Universidade Europeia
LAUREATE INTERNATIONAL UNIVERSITIES

Knowledge Engineering Architecture for the Analysis of Organizational Log Data

A software tool for log data analysis

Carlos Jorge Lameira Ribeiro Teixeira

Informática para a Gestão

Advisors:

Ph.D. José Braga de Vasconcelos

July 2018

Student grades

Ano Letivo	Período	Código	Disciplina	Nota	Classificação	Tipo	ECTS	Ciclo
2016/2017	1º Semestre	10181000	Metodologia de Investigação	14	Aprovado	Normal	6	2º
2016/2017	1º Semestre	10181016	Gestão Estratégica	17	Aprovado	Normal	6	2º
2016/2017	1º Semestre	10181040	Engenharia de Software	16	Aprovado	Normal	6	2º
2016/2017	1º Semestre	10181046	Avaliação da Qualidade dos Sistemas de Informação	16	Aprovado	Normal	6	2º
2016/2017	1º Semestre	10181057	Sistemas de Informação Distribuídos	18	Aprovado	Normal	6	2º
2016/2017	2º Semestre	10181014	Análise de Dados	17	Aprovado	Normal	6	2º
2016/2017	2º Semestre	10181079	Segurança nos Sistemas Informáticos	14	Aprovado	Normal	6	2º
2016/2017	2º Semestre	10181162	Gestão de Serviços de Tecnologias de Informação	15	Aprovado	Normal	6	2º
2016/2017	2º Semestre	10181163	Sociedade e Computação	17	Aprovado	Normal	6	2º
2016/2017	2º Semestre	10181165	Análise e Modelação de Processos	17	Aprovado	Normal	6	2º
2017/2018	1º Semestre	10181164	Arquiteturas de Sistemas Empresariais	17	Aprovado	Normal	6	2º
2017/2018	1º Semestre	10181031	Sistemas de Informação e Suporte à Decisão	19	Aprovado	Normal	6	2º
2017/2018	1º Semestre	10181067	Projeto de Dissertação / Estágio Curricular	16	Aprovado	Normal	12	2º
2017/2018	1º Semestre	10181045	Sistemas Estratégicos e Tecnologias Emergentes	18	Aprovado	Normal	6	2º

Curse average grade: 16.59

Phone: 936587217
E-mail: cjteixeira@gmail.com
Thesis type: Academic Research Dissertation

Self-presentation summary

Working since 1996 in various technological areas: financial sector, telecommunications, and general computer trade. Followed the evolution from MS-DOS to the present day - Windows 10. The first job was doing PC's assembly, testing and screening failures, evolving into other areas of computer science like networks and communications, Windows administration. In 1999 there were opportunity to work with large systems as administrator and monitoring network using Tivoli and OpenView that involve Sun Unix, Linux Redhat and Windows. Programming in dot net appeared in 2007 in a university stage that contributed to the completion of the degree in Engineering, giving rise to future work in database and programming including Informix 4GL, MySQL, SQL Server, Oracle. Visual Basic,VBA,C#, php, javascript,jason,html.

Table of Contents

List of tables

List of figures

Abbreviations

AI	Artificial Intelligence
App	Application software
ASCII	American Standard Code for Information Interchange
B2B	Business To Business
CKM	Customer Knowledge Management
CRM	Customer Relationship Management
DB	Database
DE	Development Environment
DM	Data mining
DML	Data Manipulation Language
ELF	Extended Log Format (W3C)
HR	Human Resources
IE	Integration Environment
IIS	Internet Information Services
KD	Knowledge Discovery
KDD	Knowledge Discovery Database
KM	Knowledge Management
KMA	Knowledge Management Architecture
KMS	Knowledge Management System
KMSA	Knowledge Management System Architecture
ML	Machine Learning
NCSA	Common Log File Format
OS	Operating Systems
SSAS	SQL Server Analysis Services
SSIS	SQL Server Integration Services
SWEBOK	Software Engineering Body of Knowledge
W3C	World Wide Web Consortium

Abstract

Organisation log data are generate by software and can provide help to maintenance team addressing issues reported by the client. Once an application is in production, there are defects and other issues that need to be handled. This is also cause by customisation and maintenance of the software. That can compromise software integrity and functionality. This happening in production environment which the maintenance team don't have access becomes a difficult to resolve. The issue must be handling in development environment which causes a condition to understand the problem in order to be able to fix it. To help with this, using log data from production to trace actions that occur of the issue. The log data doesn't contain any of private data; it only contains actions events as a result of software usage. The main objective of this thesis work is to build a framework for an automatic log analyser to assist maintenance team addressing software issues. This framework also provides a knowledge management system allowing registering tacit experience into explicit knowledge. A prototype was developed to produce metrics and make a proof of this framework. This was done on a real environment and is related to a software migration project which means transferring data between databases that holds company business.

Key words: Knowledge management; log files; log data; data mining; software issue

1. Introduction

Technology has brought about many new ways to think, to act and to live. This has been a revolution created by us humans in order to make life better. At this point, there is no going back; computers have come to stay. In this computer age, the processes of information, communication and decision making are completely different now. All of them have speed up faster than we can handle. The world has become a global (digital) village.

Young people have changed the world with their ideas and computers, mainly because the emergence of the Internet has led to the second technological revolution: the birth of a new kind of business. This has led to easier access to knowledge and a faster way to communicate with other people. This accelerates everything that we do, including decision making. Our lives have definitely changed, with us being capable of performing tasks that were never possible before. Things like social services, home banking, online gaming and real-time news offer examples of this new reality that has brought many benefits, but also many concerns.

1.1 Motivation

Software development has resulted in delivery of a product that meets client requirements. Once in production, however, there may be defects that were not detected in the test phase. This is due to characteristics of the production environment that differ from other contexts. As mentioned in Swebok, 'Software maintenance is defined as the totality of activities required to provide cost-effective support to software' [1].

The customisation and maintenance of a software application can compromise its integrity and functionality, resulting in issues being reported by the client. To handle this, there are teams specialising in performing software maintenance, and they work with many different environments. It is common to have three or more such environments. In a company, which focusses on work, there is one for development, a second for testing, a third for quality control and a fourth for the production environment.

Most of these issues are reported by the client while using the software in production, and to fix those issues that occurred in the production environment, they have to be replicated in other environments. This is necessary in order to understand the problem and provide corrections. Replicating the problem is not easy to accomplish, however, because the original data are not available (only test data are available), and that is when a look into log files can help to understand the actions that was performed and the errors that resulted from it.

Quality is essential for making sure that the software only does it was meant to do. If detecting and correcting software bugs or other issues were to become automatic, then it would be possible to develop more trustworthy technology and better balance its cost effectiveness. This is a part of software maintenance tools, where existing software is being modified as 'dynamic analysers, which allow the maintainer to trace the execution path of a program' [1].

1.2 Problem definition

The production environment is not accessible from outside of the client's domain because of security or data protection reasons. However, we can access log files from the production environment, because it does not contain business or personal data information, only actions performed and the 'errors' or 'success' messages as the result of those actions.

The client may be aware of some issues that have not been reported yet. Some other issues are not known by the client, but can be known by the software provider. Some that had been reported are awaiting a solution. In all cases, analysing log data on a regular basis helps the provider to be proficient, and therefore improve the software by releasing updates. Making this process automatic allows for faster issue detection and more accurate problem solving. It also helps to provide insights about software functionality for decision support.

Software is not free from bugs (errors in the software programming). These can make an app malfunction or not function at all, and this poses a major concern. An app writes log data, which can be parsed to structured data written into a database table, and then analysed with data mining and analytic tools. These tools make it possible to detect issues and defects even if the user has not noticed such, therefore enabling proactive behaviour to provide corrections. This process can be automatic and capable of processing large log data.

1.3 Goals

The main objective of this thesis is to build a framework for an automatic log analyser to find software bugs. An automatic log data analysis performed on a regular basis can answer this problem. Detecting and reporting issues can save time for the provider's team that performs the maintenance and customisation work. Overall software quality is intended by reducing the number of issues reported by the client. To this end, this paper strives to accomplish the following goals.

- Gauge the quality of software by analyzing number of issues
- Reducing issues by being proactive - correcting problems before client complain
- Add information to Knowledge Management System for easer problem solving
- Elaborate reports for decision support

1.4 Propose approach

This thesis aims to enable the running of a general proposed log analyser together with data-mining tools, thus allowing for knowledge to be developed for decision-making management and support. It is also possible for this solution to aid in software maintenance-handling issues. In order to achieve that, a short description of the process is presented to illustrate how this procedure should work.

The process starts with log production by software applications, and it can be to a text file or database table log. This means that there is no standard or singularly appropriate way to write logs. This creates a diversity of log data types. The aim is to analyse the data and information that come from logs. In order to accomplish this, we need to know what is going to be analysed and the objective of those analyses.

Figure 1 illustrates the idea in a more detailed manner. The logs that are produced can be a text file or data in a database table. They need to be parsed to extract data to a new database table in order to be prepared for data mining and analysis. Then, knowledge can be taken and applied to a knowledge management system (KMS), as well as producing reports.

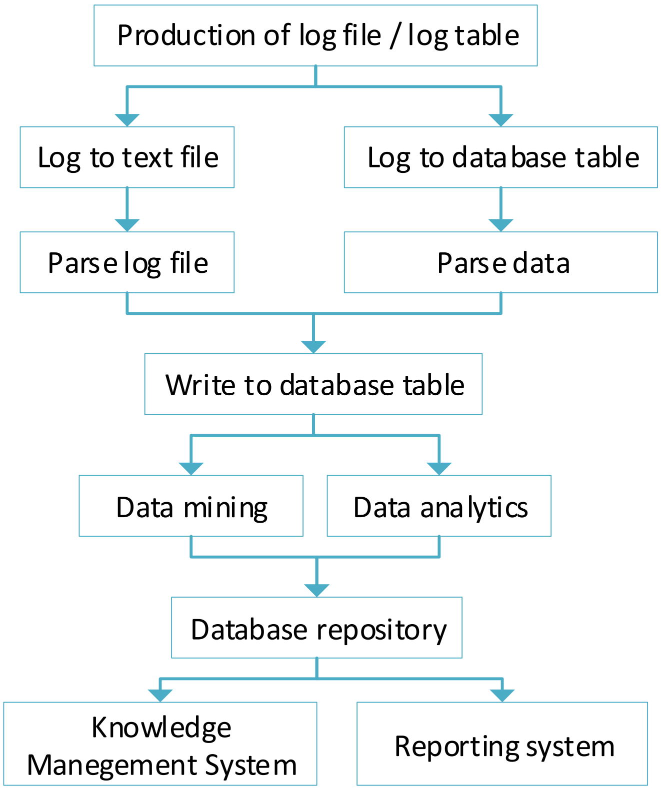

Figure 1 - Detail view of the process for the log analysis

To begin, the log data contain both data that matter to be analysed and data that are not relevant for the intended analysis. This means a selection is necessary to sort what is important to extract. These data should be written into a new database table in order to proceed with mining the data. During the mining stage and analysis, knowledge can be deducted to write into the KMS. The use of artificial intelligence (AI) is a part of the data-mining process, employing proper algorithms to help enhance the analysis for the intended purpose. In this case, the aim is automising the process to reduce human intervention.

This thesis focusses on automatic log data analysis, retrieving information and applying this to a KMS. It is organised as follows: section two reviews the concepts and understanding of the already existing solutions and tools. Furthermore, it also explores the possibility of integrating concepts like AI to enrich the automation process. The next section presents the theoretical background of the proposed methodology. Then, a prototype is presented as a proof of concept, and the obtained results are discussed and analysed. Finally, the conclusions are elaborated and avenues for future work are described.

2. Literature review on Knowledge Management and Engineering

Knowledge management engineering (KME) is the process of designing and creating an infrastructure to hold data for knowledge, and having this available to be searchable and be applied. Understanding what is involved in knowledge management is the first step to being able to select already existing algorithms or to design new algorithms or database structures in order to have a system that people in organisations can make use of.

2.1 Knowledge Management Architecture

The highly competitive business knowledge plays a key role [2] in the organisation, meaning that it can be an asset enabling the company to succeed. Therefore, having a solution to capture, create and manage knowledge should be done properly to be able to make the best of it. Preserving information, learning, solving problems and developing new competencies are ways to improve competiveness and maintain a sustainable business.

Knowledge management architecture (KMA) is a definition [3] of how software should work in order to add value to the organisation and its clients. There are an array of technologies like on-line (cloud) solutions and tools that are developed in-house, but every implementation needs to be appropriate for the kind of information that is appropriate to the organisation's goals.

Knowledge based systems

Knowledge is present in the organisations as either tacit or explicit[4]. Regardless of the type, it is important to record the knowledge into a system that can be consulted and used or re-used so that knowledge can enable a better decision-making process. In addition, this can enable one to solve a problem, analyse trends, benchmark against peers, and if it is possible, to perform all of these tasks efficiently. Then, one may achieve success in implementing a KMA.

