Int J Inf Qual*. 2007;1(1):100-126. doi:10.1504/IJIQ.2007.013378

12. Huang K-T, Lee YW, Wang RY. *Quality Information and Knowledge*. Prentice Hall PTR; 1998.

13. Tretten P, Karim R. Enhancing the usability of maintenance data management systems. *J Qual Maint Eng*. 2014;20(3):290-303. doi:10.1108/JQME-05-2014-0032

14. Rogova GL, Bosse E. Information quality in information fusion. *2010 13th Int Conf Inf Fusion*. 2014:1-8. doi:10.1109/icif.2010.5711857

15. Wang RY, Strong DM. Beyond accuracy: What data quality means to data consumers. *J Manag Inf Syst*. 1996;12(4):5-33.

16. Redman AVL and TC. Data as a resource: properties, implications, and prescriptions. *Sloan Manag Rev*. 1998;40.1 (Fall(1):p89+.

17. Ebrahimi N, Maasoumi E, Soofi ES. Measuring Informativeness of Data by Entropy and Variance. In: *Advances in Econometrics, Income Distribution and Scientific Methodology*. Springer; 1999:61-77. doi:10.1007/978-3-642-93641-8_5

18. Dekker R, Pinçe Ç, Zuidwijk R, Jalil MN. On the use of installed base information for spare parts logistics: A review of ideas and industry practice. *Int J Prod Econ*. 2013;143(2):536-545. doi:10.1016/j.ijpe.2011.11.025

19. Sandtorv HA, Hokstad P, Thompson DW. Practical experiences with a data collection project: The OREDA project. *Reliab Eng Syst Saf*. 1996;51(2):159-167. doi:10.1016/0951-8320(95)00113-1

20. Stenström C, Aljumaili M, Parida A. Natural language processing of maintenance records data. *Int J COMADEM*. 2015;18(2):33-37.

21. Mukherjee S, Chakraborty A. Automated fault tree generation: Bridging reliability with text mining. *2007 Proc - Annu Reliab Maintainab Symp RAMS*. 2007:83-88. doi:10.1109/RAMS.2007.328096

22. Edwards B, Zatorsky M, Nayak R. Clustering and classification of maintenance longs using text data mining. 2008.

23. Heimerl F, Lohmann S, Lange S, Ertl T. Word cloud explorer: Text analytics based on word clouds. *Proc Annu Hawaii Int Conf Syst Sci*. 2014:1833-1842. doi:10.1109/HICSS.2014.231

24. Bateman S, Gutwin C, Nacenta M. Seeing things in the clouds: The effect of visual features on tag cloud selections. *HYPERTEXT'08 Proc 19th ACM Conf Hypertext Hypermedia, HT'08 with Creat WebScience'08*. 2008;(January):193-202. doi:10.1145/1379092.1379130

25. Rivadeneira AW, Gruen DM, Muller MJ, Millen DR. Getting our head in the clouds: Toward evaluation studies of tagclouds. *Conf Hum Factors Comput Syst - Proc*. 2007;(January):995-998. doi:10.1145/1240624.1240775

26. Navinchandran M, Sharp ME, Brundage MP, Sexton TB. Studies to Predict Maintenance Time Duration and Important Factors From Maintenance Workorder Data. *Phm 2019*. 2019:1-11.

27. Bastos P, Lopes I, Pires L. A maintenance prediction system using data mining techniques. *Lect Notes Eng Comput Sci*. 2012;3:1448-1453.

28. Malave C. Deep hole drilling Cutting forces and balance of tools. 2015.