Knowledge Acquisition

Once an effective process has been developed, one needs to acquire knowledge [5] which can come from a written document or a presentation that addresses a recurring need. This knowledge can arise from someone figuring out how to solve a common problem. It should be recorded in KMS in a way that others use the process each time a similar situation arises in the future. As explained in the [6] human resources comprise a considerable portion of the organisation that can provide tacit knowledge based on their capabilities and experience.

One way to make use of a KMS is to reuse it. For instance, if there is a database of knowledge available in the organisation, then reusing that to solve a problem accelerates progress and minimises work. If information is well categorised and classified, then it can be consulted efficiently to determine whether solutions for an issue already exist, and this can prevent redundant effort or having to make the same mistake twice. This means that an organisation can learn from its mistakes and gain maturity. However, a culture of trust and openness is essential to feed the KMS. If people in the organisation can freely discuss the failures and success of their work, this will help to convert tacit knowledge into explicit with the aim of adding this to the KMS.

Knowledge management strategies

Organisations can use their management style and culture to create the openness needed to motivate individuals to contribute to developing know-how and converting tacit knowledge into explicit knowledge that can be properly documented. Repositories of information are built up from well-structured knowledge that can be classified and categorised to create taxonomies, thereby making it easy to find the knowledge when needed. As described below, human resources (HR) can contribute to a KMS in three main areas: descriptive, procedure and reasoned knowledge.

Human Resources

- Descriptive knowledge
- Procedure knowledge
- Reasoned knowledge

Figure 2 - Human resources types of contribution for KMS

These areas are the most meaningful contributions for a KMS, as the know-how lies in people, and this needs to be converted into explicit knowledge. Tacit knowledge is developed by experience and observation. It can be applied to activities that will need to be performed in the future. Explicit knowledge is documented, and so can be quantified, written down and communicated to another person. There are techniques available to convert tacit to explicit knowledge. One way is to define precise attributes, characteristics and behaviours in a measured manner. One can pass information to a KMS by converting tacit to quantifiable knowledge.

Descriptive knowledge is also known as declarative knowledge, which is naturally expressed by an individual through sentences or propositions, enabling a person to describe a rule and apply it. This is a description of how a person carries out a task and the person's perceptions and comments concerning it. This can be referred to as theoretical knowledge.

Procedure knowledge involves knowing how to do something by carrying it out. For instance, when a person performs a task, he or she develops knowledge related to this; this hands-on developed know-how is useful to apply in future similar situations. Practice at solving problems and understanding the limitations of a specific solution is tacit knowledge that needs to be converted to explicit.

Reasoned knowledge occurs instinctively and helps one to make a decision regarding the best path to take if there is a previous experience with similar situations. As one is aware of choices made and the extent to which a decision can be trusted, this determines how valid that knowledge is. It is also described as logically thinking through a problem or supporting a theory with due reasoning and justification.

Technology adoption is responsible for a significant portion of cultural knowledge in the organisation. This refers to acceptance of a new product or innovation making changes in how the work is performed. There is a typically reluctance to accepting this, because it requires people to learn new things and develop new competencies. Barriers to its success include the lack of leadership that fails to inspire confidence or problems with infra-structure, such as slow networks, buggy software or incomplete instructions. On the other hand, the success of introducing new technology will increase explicit knowledge that needs to be passed on to other people.

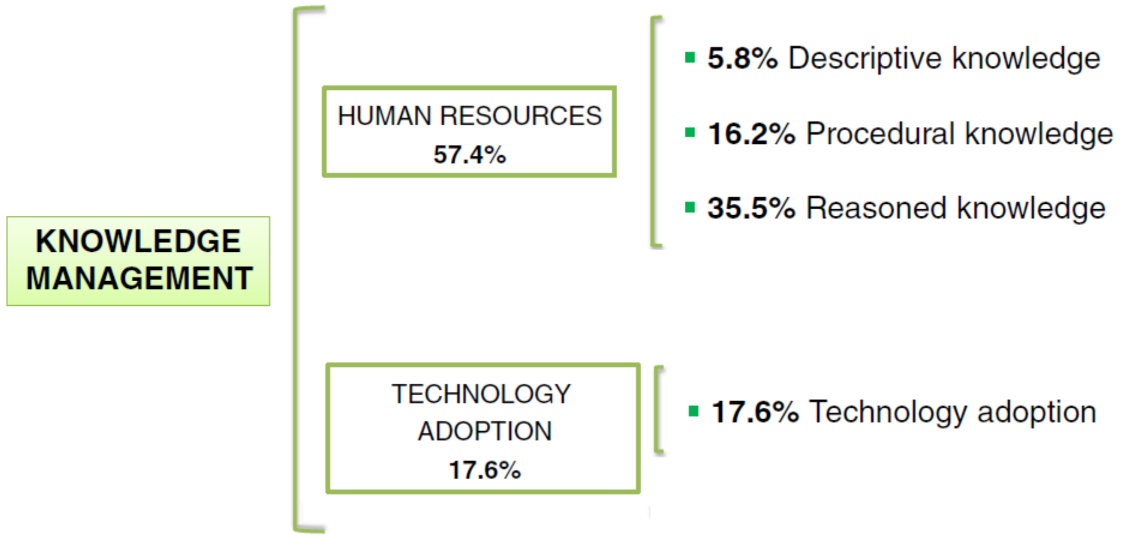

Figure 3 Contribution for KM. Adapted from [6]

The graph above illustrates the two major contributors to knowledge, as described by [6]. A total of 57.4% of the knowledge comes from people in the organisation, and 17.6% comes from technology adoption. This means that a successful KMS is dependent on human resources together with technology. Creating an environment of sharing with full understanding of the competitive consequences, and that is aligned to the organisation's mission, statement of ethics and policies, is a facilitator of knowledge development.

Building employee skills, competencies and career emphasis, as well as creating knowledge, is achievable through the combination of traditional training and development responsibilities. This means that teaching must become part of everyone's job, so the knowledge sharing must be expected, recognised and rewarded. All of the organisation's resources must be used in order to create strategic capabilities and to provide a framework for the mission and vision.

Competitive advantage

As mentioned, 'KM does not only promote high innovative performance, it also develops a firm's competitive advantage' [7]. The competitiveness can manifest in the organisation's finances. Aside from benefiting these finances, there is another way to enhance competitiveness by encompassing innovation, operationalisation, productivity and quality all in one. Innovation can be as simple as modifying processes and enhancing the customer's experience. This will produce value for the client, and therefore, competitive advantage. Operational optimisation and more efficient products and services are also a way to improve customer satisfaction. The [6] says "Both knowledge and technology are considered to be an enterprise's strategic asset and a main source to create competitive advantage".

Customer participation in KMS and customer value

Article [6] states that ‘knowledge and KM processes are sources of value creation for the customer’. In other words, KM implementation helps to create, innovate and problem solve, contributing to customer value and sustainability of the business. One way to make customers participate in KM can be through online forums in which the service provider and the client can participate actively, therefore creating new knowledge from problems that others experience.

The company can create value based on client needs even if they do not realise them. As [6] explained, ‘To determine what the customer wants from a product and/or service also helps the company to make its value proposition’, meaning that a KMS can be used for better understanding of the service or a product and their relation to the client. Opportunities for value propositions emerge and the flexibility to adapt and make changes can lead to competitive advantage, as explained by [6]: ‘These authors argue that innovative firms tend to be more flexible and adaptable to changes’.

Software testing

In order to ensure that the software is complemented with the requisites, it needs to be tested properly. This task is a process that includes a number of activities and resources, and it is necessary because software applications have become more complex. In addition, a product’s success depends on delivering a product that works exactly as intended [8][9]. To achieve this, KM can be a great help. However, testers must be willing to understand and learn new skills, and they must be open to change. Testers furthermore need to be diligent and stay up to date on trends by talking to people and finding out what the organisation is doing. To make it easier, they can consult a knowledge base that has the information they need.

Test case design [8][10] is a plan with cases to test whether they are documented. The results obtained are analysed and registered for whether they passed or failed. The principle underlying this process is the need to identify a representative set of program behaviours [8], White box testing is based on information regarding how the software was designed. Black box testing is based on input and output behaviour. Defect testing reveals software faults. A variety of knowledge is necessary, so it can furthermore be considered a knowledge-intensive process [1]. Testers gain experience from diverse sources. It is necessary to provide support for those tasks by disseminating knowledge for reuse, which helps to increase the test effectiveness and leads to competitive advantages.

Software quality and innovation

The purpose of software quality is the ‘capability of software product to satisfy stated and implied needs under specified conditions … the degree to which a software product meets established requirements’ [1] as stated in SWEBOK. Therefore, a match between the requirements and the functionalities of the software represents quality. To better define the requirements, it is possible to capture knowledge during the cooperation between an organisation and its customers [11]. Customer knowledge management (CKM) is the strategic practice based on forward-looking organisations [11].

There are three main types of CKM, as stated in [11]: Knowledge for customers, which involves products, markets and suppliers applied to satisfy the customer’s needs; knowledge about customers, which is created based on the analysis of customers' historical data and information; and knowledge from customers, which involves the customers' feedback. All of this aids software quality.

As in Figure 4, there are a number of factors that lead to the acquisition and sharing of knowledge [5] and the outcome is product quality and innovation. The measurements of reliability, efficiency, usability, maintainability and functionality are tangible practices to look for when involving the customer in contributing to knowledge management. Therefore, the CKM wields significant influence as a valuable asset for businesses.

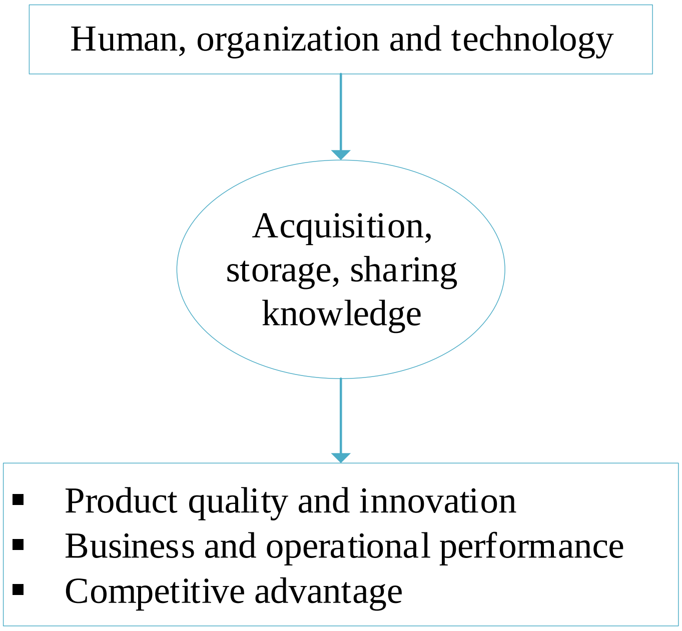

Figure 4 - Organizational acquisition and sharing. Adapted from [5]

Software evolution

Software maintenance is also known as software evolution, and this refers to the period after a computer application (app) is completed. In other words, the next stage is maintenance, and this means modification of requests, reporting and fixing problems, updates and upgrades. All of these activities are needed, and by developing them, a level of abstraction is introduced in this process, which unfortunately means that some knowledge is lost [12]. For the maintenance of a particular system, detailed knowledge and familiarity with the domain is essential [13]. To be able to accomplish such a task, KM helps by recording database information, bug reports and correspondent problem fixing, which can be reused for similar problems.

Another important fact is that a knowledge gap can be created by individuals that leave the software project, resulting in the knowledge not being completely transmitted to new people that enter the project [12]. This supports the theory that KM is necessary to the dynamics of software maintenance and to pass knowledge to the new members of the maintenance team.

Knowledge management architecture model

Developing a KMS [14] requires careful planning before selecting the tools for supporting the knowledge processes. The designed system architecture must be in line with the company culture and business needs. The KMS can be a file in a sever computer or a complex business intelligence [7] system that can use data visualisation and AI. Architecting activities are mainly conducted by an architect in collaboration with a set of stakeholders involved. This will in turn identify the gaps in the existing application of knowledge-based approaches to various architecting activities.

The faster and more effectively we can share ideas and solutions, the better the final product and the service provided to customers will be. This helps when building a team of employees that better serves the purpose of the business, and consequently leads to profit for the shareholders. This is how the return on investment is connected to a KMS.

Knowledge Management System Architecture

There are several KMS architectures that support KM processes. An architecture proposed by Tiwana [15] pointed out that a KMS should possess four major components: repository, collaborative platform, network and culture:

- Repository holds explicated knowledge, such as descriptive and procedure knowledge related to a context. This component acts as the core of the KMS with the objective of storing and retrieving knowledge for re-use.

- Collaborative platform can be a forum in the organisation that supports the discussion of ideas and work. It can include databases, Intranet, informal communications channels and other tech tools.

- Network is a physical and social network that supports communication between computers and between people. Technology adoption can be related to an organisation's computers being connected to each other, allowing people to use tools like Communicator, Skype and in-house developed software. This enables the development of communities of practice (CoP), which are working groups contributing to KM.

- Culture of the organisation is essential for encourage the sharing of knowledge and the transition from tacit to explicit knowledge. As described previously, reason and sharing knowledge should be part of company culture, since human resources possess most of the company knowledge in tacit form. Developing a proper culture unlocks the potential of a KMS.

Knowledge Management System

First, one needs to assess the benefits and faults of the study of a KMS. A decision needs to be made when taking the next step [16]. This next step involves implementing the framework [17] in an organisation. To achieve this implementation, one has to take into consideration the importance of knowledge in the business, which can bring competitiveness, along with the collaboration of all the workers and also the customer. Furthermore, this must be joined with the KMS being either a knowledge giver or a consumer. Each implementation of the KMS is customised to the needs of an organisation; the knowledge to be kept and reused must be in line with business objectives. In order to achieve this, there are some ground rules to provide guidance.

Identifying business problems

The company has to define areas in which they want to build knowledge [6]. For instance, if the business is about maintaining a software application, then possible knowledge to be collected could be the solution to a specific issue that can occur with another customer. Re-using that knowledge will accelerate the solving of customer problems. Identifying areas of knowledge and categorising and classifying the needed information is an essential step. By working closely with employees, it is possible to identify their knowledge needs and encourage them to participate more with this framework.

Knowledge sharing culture

The employees' contribution [18] through their tacit or explicit knowledge is essential, and a way to encourage that culture is having them teach and coach others with the added benefit of being recognised and rewarded. Technical tools, such as a forum in the organisation, can help to achieve a sharing culture. Identifying what knowledge is needed and what knowledge is missing makes it possible to fill the gaps for more effective problem solving.

Evaluate and audit existing knowledge

In order to perform a survey in the organisation for understanding what knowledge already exists and what is needed [19], it is crucial to determine the infrastructure to assembly in order to solve a target problem, or to re-use KM for the daily job functions. Identifying missing knowledge will help to determine what explicit resources currently exist and how they might be classified and categorised for more effective use. Elaborating conceptual models is useful to represent knowledge to organise around topics and areas that contain valid information.

Customizable

Adaptation of a KMS is needed to ensure the company's process and people in order to reflect the needs of knowledge for the efficient and effective re-use of explicit know-how for better decision making and quality of work. Over time, knowledge needs can change, so the KMS must follow this in order to remain useful. It has to be updatable and upgradable to maintain proper competitive advantages.

Knowledge mapping

For query and accuracy of knowledge, a mapping [14] is essential to employ mapping with proper classification and categorisation to taxonomies. This categorisation can be accomplished by the conceptual models and submission of knowledge from employees using a knowledge editor. This process has to be flexible enough to adapt to each organisation's unique environment.

Knowledge mining

As the amount of information increases, large result sets occur. This makes it necessary to employ knowledge categorisation to enable clustering of search results with a knowledge map. Data mining [20] allows for quickly drilling down or mining the most relevant information. In this way, it is possible to feed the KMS with the discovery of knowledge and, at some level, to automate the process of acquiring knowledge for later use, such as in decision support.

2.2 Artificial Intelligence Techniques

Artificial intelligence (AI) can be defined as ‘the science and engineering of imitating, extending and augmenting human intelligence through artificial means and techniques to make intelligent machines’ [21]. What is important in this is to develop computer software that automates a task [22], is able to make decisions and suggests possible solutions for problems. Reasoning is the ability to produce new judgment based on previous knowledge, such as how people resolve a problem based on experience.

Appliance of AI in log data analysis

To analyse data from a log, it is necessary to pre-process or process raw data and to transform this into cleaner data that are suitable to be applied to other algorithms for explaining or solving a problem using more specific methods of data mining.[23]. Nowadays, models by Machine Learning (ML) [21] and Knowledge Discovery (KD) applied[22] to data sets of a specific domain [24] are efficient because those algorithms have been improved**.**

One of the objectives for utilising AI is to automate the process of analysing [25]; This means that a computer algorithm is capable of analysing datasets without human intervention. It knows what to look for and can make decisions in order to produce a final result with information that matters for decision making. Automatic data analysis can deal with big data by producing a valuable result just in time to take action. A more evaluated algorithm can perform predictive analysis and make suggestions for possible actions to take.

It can also be used for automatic data preparation, data reduction, data cleaning and transformation [22]. These steps represent stages for data analysis, one of which can be automated using AI. This allows for handling growing log data, producing the expected results and connecting all stages together, passing from one stage to another. This can furthermore provide input for the KMS, allowing it to register knowledge for later use.

AI in pre-processing data

As already stated pre-processing[26] log data is essential for extracting information and knowledge. There are many ways to do this, as previously explained, and those algorithms can be enhanced by adding AI. This means that it is possible to add intelligence to parsing log entries extracting information and knowledge. In addition, this is more dynamic in adapting to new situations and finding ways to handle unpredicted issues. As stated in [25] this is used as the first step of data processing.

Machine learning

The aim of machine learning [27] algorithms is having a computer program that is capable of learning or improving itself. These systems have the ability to learn from data, using already existing knowledge in its memory to analyse and produce new knowledge. There are many branches of this, such as a decision tree, association rule, artificial neural networks, support vector machines and clustering.

When applying these algorithms to log data entries, knowledge can be produced and stored for later use. Considering the context of this thesis, it also helps to detect software bugs by relating the information and previous knowledge. Once the problem is detected, the algorithm can produce a possible solution that is presented for technical personal evaluation and then taken under consideration.

2.3 Data mining applied to log data

Once log data are pre-proceeded, this means that the file is filtered, unwanted information has been removed and it is ready for data mining. Pattern recognition is one of the fields of data mining, and is the main characteristic for this purpose. The aim is to analyse the log data with a pre-defined rule or keywords and to use basic statistics, such as frequency, to obtain knowledge, and thus to help make decisions.

Appliance of data mining

Log files are in ASCII (American Standard Code for Information Interchange) format [12], which is a numerical representation of a character, also known as a clear text. The collected information represents the user behaviour and the corresponding technical aspects. This can indicate how software application can be improved, either in performance or functionality.

There is a three basic file log format [12]. The common log file format (NCSA) is a standard text file generated by web servers. The extended log format (W3C) allows for customisation by choosing attributes for collecting data. Finally, there is a custom log file format, which is generated by a software application, with developers making their own log file format with attributes that were specified by the requirement document or just information about the app behaviour that might be of use for the developer team to improve the app.

Figure 5 - NCSA log file from IIS (version 6.1) example illustrates the number of logons performed by the app services over time. That information can be filtered to obtain knowledge regarding the basic statistics, allowing for understanding the server's load in a time interval and from a service that is being used. This helps to make decisions concerning the performance or app functionality in order to improve it.

```
#Software: Microsoft Internet Information Services 10.0
#Version: 1.0
#Date: 2018-04-28 14:23:07
#Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User-Agent)
2018-04-28 14:23:07 192.168.1.91 GET / - 80 - 192.168.1.91 avast!+Antivirus - 403 14 0 162
2018-04-28 14:23:09 192.168.1.91 GET /HNAP1/ - 80 - 192.168.1.91 avast!+Antivirus - 404 0 2 0
2018-04-28 14:25:02 ::1 GET / - 80 - ::1 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537
2018-04-28 14:25:02 ::1 GET /favicon.ico - 80 - ::1 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+Appl
```

Figure 5 - NCSA log file from IIS (version 6.1) example

Data mining algorithms

There are many algorithms for data mining[28], but only those that were mostly useful for the scope of this thesis were selected. For data mining, certain principles must be considered [29]: The first is association rules, meaning a relation between values in a dataset. Normally, a threshold is used based on the algorithm adopted. Cluster analysis is a group of data that are similar or possess a common characteristic. Groups of data are different from each other, and by analysing these groups of data, information can be extracted to support decision making. Outliers are data that do not fit any of the clusters. These data are isolated, but depending on the context, these can be meaningful through revealing abnormal behaviour.

Apriori Algorithm was proposed by Agrawal and Srikant in 1994 [30][31]. It is based on association rules that identify the frequent individual items using patterns in a dataset. It is an influential algorithm for mining frequent item sets for Boolean association rules. It utilises the bottom-up approach where frequent subsets are extended one item at a time.

A log file preprocessing technique [32] is used for pre-processing, followed by an analysis of that information. The algorithm pre-scans the file as the first step, and then conducts a second scan and records the frequency of items in a proper structure. Finally, an analysis of that data is performed to identify the most used items. One advantage of this method is its very fast processing speed, as it uses a binary-based approach.

EPLogCleaner [32] analyses web server logs to improve data quality. It uses a time filter to select the log entry that matters for the analysis. Then, it removes irrelevant information, reducing the file size to be analysed. There are a total of three stages in this process to be able to analyse user requests of a site over the Internet.

Event correlation[33] is one way to perform data mining. However, this method requires the log file to be pre-processed in order to extract events within a time window. Then, events should be skipped that have no meaning for the analysis. This is accomplished with a mathematical calculation [14] based on probability. The resulting events are correlated based on mathematic equations indicating, in this case, a probability of malicious activity.

FP-growth algorithm [34] is an association-rule-based data-mining process. It performs better than the Apriori algorithm. Furthermore, it is a recursive algorithm that uses frequent pattern growth to mine. This is performed in two steps: FPtree construction and recursive mining. It needs to scan twice, once for each step.

SFIHtree [35] is an algorithm based on a heuristic rule that creates a root node and then children nodes composed into a tree of nodes and sub-nodes. Once this is done, a second algorithm is used recursively through the generated tree to count all items in a proper manner, resulting in a table of statistics of all items analysed.

Descriptive data mining

Descriptive mining is a process of analyzing data[36] to explain what phenomena occurred, providing insight into the past. The aim is to summarise raw data into understandable information in order to make an assessment or aid in making decisions. This is useful because it allows people to learn from past behaviour. It is accomplished by applying a set of mathematical tools, such as basic arithmetic used to determine averages, counting, percentages or summing data. Basic statistics like frequency are also used. Data aggregation is used to apply the described tools.

Predictive data mining

Predictive mining is aimed at understanding the future[36][37], and striving to determine what can happen. As it provides an estimation of future outcomes, the accuracy is never 100% correct; there is always a margin of error. This forecast uses probabilistic models applied to sets of data and studies their relationships. It can be used to analyse trends and to predict occurrences and similar phenomena in the future.

Prescriptive data mining

Prescriptive analytics[38] analyses data and generates a set of possible actions to be taken. It aims at advising what can be done in order to deal with the issue at hand. It provides a recommended course of action in order to take advantage of either descriptions of the past or predictions made. A combination of techniques, such as business rules, algorithms or machine learning, which are applied to different datasets, can also be used in simulation algorithms to offer a set of options that can be taken.

Pattern recognition

Pattern recognition is present in machine learning, data mining and a knowledge discovery database (KDD). These components are based on algorithms that search datasets to try and match a predefined pattern with ones read from data. This is also known as pattern detection. In order to enable text recognition, one can use a pre-defined sequence of characters that is compared with input data. If a match is found, an action takes place. This may lead to extracting that data or creating a record with a location and occurrence of that detection.

Pattern discovery

Pattern discovery is a process of uncovering patterns from datasets, and this is performed by relating datasets. One technique is discovering specificities of patterns and then evaluating term weights according to the distribution of terms in the discovered patterns. For a given initial extractor, the pattern discovery feature groups input contexts with similar semantics and redraw patterns from them.

Open source mining tools

There are a number of open source data-mining applications that can be used freely. It is important to know them and see how they will fit the proposed work. By using already existing software applications, one saves time when applying the intended developing model for log file analyses.

Weka is a collection of machine learning algorithms for data-mining tasks developed by the University of Waikato in New Zealand, and is now in version three. The algorithms can either be applied directly to a dataset or called from one’s own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules and visualisation. It is also well-suited for developing new machine learning schemes.

Rapid Miner includes standard data-mining features like data cleansing, filtering and clustering. It also features built-in templates and integration with languages like Python and R. It is used for business, commercial applications, research and education. There is a free version with limitations and a pay version with more tools available.

Orange data mining is a Python library that powers Python scripts with its rich compilation of mining and machine learning algorithms for data pre-processing, classification, modelling, regression, clustering and other miscellaneous functions. This comes with a visual programming environment, and its workbench consists of tools for importing data.

Apache Mahout, a project of the Apache Software Foundation, is a library of machine learning algorithms that can aid in clustering, classification and frequent pattern mining. It provides Java libraries for common math operations focussed on linear algebra and statistics.

DataMelt is free mathematics software for scientists, engineers and students. It can be used for numerical computation, statistics, symbolic calculations, and data analysis and visualisation. It can be used with several scripting languages, such as Python/Jython, BeanShell, Groovy, Ruby and Java.

2.4 Log data analysis and architecture

Knowledge management is a necessary process for allowing teams and organisations that worry about innovation to interact freely with each other in order to converge towards the project goal. One way to capture knowledge is through log data that register software application (app) activities on runtime. For instance, when a user utilises an app, it will register all activities on file. Once that information is registered, it is possible to understand problems with that app or the way it is being used. This knowledge can be used to improve app functionality, detect bugs, solve problems or predict errors that might occur in the near future.

Production of log data

Log data are produced by operating systems (OS), databases and software applications. As those apps are being created, log file writing is also created in the app. This can follow the structure of an existing format, like W3C Extended used by Microsoft, or a customised structure could be created instead. The data or information to be written are either established by the requirements or are customised. These data represent the state of the app, the events that occur, performance data or information on software issues, such as errors or faults that occur when using the software.

Software applications (apps) are the source of log data addressed here. This means that apps write log files registering events, error messages, user activities and automatic activities. The structure of that file is not standard. Usually, they are custom-made log data entries, resulting in a custom structure. On the other hand, log data can also be written into a database table. This makes it structured data, allowing for querying that table log. When compared to text files, there are tools available to filter and extract data that text files do not possess.

With single developer apps, the programmer decides what information is written into the logs, while a larger team of programmers specify this in requirements or else develop their own structure that valid across the organisation, and they keep this structure for future software versions. They can also use already existing standards or make their own version of these. There is a diversity of log data productions. Figure 6 - A relation between log data and applications provides an example of a structure of software application that produces log data.

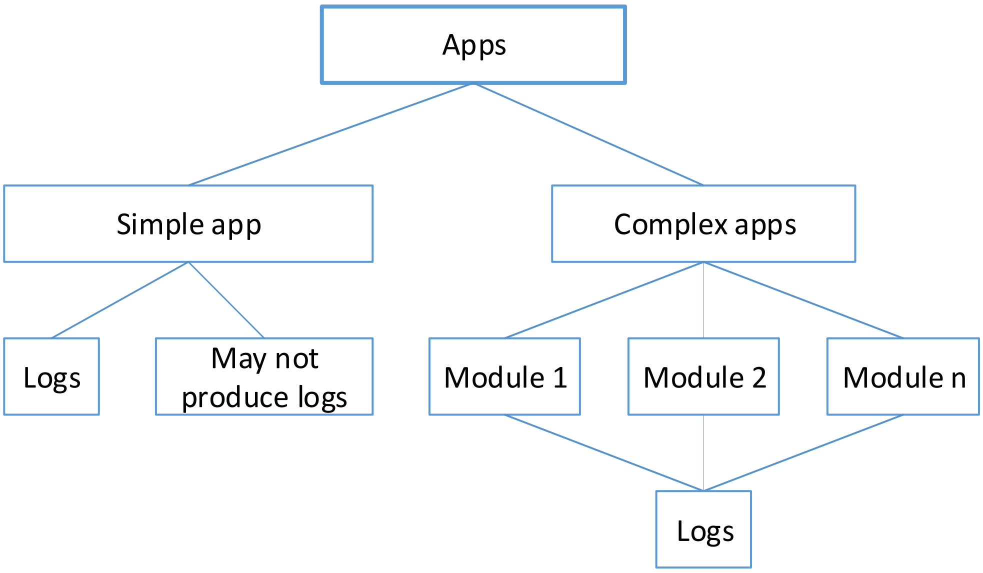

Figure 6 - A relation between log data and applications

Pre-processing algorithms

As stated previously, with log data reflecting user interaction with an app, the data volume can easily reach a gigabyte [40] of data, either in the file or the database table. If the purpose of the analysis possesses a narrow scope, then there is unwanted information in that log. Filtering each log entry is a necessary step that should take place before the analysis. Moreover, the resulting data, which will be smaller in size [14] compared to the original raw data as a result of the filtering, should only include data that matters. For this approach, specific set algorithms were consulted and chosen for data pre-processing.

The information written in a log can be diverse and can include a large amount of similar data that can be grouped together. The data that are needed is dependent on the target evaluation. For instance, if the objective is to know all errors that occur in a given time period, then all other information in a log data is not needed. Extracting only the needed data is the purpose of parsing log entries.

A binary tree technique for pre-processing a log file was proposed in [31]. This method uses two algorithms. The first is a pre-scan of the log file, which will create a table detailing the frequency of the desired information. The second algorithm reads every log entry taking into account the table previously generated. The result is a set of wanted data and their frequency.

Other algorithms like EPLogCleaner [32] consider a date time of log entry to filter information, using a prefix to parse data and write the structure. This dramatically reduces the file size, as explained in [31]. a Furthermore, it also makes it ready for data-mining exploration. An open source approach proposed by [33] uses a configuration file with rules, executes actions that eliminate unwanted data and then applies pattern recognition using regular expression.

Apriori algorithm [41] is a classic mining algorithm that can provide mining association rules and sequential patterns. This was proposed by Agrawal and R.Srikant in 1994, and it reduces the scanning times for discovering frequent sequence patterns. There are many variations of this algorithm, such as AC- Apriori [41] which reduces the runtime and improves efficiency.

Software applications and tools for log data analyses

There are tools and apps to handle log files and even to perform analysis. Some such apps work online through a website, while others have to be installed on a computer. A few apps that offer meaning for this thesis in order to known what already exists and how they perform their analyses are Log Parser, WebLog Expert, Loggly and Splunk.

Log Parser is a powerful, versatile tool that provides queries to text-based log files, files and key data sources on the Windows operating system, such as the event log, the registry, the file system and the active directory. This tool was developed by Microsoft and possesses a command prompt interface, allowing for ‘Select’ queries to be used to log files.

WebLog Expert is a log analyser that provides information about a site’s visitors, including activity statistics, accessed files, paths through the site, information about referring pages and operating systems. The program produces reports that include both text information and charts.

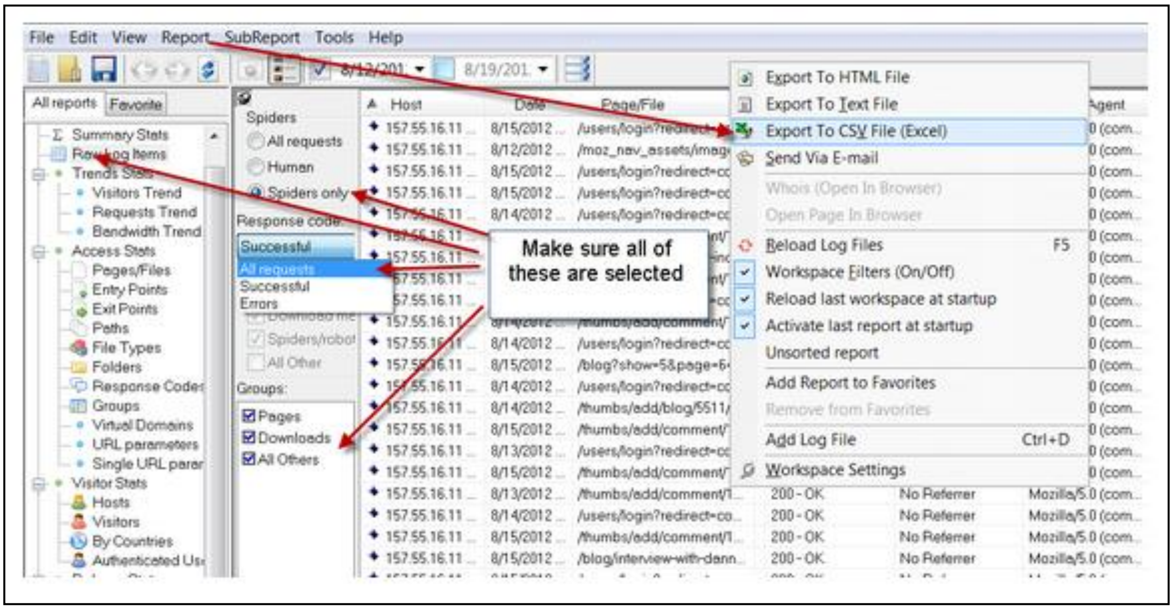

Figure 7 - Example of log file analyzer interface

Loggly can perform full text searches by individual fields and ranges. It employs troubleshooting and data analysis from log files to input information. It is able to trace issues and evaluate interacting components and their correlation. It further analyses and visualises data to answer key questions and performs key performance index reporting.

Splunk aims to provide a deeper understanding of real-time data. This app can read structured and unstructured log files, as well as a multi-line application log data entry. It furthermore possesses built-in reporting with charts and dashboards. It also has the ability to perform ad hoc queries and correlate events and is capable of drill down analyses to reveal spikes and anomalies.

3. Methodology

For the specified problem, a design science methodology is appropriate. This involves a design of artefacts to solve observed problems and make research contributions to communicate the results to an appropriate audience. One needs to infer the objectives of a solution of what is possible and feasible. The objectives are qualitative, including a description of how a new artefact is expected to support solutions to a problem.

3.1 Design science methodology

There are six steps [42][43] for this methodology as illustrated in Figure 8 - Design science research model. Adapted from **[17]**. They represent the entire process with a possibility of several iterations, allowing for attempts to improve the solution. The iteration is intended to understand what can be improved compared to previous iterations. This process of continuous enhancement leads to a functional solution that achieves the intended goals.

.

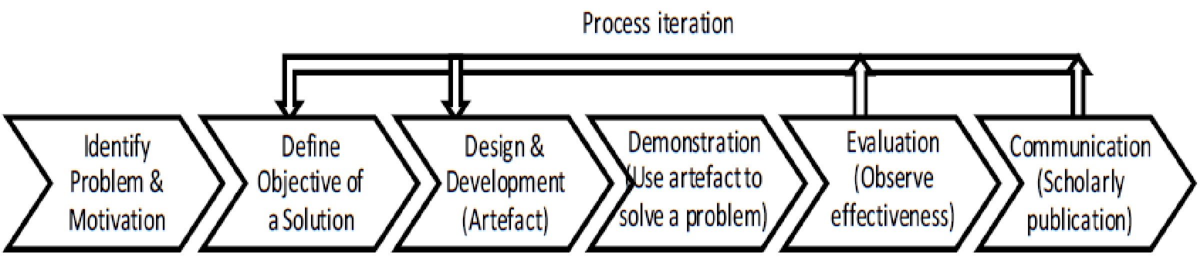

Figure 8 - Design science research model. Adapted from [17]

Identify problem and motivation helps to define the specific research and justify the value of a solution. The utility is to develop an artefact in order to provide a solution, and it further helps to automise the problem conceptually, allowing for the complexity involved to be understood. The audience can be motivated for this research and expect a solution with its results. Certain resources are needed, namely, knowledge about the state of the problem and the importance of its solution.

Define the objective of a solution involves relaying the problem definition and knowledge to infer qualitative objectives describing how a new artefact is expected to support solutions to a problem. This is performed rationally for the problem definition in order to redefine it, if needed, in accordance with the iteration process involved.

Design and development involves constructing a model and methods. This includes the purpose of the functionality and its architecture in order to create the actual artefact. Knowledge and theory are necessary to create a solution to be demonstrated and to draw conclusions for how to redesign, if needed, as part of the iteration process.

Demonstration is using the artefact to solve the problem. This involves experimentation, simulation, case study and proof. Resources are required for the demonstration, and this means that one needs to have effective knowledge concerning how to use the artefact to solve the problem that was defined and designed in earlier steps. A number of demonstrations may be needed, because the results obtained may lead to another iteration.

Evaluation describes observing and measuring how well the artefact supports a solution to the problem. This involves comparing the objectives of a solution to the actual observed results. It requires relevant metrics and analysis. It can be a comparison of the artefact functionality with the solution objectives. This can be repeated a number of times as part of the iteration process.

Communication is about explaining the problem, its importance and its utility. The design and effectiveness has to be communicated to audiences, such as practising professionals. Communication requires knowledge of the culture of the target audiences. One way to communicate a solution to a problem is through a research paper or an expositive method in a seminar.

3.2 Work related to methodology

The problem and motivation was already introduced in the first chapter. Through a more technical approach, the problem can be identified as just-in-time problem solving with analyses from log data of the application activity. This means, during the maintenance phase of the software life cycle, adding a tool to help the maintenance team in their job.

The objectives involve measuring software quality by reducing issues and better aligning with the business activity. Further objectives are to help maintenance team to be proactive by analysing production logs, identifying problems and providing solutions before clients complain. This data analysis can be used to feed a KMS for later reference should a problem reoccur. Having all of this recorded on a database makes it possible to produce reports to help make decisions about business.

An architecture design that allows for the solution to be developed is intended. This architecture will provide a general vision of the solution and how it works. Then, analysing each part in more detail will provide an understanding of how to implement this solution. A log data analyses architecture is intended to enable understanding of how this will work. Then, defining details of that architecture and being able to specify the requirements for the software prototyping should make the development possible.

After a software prototype is complete, then it is possible to execute and obtain results. These are the metrics that can be used to illustrate how this proposed solution helps to solve the problem. Analyses of log data will be used to assess this architecture and to determine how much it helps in maintaining the software.

Understanding the improvements that can be made to this system is a part of the iteration process. To this end, the results of the execution of this prototype should be evaluated. This evaluation makes it possible to know whether this solution helps to achieve the intended goals. It also helps to make improvements to this architecture. Finally, a conclusion can be drawn that can be prepared to be communicated with the audience.

The aim of this communication is to explain the problem, the motivation and the goals, and then to present the software architecture. Once the audience understands this, then the software prototype can be presented and how it works can be demonstrated. This will produce results that can be explained and discussed with the audience. Finally, an assessment can be made from the actual produced results.

4. Early prototyping software

A software project is the way to probe this architecture. Therefore, developing a prototype will provide metrics that can be analysed to prove its value. Prototyping allows the audience to evaluate proposals and try them before implementation. It also helps to understand the requirements. It is intended to increase audiences' involvement in the proposed solution before its implementation.

4.1 Solution Requirements

Solution requirements for log data analysis are to specify the conditions and capabilities required in order to implement the solution and provide clarity in its implementation. Functional requirements define the specific behaviours, responses, information, rules or operations of a solution. They outline the functionality that the solution will support and information or data that will be managed. Non-functional requirements specify the environment in which a solution is intended to operate. They describe conditions it must meet for usability and reliability.

Functional requirements

- Read and record data into the database for data mining and the KMS.
- Read log data and filter the information saved to a new database table.
- Apply data-mining techniques for providing information and knowledge.
- Register knowledge into the database for later consultation.
- Provide reports for decision support.

4.2 Solution Design

To define a solution from the above identified problem, an overall architecture needs to be defined. This will help to identify all parts of the software to be developed and also exemplifies how all parts come together. Figure 9 - Architecture of the propose solution illustrates a representation of that architecture. It is composed of four layer modules, each of which performs its own independent function. A controller named ‘data analysis management’ is used to control all modules. This controller ensures that each module is executed in a proper order to obtain the expected result.

Data Layer is where all data are stored. This includes log data generated from the software application, which is the original raw data. The knowledge base corresponds to a database that stores knowledge obtained from this system. Analysis data is also a database containing the results of the analysis performed on the log data. The report data is a database that generates reports after the analysis.

Service Layer is responsible for pre-processing and mining data. This layer includes two modules, one for filtering data from raw log data, producing a new file, and the second module mines that new file. The results of that mining are registered in a database named ‘analysis data’.

Business Layer corresponds to processing information for the business. It is composed of two modules. One is decision support, which includes the KMS. The other module is a report service, which aims to produce reports on the data and knowledge recorded on the database as a result of this system.

Presentation Layer is intended to help the user interact with this system, making it easier to be used. This is aimed to work on a web browser and to execute all processes just by selecting log data to be analysed and waiting for the result. Then, the user can generate reports based on available information. That report is to be used as decision support.

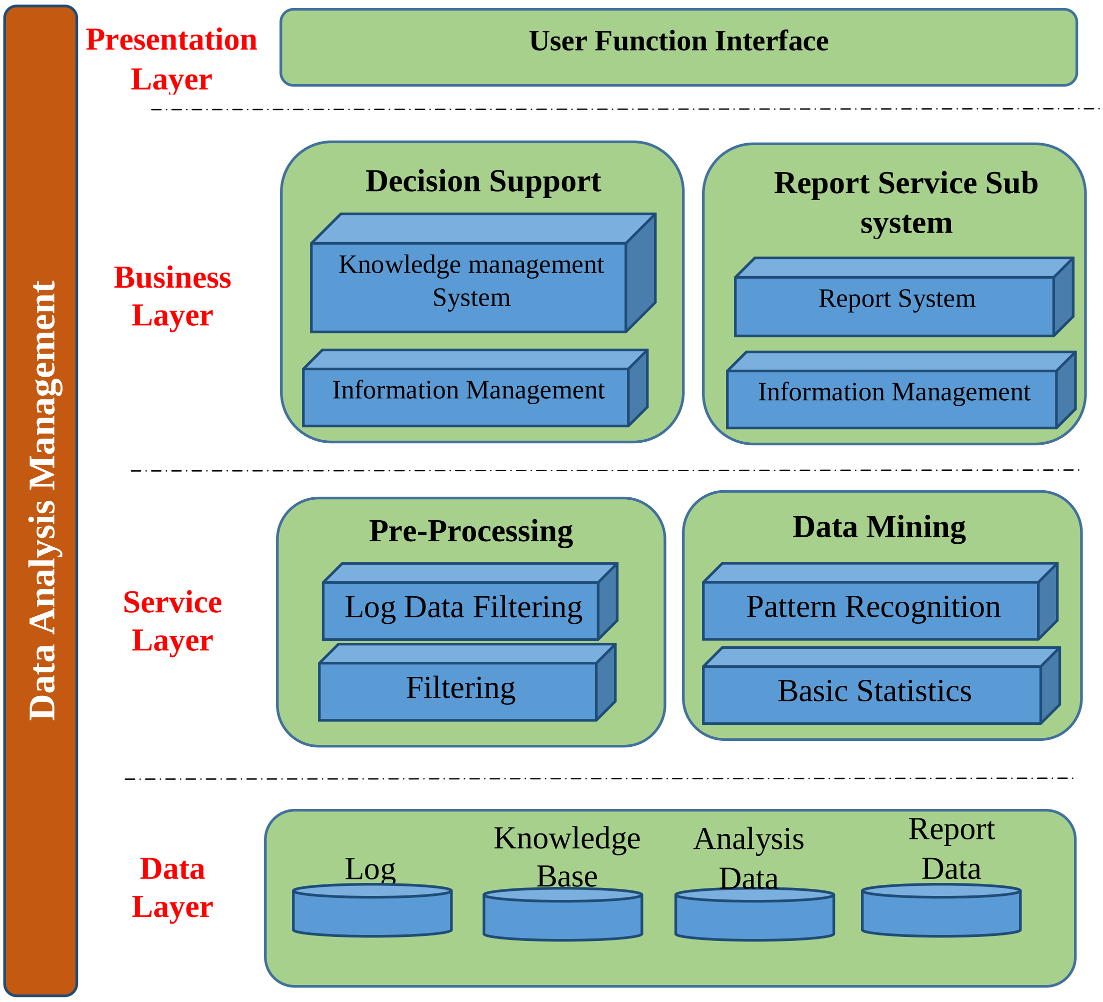

Figure 9 - Architecture of the propose solution

This system integrates a complete set of analysis procedures to be performed regularly in order to help business management and development. It is highly dynamic, because each module is independent and can be improved separately. This means that the data-mining module and its algorithms can be improved without provoking changes in other modules.

4.3 Model Engineering for parsing log data

The first step for log analysing is pre-processing the raw log data. This means that log data are produced by the software application, and then they are available to be analysed. An overview of this process is illustrated in Figure 10. The input is all log data to be processed, which is then filtered, and as the output, a new file is generated with data to be analysed. This new file will be used in the next step, which is data mining.

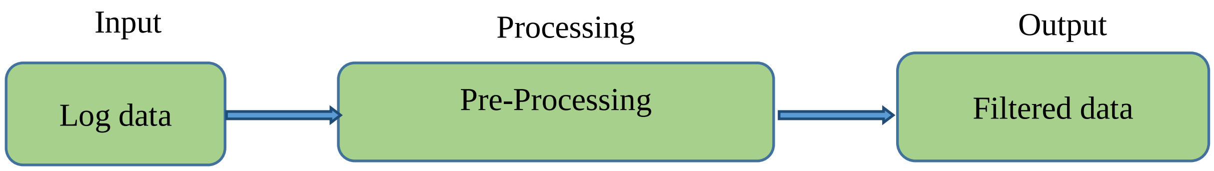

Figure 10 - Parse of log data process

Certain rules are applied to this process. The filtering requires knowing exactly what the objective of that analysis is. If the log data are in a file, the algorithm used for filtering needs to be in compliance with the intentional data extraction. If it is a log table, this must be queried to extract data or other tools must be used to accomplish this, and the needed data must be selected. This can be applied to all log data or all log tables, or it could be designed for just one unique situation. A more detailed description of this process is provided in Figure 11 – The preparation of log data to be analyse. The parsing log data entries activity corresponds to reading the file or a log table, then detecting each log entry and parsing that entry line by line, applying requisite criteria for the purposes of the analysis. Then, data are extracted and written in a new database table to be used in the data-mining process.

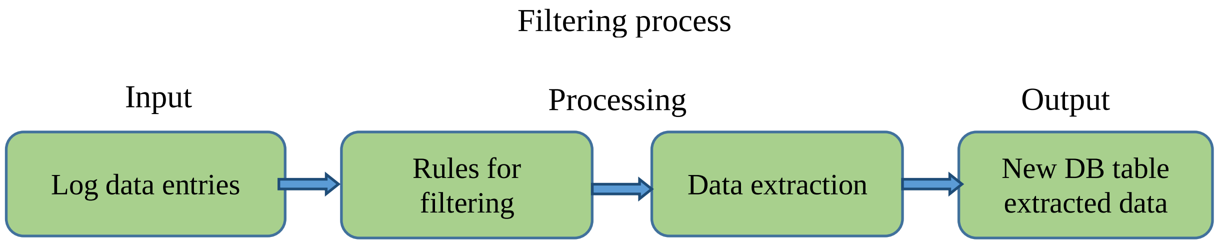

Figure 11 – The preparation of log data to be analyse

Another way to see this process is through pseudo-code, as illustrated next. The algorithm starts by creating a new database table, which will be the output of this process. Then, the rules of data extraction are set. The way of setting these rules differs if the log data are a file or a table. The log data to be processed should be defined, along with the scope of log entries. After all the settings, the process of parsing begins with a double loop; the outside loop cycles through all entries, and the inner loop cycles through various actions of a single entry. The entry actions are parsed, the rules are applied, and the data are extracted and written into a new database table.

```
Create a new database table with extra columns to hold the extracted data
Set rules to be used in order to filter the wanted data
Set a log file or database log table to be processed

While log entries are processed
        For each entry
                Parse log entry
                Apply rules
                Extract data
```

```
                Write the extracted data into database table
        End log entry
End log entries

Process the database table with extra columns
        Insert data to aid analysis of clustering algorithms
        Insert data to help mining algorithms
        Insert data to help forecast algorithms
End process
```

At the end of this process, a table with data from the original log is structured and ready to be analysed. It is important that rules are well defined in order to obtain the expected result, namely, the appropriate data for analysis. The next step is to apply data mining and extract knowledge from it, as described in the following section.

4.4 Model Engineering for Data mining

For the mining process, explaining in Figure 12, the file produced in earlier steps of this architecture should be used as input. This file contains a set of data that were filtered. This is the data to be analysed with data-mining algorithms. These algorithms will perform basic statistics and counting. As for output, a new file with results of that mining is produced and available for the next step in this architecture.

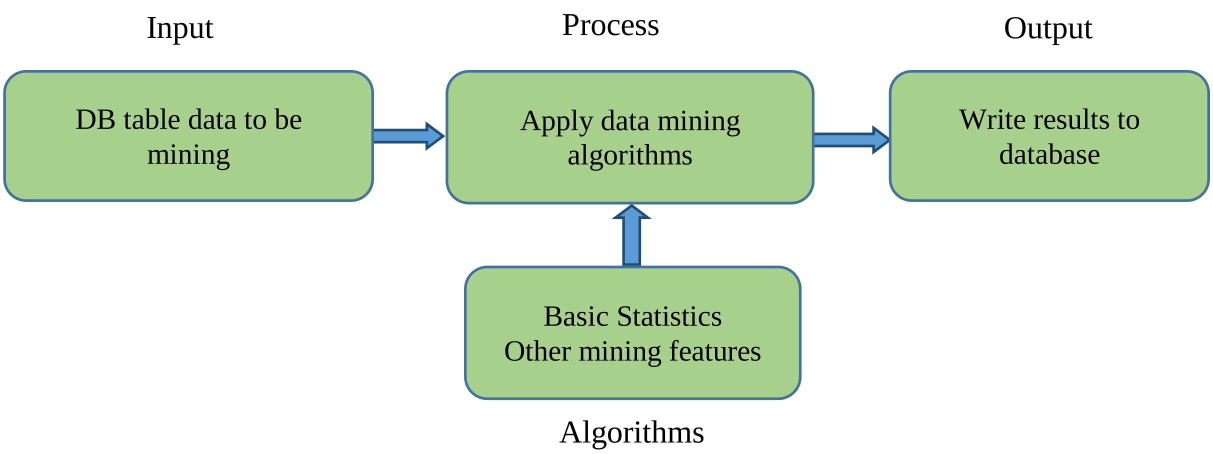

Figure 12 - Overview for data mining process

This mining algorithm can solve only numerical computations, but tasks such as searching for relationships between variables, clustering, classifying, regression and summarising are achieved using AI algorithms. Adding AI to these functionalities also allows for autonomy of the process with less human intervention.

4.5 Model engineering for knowledge management system

The KMS is mainly a database that records and searches information. This means that information derived from the data-mining process has to be recorded into the database and then associated with other information in order to generate knowledge. Relationships that exist between tables in a database allow for crossing information, and that can produce knowledge.

The KMS represents an analysis and reporting tool. The analysis of log data allows for tracking usage of the app, while the reporting helps management to understand the relationships between people and the app being analysed. By knowing how people interact with app, it is possible to build connections in such a way that employees can benefit from what others are doing. In addition, it should be noted that the KMS utilises it to find and recommend procedures or documents to address an issue. This is the effect of knowledge in an organisation.

Figure 13 illustrates the model of the KMS. The input data is the file produced by the data-mining process. This is the analysis of the log data that contains information that needs to be crossed with already existing information in order to produce knowledge. The process of knowledge acquisition utilises the library algorithms to cross information and gain knowledge. Then, this is written into a database to be available to and consulted by other users.

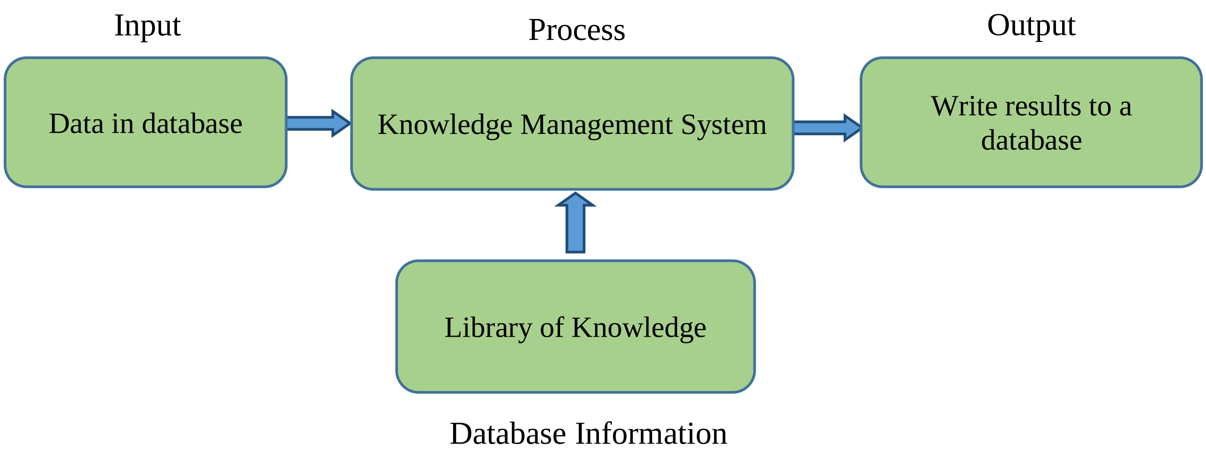

Figure 13 - Process of Knowledge Management System (KMS)

It is important to retain knowledge that comes from interaction with clients in a way that sharing relevant knowledge is possible. Not only does this allow better handling of clients' problems, but it also provides valid information for decision making. The 'library of knowledge' is the information already existing in the database, which can be crossed with the new input information.

4.6 Model Engineering for Artificial Intelligence

Artificial intelligence techniques become easier to use due to the library of programming languages available from Microsoft AI, Google AI, Open AI or PHP-ML, as examples of free usage to help software applications better adept to users' needs. The AI provides an agent that helps in data mining and KM. These AI algorithms can be integrated into the processes mentioned previously, as illustrated in Figure 14.

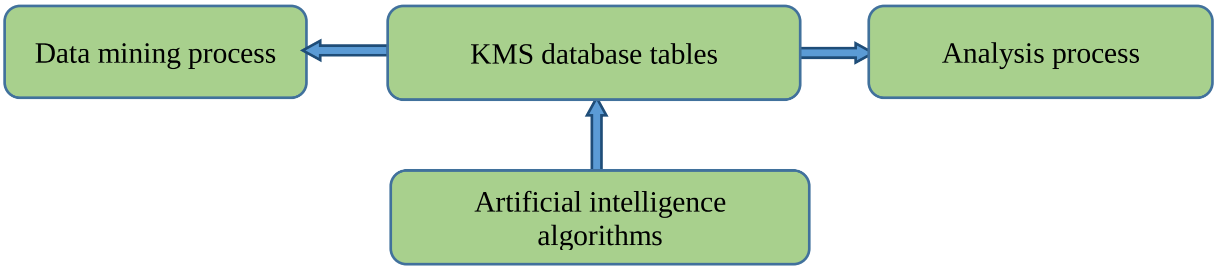

Figure 14 - AI model to aid KMS and data mining

4.7 Software Prototype Overview

This prototype is a demonstration of the solution presented in this thesis, and is therefore not a full functional product. It used tools that allow for processing log data and produced results with metrics that allow for drawing conclusions. The modules of the architecture and how they were prototyped is described and discussed in 4.8 and 4.9.

Prototype Components

To produce this prototype, we used tools that were already available, and when put together, they allow for testing individual parts of the proposed architecture. The components are SQL server 2014 with SSIS and SSAS services and the original data log tables from the migration process, which occurred in two different environments. In addition, free editions of the Power BI software were used to view the results as on way to functions as interface for the final user.

Resources

The log data resources are from a real work environment's customer resource management (CRM) application oriented for car loans and car buying. These involve financial leasing, operation leasing and renting as the core business for this software. The log information tables were designed and produced for a migration process, and they are used in this prototyping.

Perform data transformation

The process starts with original log data in the SQL table. Then, a pre-process is conducted using SSIS by reading the entire log table and filtering the data line by line. Queries were used in this process. The SSIS created another table with log data to be analysed and keep track of log entries for future reference. The result is a new log table containing the data to be analysed. The next process is the data mining, which was performed by SSAS together with queries to the database, producing the metrics and knowledge about the migration process. The next step was registering that knowledge to KMS tables, and that was performed with DML commands in this prototype. Finally, reporting was conducted by querying and used Excel for data extracted from the database.

4.8 Prototype Evaluation

This prototype is the proof of concept of the architecture described. This prototype will process log data and produce metrics with scientific value. A scenario is described and a step-by-step explanation is provided to illustrate how changes are made through the architecture with examples and figures. Then, metrics are elaborated, and they are represented as tables and graphics. An explanation for each result is further provided. This helps to understand the overall process and draw conclusions.

Scenario

The scenario is based on a real working environment and relates to my current work. This company utilises CRM, which is a web-based software application designed to handle all aspects of financial credit related to automobile acquisition. For example, a client that wants to buy a car with a loan or a lease asks for a credit simulation. A proposal is subsequently generated, and if the client accepts that proposal, then risk analysis is evaluated. If all risk analysis checks out, then a contract is created. If the client accepts all conditions and signs, then the schedule for the rent payments are agreed upon, moving the process to the next stage. The vehicle is handed over to the client and the contract operation phase begins, which involves the accounting, cash flow, rent payment and post-sale assistance.

From a B2B (business to business) point of view, the software sells to the financial sector. When a company acquires this new software, a process of transferring all data is necessary. This process is called ‘migration’. It means that the client’s old data have to be transferred to this new software. Unfortunately, there are risks involved. The migration is a semi-automatic process that has to be designed, along with the log files or log records. The objective is to generate log information during the execution of data transferring. Everything about the migration should be written into a log. This means the timestamp and number of records. Furthermore, each record should be tagged and successes and errors should be in the log. This is a critical phase, and it must be well controlled in order to make sure that data are properly transferred.

To accomplish this control, log records in a database are used to perform a registration of every change in the process. The expected result is a table with a large number of records in a database. Processing this log information with the proposed architecture is the aim of this prototype. To this end, Microsoft SQL Server Integration Services (SSIS) and SQL Server Analysis Services (SSAS) are used.

There are two log tables: one to register the temporary loading of data, a second one to register the mapping procedure and a third to register the changes made during the migration. By applying the proposed architecture, this prototype employs the third log table, which registers in a verbose manner all changes made to a definitive database. There are two environments where this prototype was tested: call development and integration environment (IE). These are similar environments, but the difference is that one has more restricted access than the other. The data on the IE are consolidated and are closer to production environment.

Pre-analysis step

This is the first step in the architecture, and what is expected is preparing the log records data using SSIS. This involves working with raw data in order to eliminate information that is not important for analysis. Making connections within data helps prepare for the mining process. The result is to have information in a table run through SSAS with all data prepared for analysis.

There are two main log tables with important data for this purpose: ENV_DEV_Log and ENV_INT_Log. Both tables record the execution of migration corresponding to each environment. Other log tables, such as a mapping log table for each environment, record which data were or were not possible to be used for mapping a new system. For this step, just one field was used, namely, Migration_Status. The migration requirements establish rules for this mapping, but there were records that did not match those rules, and thus failed to migrate. These data were merged together into a final log table to be analysed.

Three more fields were created in preparation for the data mining. One of these fields is Flag_error, which is a 0 or 1 of the column Migration_Status. Another field is Accumulated_Errors, which contains the added error reported by the migration process. The third field is Accumulated_Processed_Records, which is the count of records that were migrated regardless of their status.

Data analysis and data mining step

This step involves applying the analysis and mining algorithm available in SSAS. Some metrics were obtained by querying the database, while others were gathered by using algorithm clustering and linear regression. The objective is to obtain metrics, and with those, to obtain knowledge for the KMS. The following elaborates business statements about this migration that are intended to be reported on:

- Performance: understanding the time taken for the migration to be complete, including the human intervention and taking into account the volume of data to be migrated.
- Determine the percentage of errors and success of the migrated data.
- Classify and categorise the most common errors.
- Predict how long it will take and the percentage of success in production systems based on migration performed on other environments.

Knowledge Management System

For this prototype, the KMS is a set of tables in a database that registers algorithms or scripts that solve an issue. This solution can be evaluated by peer's classification, and constructive comments can also be registered. If a solution for a problem is well accepted by other colleagues in a company, and if it has good reviews, then it can be used for other similar problems. The intention is to write into these database tables the issue, to propose a solution with the corresponding source code and to have it available to be voted and commented on in the organisation by the maintenance team. This step involves the following guidelines:

- Evaluate the migration solution or its parts.
- Register the most common problems and the corresponding solution.
- Point out the algorithms used, having them available for other collaborators to evaluate on or to suggest enhancement to it.

Human resources should fill this knowledge about the migration process into the database. The point here is to have a platform that allows for registering problems and solutions that can be searched and applied if a similar issue occurs again. This way, if a solution already exists, the maintenance team just has to search for and use or adapt the solution to the issue in question.

5. Results

This prototyping was conducted in a local environment and a remote IE. The process of data migration in a local environment was executed with a ThinkPad, Lenovo E560, i5 core, while the remote system employed virtual servers with restricted access and unknown hardware. The results of the prototype and corresponding analysis are presented. Furthermore, four questions are answered that discuss the results of the performance and the success of this process of migrating data. A forecasting is further performed for the production environment.

For the migration process to work, it needs to work offline. This means working when both systems involved (old and new software systems) are not available for the users. During the migration process, it is not possible for their customers or internal users to make use of it. The company's business is thus stopped. This migration is divided into six steps, with a human action included between each. This is semi-automatic, because there are six scripts that perform automatic processing of records, and then a member of the maintenance team checks the result of each step before deciding to proceed to the next automatic script or to abort the migration.

Question: How long did the migration take, taking into account the volume of data to be migrated and human intervention?

The intention is to take as little time as possible with a maxim number of records being migrated successfully. The log table being analysed includes a timestamp at both the beginning and end of the migration. Their difference provides the time taken for the entire process to run and is summarised in Table 1.

Table 1 - Performance of migration process

Migration results	Dev Env.	Int. Env.
Time stamp at beginning	2018-05-21 10:48:51.627	2018-05-24 21:06:16.170
Time stamp at the end	2018-05-21 17:01:01.343	2018-05-25 02:56:55.993
Total time	7 hours	5 hours
Total records processed	24524	25044

At the top is the ‘Dev. Env.’ column, which represents the development environment (DE) and ‘Int. Env.’, which represents the IE. The first column illustrates that it took seven hours to process 24524 records, taking into account human interaction. For the second column, it only took five hours to process records. More records were processed in less time because the IE was restricted access servers with improved performance compared to the developing environment.

Question: What is the percentage of errors and success of the migrated data?

This determination utilises a column named Migration_Status in the log table being analysed, which registers whether a record was migrated successfully or not. This field is a signed integer number, with values larger than zero meaning the record was migrated successfully. If it is less than zero, this means there was an error and the record was not migrated. If the value is zero, it means it was not processed, as this is the starting value.

Table 2 - Categorizing error type of migrated data for IE

Migration results	Dev Env.	Int. Env.
Total number of records with success	23210	24275
Total number of records with errors	1314	769
Total records processed	24524	25044
Percentage of success	94.64%	96.93%
Percentage of errors	5.36%	3.07%
Ratio	18:1	31:1

The result was 23210 records migrated successfully for DE against 24275 for IE. As for errors, there were 1314 versus 769 for DE and IE, respectively. This means more successful records were processed in IE. The ratio between success and errors was 18:1 in DE compared to 31:1 for IE. This means that there are 31 successful records to 1 record with errors.

Question: How can the most common errors be categorised and classified?

For this migration process, it was expected that errors would occur. Taking that into consideration, a table of error types was created in order to classify the migration errors, as illustrated in Table 3 - Error classification for the migration process. This enables mitigation or performing a correction for the detected cases. The migrations status column on the log table cross links with this table. All errors are coded with a negative integer number, with a zero value meaning that no error occurred. A column presents an abbreviation of the error type code that helps with filtering, and the last column describes the error type.

Table 3 - Error classification for the migration process

Error Code	Error Type	Description
0	NO_ERROR	There were no errors. A DML command was performed without errors.
-3	SQL_ERROR	Error reported from SQL.
-5	UNKOWN_ERROR	An error that does not match any defined criteria.
-10	UNIQUE_KEY_CONSTRAINT	Violation of UNIQUE KEY CONSTRAINT.
-15	FOREIGN_KEY_CONFLICTED	The integrity of foreign key is not being respected.
-20	NULL_ERROR	Cannot insert the value NULL into column.
-25	CONVERSION_ERROR	A conversion error occurred during execution of an SQL statement.
-30	TRUNCATION_ERROR	A truncation error occurred during execution of an SQL statement.
-35	DDL_ERROR	A runtime error occurred during execution of an SQL DDL statement.
-40	NO_SUCH_TABLE	There was an invalid table reference.
-50	MISMATCH_TYPE	Type of variable is not correct.
-51	REPEATED_NAME	A name already exists in the referenced table.
-52	REPEATED_ABREV	An abbreviation already exists in the referenced table.
-53	REPEATED_TYPE	A type already exists in the referenced table.
-54	REPEATED_NUMBER	A number already exists in the referenced table.
-55	COUNTRY_MANDATORY	A country name is mandatory.
-56	STREET_MANDATORY	A street name is mandatory.
-57	CONTACT_MANDATORY	A contact name is mandatory.
-58	PHONE_MANDATORY	A phone number is mandatory.
-58	EMAIL_MANDATORY	An e-mail is mandatory.
-59	ID_DOC_NUMBER	An ID document number is mandatory.
-60	RATING_TYPE	The rating type is missing.
-61	RATING_VALUE	The rating value is missing.
-62	RATING_DATE	The rating date is missing.
-63	RISK_TYPE	The risk type is missing.
-64	RISK_VALUE	The risk value is missing.
-65	RISK_DATE	The risk date is missing.

Applying cluster mining algorithm in SSAS results in Figure 15. Cluster 1 of which illustrates the successful migrated record, while Cluster 2 illustrates the unsuccessful. The ‘Population All’ column presents the relation between Clusters 1 and 2. The variables column demonstrates the grouping criteria. The row MAP STATUS illustrates that there are three types of statuses, which correspond to the Reason row.

Figure 15 - SSAS Cluster mining related DE

Table 4 - Categorizing error type of migrated data for DE summarizes the Figure 15 that relate to the DE. The ‘Error Type’ column presents the abbreviation code for the kind of errors that occurred. The centre column illustrates the number of records in each error category, of which there are two groups of errors. A total of 1308 errors occurred due to missing data that was mandatory for migrating the record. The original data did not provide what was necessary for this process. There are six errors that were reported by the database. This is related to SQL instructions that failed to execute. Finally, a total of 23210 records migrated successfully and without errors. Adding 1308 plus 6 obtains a total number of errors, which is 1314.

Table 4 - Categorizing error type of migrated data for DE

Error Type	Number of Records	Description
NULL_ERROR	1308	Cannot insert the value NULL into column
SQL_ERROR	6	Error reported from SQL.
NO_ERROR	23210	There were no errors. A DML command was perform without errors

As for IE when cluster mining algorithm is used it results in Figure 16. Cluster 1 illustrates the successful migrated record, while Clusters 2 through 4 present the unsuccessful grouped by type of error. The Reason row presents a description of errors, corresponding to the Map

Status row. There are three types of errors and one code that indicates successful migrated records.

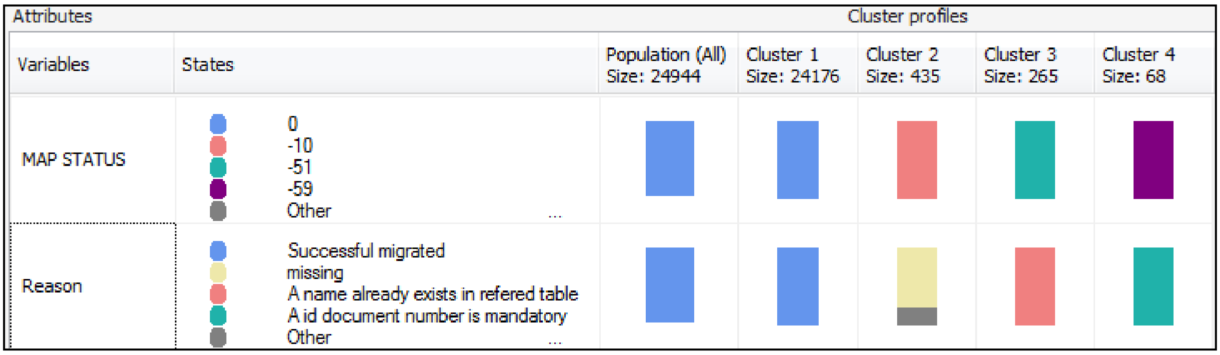

Figure 16 - SSAS Cluster mining related IE

Analysing the IE in Table 5 of errors that occurred had to do with already existing information that could not be duplicated, of which there were 265 cases. A missing identification document was responsible for 68 cases of not migrated data. The SQL errors that occurred related to database constraints, resulting in 435 records failing to migrate. Regarding success, 24176 records were migrated to a new system with no issues.

Table 5 - Categorizing error type of migrated data for IE

Error type	Number of records	Description
ID_DOC_NUMBER	68	An ID document number is mandatory.
REPEATED_NAME	265	A name already exists in the referenced table.
UNIQUE_KEY_CONSTRAINT	435	Violation of UNIQUE KEY CONSTRAINT.
NO_ERROR	24176	There were no errors. A DML command was performed without errors.

There are completely different reasons for errors in the developing and integration environments. The DE NULL values and SQL errors represent all errors that occurred. The same process in IE, the validation of business rules, represents all the errors. The lack of an ID document or a repeated name and the constraint of a unique key all result in the original data being incoherent. Filtering and validating the original data is essential to minimise these errors.

Question: How long it will take and the percentage of success with higher volume of data based on historical data?

Figure 17 – linear regression graph illustrating number of errors over number of records presents a linear regression calculation that predicts the number of errors, taking into consideration the number of records processed. This was performed with SSAS by using log table information and applying the available mining model of linear regression. From the log

table, when imported by SSIS, a column Accumulated_Errors was added, which describes the total number of errors that occurred in each log entry. The column Accumulated_Processed_Records is a continuous variable accumulating the number records processed for each log entry, and independently of the status of that record.

The independent variable is the Accumulated_Processed_Records column while the dependent variable is Accumulated_Errors, since the number of errors that may occur depends on the number of records processed. Each record can be either successfully or not successfully migrated to the new system. The red line in the graph presents the ideal prediction, while the blue dots describe the actual data. The X axis illustrates the scale from the actual data, coming from the accumulated errors column, and the Y axis is the scale of prediction values.

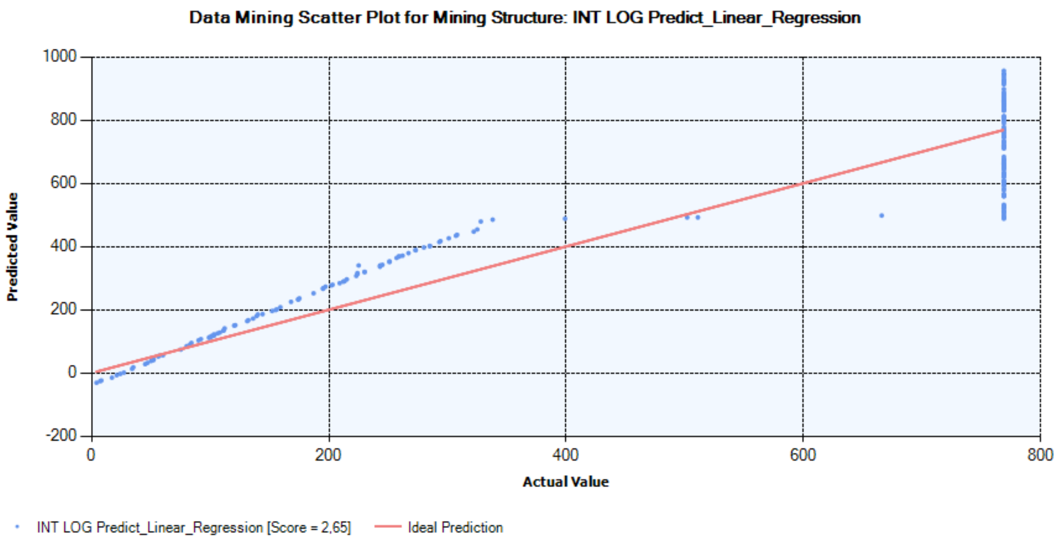

Figure 17 – linear regression graph illustrating number of errors over number of records

In order to predict a future scenario, the SSAS function was parameterised for processing roughly double the number of records. The objective was to forecast the number of errors that might occur in a production environment that possesses more records then the other environments presented here. It was thus set to 50,000 records. A prediction function returned the value of 1967 possible errors, as illustrated in Figure 18. This is what is expected when this process is executed on production servers.

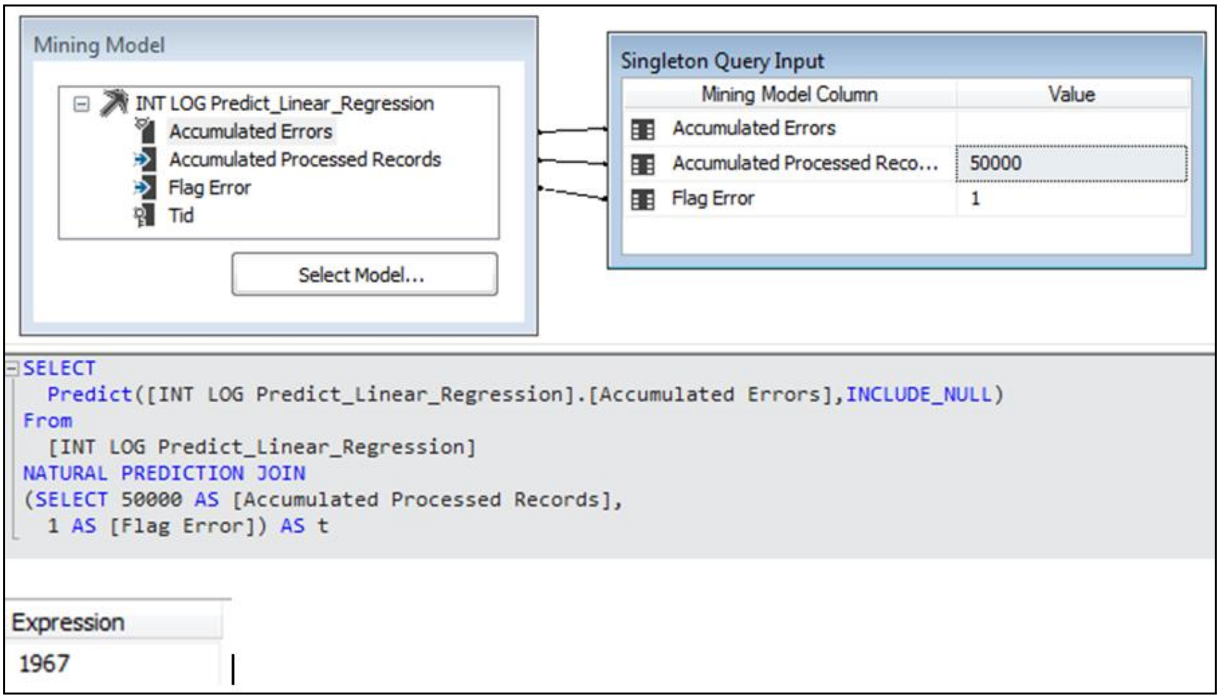

Figure 18 - Forecast of errors in the migration process

The data presented here come from a real work environment, and this is preparing for the actual process of migrating data between different systems. Registering all of these data into a log table made it possible to apply data mining and conduct analyses to help understand the main problems, and thus fine tune this process. This model reveals that increasing the number of records to be migrated directly affects the number of errors that might occur. The percentage of not migrated records would be around 4%.

For the knowledge management step, this prototype is performed by DML commands to the database, registering these findings to KMS tables for later consultation. The data inserted there are the migration script, along with a description of what it does. Then, issues that occur that are associated with the script, such as the errors presented here, as well as a solution for those issues, are presented.

Then, the comments table and evaluation table are connected to the script and issues table to allow other peers to discuss and make suggestions for how to improve this migration solution. The utility of this KMS is to make the migration script available to next similar process for the maintenance team. Knowledge about migration is explicit and recorded to the database. Over time, this behaviour enables building a library of functionalities to be used again without reinventing the scripts. In appendix C.1 there is a sample of KMS table's content.

The reporting system is also made in this prototype as queries from the database that hold with the findings presented here. This helps decision makers to know how it works in order to improve this process. Reports like the ones presented here help to understand what can happen in future scenarios, and to prepare accordingly. Furthermore, it also helps in relation with the client. For reporting there are a sample in appendix C.2.

6. Conclusion

The life cycle software development includes maintenance that starts after deployment of the solution to the client. The objective is to change and customise the software for the customer's needs and to correct faults. Regular updates are needed for continuous improvement of the software. Software maintenance processes includes problem identification, analysis and correction. Adaptation to changes in the technology or the market needs are also part of maintenance. Other processes, like platform migration, involve transferring data or an entire system to a different platform while maintaining its functionality.

6.1 Main findings

The general proposal of this work is to present a solution to a problem of software maintenance teams that relates to handling software issues. A service provider needs to quickly answer the customers' needs and to fix defects and other issues, even without having access to the productions environment, which is where the issue occurs. The findings are presented by topic with observations about the work developed here.

Propose software architecture

This master thesis focusses on software architecture as a solution to the presented problem. The architecture is composed of four layers. The data layer is represented in the prototype by SQL Server, which includes raw data log tables, KMS tables, SSAS analysis tables and report tables. The service layer comprises SSIS for the pre-processing module and SSAS for the mining and data analysis module. The business layer is composed of two modules, one for knowledge management by querying the KMS tables, and the other for reporting services, which was not implemented in this prototype. The presentation layer, which is the interface to the user, was not implemented, but a demonstration of data by Power BI was provided.

Knowledge management approach

Understanding the knowledge needs in an organisation is the key to implementing a KMA and taking advantage of this by adding value and competitiveness [7]. Registering explicit knowledge into a digital platform is not difficult to accomplish, since there are pre-existing tools for such, as mentioned, as well as online services, which makes this easier. The difficulty lies in the human factor, where all knowledge is developed and made explicit. This is the challenge for knowledge acquisition and strategies that should be developed.

To implement the KMS, strategic decisions in the organisation should be developed. Only then can a project be designed and developed by preparing the organisations technology for it and changing the local culture to take advantage of this new system. Encouraging human resources to be more participative with knowledge sharing about business problems and their solutions is the main issue to be overcome. Auditing existing knowledge is one way to have collaborators involved in this process, which also helps model the KMA by understanding the organisation's knowledge needs.

Data mining

Appling data mining in log data by using algorithms allows for revealing information and knowledge that might be missed if that analysis were performed by a human. The computer can perform different types of mining, such as descriptive, predictive and prescriptive mining over the log data. Those are more advanced techniques of mining that provide knowledge that can be preserved in the KMS for later usage. As for mining tools, commercial and open source tools are available. The second choice can be the starting point to learn more about mining and perform experiments with log files to see what comes out. These tools can develop interest for individuals or an organisation to see log file analysis as a path for information and knowledge that the organisation might be missing. Therefore, this increases motivation to implement a KMS and create routines to analyse log information.

AI

Artificial intelligence is constantly being developed, and platforms like Open AI or Microsoft AI make them easier to use in software applications and tools. This is also connected to ML, KD, data mining and data analytics. Artificial intelligence can be a part of mining models and analytic models, enhancing those processes. Automisation of analytic process is the objective, which involves handing that task to the computer and just waiting for the result. This allows for working with gigabytes of log information and obtaining valuable knowledge for businesses that help in making decisions. In this prototype, AI is intrinsic to data-mining algorithms.

Log data

To handle an automatic log data analysis, the proposed architecture was tested with a software prototype. This framework is actually about consolidating already existing tools, which, when put together, creates a solution. All parts culminate into a tool for the software maintenance team to use, and it produces results that an organisation can take advantage of through the knowledge that was developed.The log information to be analysed was produced by a process of migrating data between two different platforms conducted on a real work environment. The log information is the result of actions in the migrations, which includes business rules and destination platform rules. To apply these rules, data transformation had to be performed. For each record, all actions was recorded into a database log table.

For data filtering, SSIS reading raw log data were used, and the data that mattered to the analyses were transferred. This made querying and using SSAS to apply data-mining algorithms possible to obtain knowledge. This provided metrics about the migrating process, specifically concerning performance, errors, error classification and predictions for a future scenario. This knowledge can be used to improve this process and better understand the complex data transformations. Correct errors after migration is an essential part of this process, and relying on log information and knowledge, it is possible to group the record that possessed errors of the same kind and to provide a solution for them. Without the log data and knowledge retrieved from them, this task would be more complex and consume more time compared to these log data solutions.

This analysis can be used as a report to the chief operating officer (COO) as a means to help make decisions. The maintenance team that conducts this migration possesses knowledge about the results before the client, and therefore can take proactive actions. Workaround solutions can be performed effectively and efficiently because of this architecture. The customer relation management can further benefit from this.

6.2 Considerations and recommendations

The difficulty was centred on the process of data mining. The available tools for this that are studied in this document are dedicated to treating data as records of a database. Those mining tools expected data to be prepared in a table format to apply the mining algorithms. The first attempt of performing mining was a clear text log file, the results of which were not what were expected. The tool employed was the PHP-ML library, and from this research point of view, it was not achievable for this purpose. Clear text mining is different from record-like mining, and there are less available tools for this compared to record-like mining.

Achievable goals

It is important to understand whether the goals were achieved. This will allow the metrics that were presented to aid in understanding the software quality. Knowing less than six percent of errors can be well acceptable in software. Moreover, as being detected by analysing log information, it helps to have a proactive behaviour for correcting and improving the process. This knowledge being recorded to a KMS database allows collaborators to participate.

Recommendations

The KMS is the company memory of competencies, utilised as an asset. Preserving knowledge within an organisation is one key factor for solving problems and developing new competencies. This helps to improve competiveness and maintain a sustainable business. A KMS is one way to add value to the organisation and their clients. Producing knowledge from log data can be of use to a KMS, and therefore add value. Automising log data analytics and knowledge extraction creates routines in a way that can motivate organisations to apply this solution.

6.3 Future work

The architecture presented here is flexible enough to adapt to a different kind of log data, which can be in text file or a table log. If a log is in text file format, then algorithms have to be developed to extract the data and save them to a database table. Performing data mining on text files directly is more difficult than on database tables, since there are more tools available to work with a database. Developing tools to conduct mining in different types of log files is the recommended task for future work. Nevertheless, if new tools are developed, they can be integrated in the architecture discussed here because, being a modular solution, each software module can change and adapt to a new situation. For example, having modules in the service layer change to meet new requirements while preserving other layers would make better use of this framework. Having one single software application that implements these solutions is the ideal. Moreover, using AI to automate the entire process such that no human interventions would be required is highly recommended. More research about automatic log data analysers is needed for further enhancement of these solutions.

Bibliography

[1] P. Bourque and R. E. Fairley, *Guide to the Software Engineering - Body of Knowledge.* 2014.

[2] S. Liao, "Knowledge management technologies and applications—literature review from 1995 to 2002," *Expert Syst. Appl.*, vol. 25, no. 2, pp. 155–164, 2003.

[3] P. Sureephong *et al.*, "Knowledge Management System Architecture for the Industry Cluster To cite this version :," 2007.

[4] S. K. Patil and R. Kant, "Methodological literature review of knowledge management research," *Tékhne*, vol. 12, no. 1–2, pp. 3–14, 2014.

[5] H. Mao, S. Liu, J. Zhang, and Z. Deng, "Information technology resource, knowledge management capability, and competitive advantage: The moderating role of resource commitment," *Int. J. Inf. Manage.*, vol. 36, no. 6, pp. 1062–1074, 2016.

[6] G. Castrogiovanni, D. Ribeiro-Soriano, A. Mas-Tur, and N. Roig-Tierno, "Where to acquire knowledge: Adapting knowledge management to financial institutions," *J. Bus. Res.*, vol. 69, no. 5, pp. 1812–1816, 2016.

[7] V. H. Lee, A. T. L. Foo, L. Y. Leong, and K. B. Ooi, "Can competitive advantage be achieved through knowledge management? A case study on SMEs," *Expert Syst. Appl.*, vol. 65, pp. 136–151, 2016.

[8] É. F. De Souza, R. D. A. Falbo, and N. L. Vijaykumar, "Knowledge management initiatives in software testing: A mapping study," *Inf. Softw. Technol.*, vol. 57, no. 1, pp. 378–391, 2015.

[9] Y. Ma and Y. Qu, "Testing Requirement Analysis Method Based on Signal Path for High Reliability Software," p. 2011, 2011.

[10] W. Jie, T. Pei, S. Wen-qing, and X. Yan, "The generation of software reliability test cases based on software reuse," *2017 6th Int. Conf. Comput. Sci. Netw. Technol.*, pp. 161–164, 2017.

[11] A. Khosravi, A. R. C. Hussin, and M. Nilashi, "Toward software quality enhancement by Customer Knowledge Management in software companies," *Telemat. Informatics*, no. May, pp. 1–20, 2017.

[12] M. V. J. Veenman, L. Bavelaar, L. De Wolf, and M. G. P. Van Haaren, "The on-line assessment of metacognitive skills in a computerized learning environment," *Learn. Individ. Differ.*, vol. 29, pp. 123–130, 2014.

[13] J. B. de Vasconcelos, C. Kimble, P. Carreteiro, and Á. Rocha, "The application of knowledge management to software evolution," *Int. J. Inf. Manage.*, vol. 37, no. 1, pp. 1499–1506, 2017.

[14] Z. Li, P. Liang, and P. Avgeriou, "Application of knowledge-based approaches in software architecture: A systematic mapping study," *Inf. Softw. Technol.*, vol. 55, no. 5, pp. 777–794, 2013.

[15] A. Tiwana, *Knowledge Management Toolkit , The Amrit Tiwana Knowledge Management Toolkit , The*. 1999.

[16] Dataware Technologies Inc., "Seven Steps to Implementing Knowledge Management in Your Organization," p. 15, 1998.

[17] Ł. Ostrowski, M. Helfert, and S. Xie, "A conceptual framework to construct an artefact for meta-abstract design knowledge in design science research," *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, pp. 4074–4081, 2012.

[18] H. A. Al Mehairi and K. Binning, "Employee oriented vs. work oriented cultures and the moderating role of education on intention to share knowledge," *ITNG 2014 - Proc. 11th Int. Conf. Inf. Technol. New Gener.*, pp. 545–551, 2014.

[19] V. Cavalcante, S. Bianchi, A. Braz, F. Amorin, and N. Nauata, "Investigating business needs fluctuations on IT delivery operations," *Annu. SRII Glob. Conf. SRII*, pp. 19–26, 2014.

[20] D. L. Olson and D. Delen, *Advanced Data Mining Techniques*. 2008.

[21] Z. Autor:Shi, "Advanced Artificial Intelligence," in *Advanced Artificial Intelligence*, 2011.

[22] J. A. Lara, D. Lizcano, M. A. Mart??nez, and J. Pazos, "Data preparation for KDD through automatic reasoning based on description logic," *Inf. Syst.*, vol. 44, pp. 54–72, 2014.

[23] B. Kotiyal, A. Kumar, B. Pant, and R. H. Goudar, "Big data: Mining of log file through Hadoop," *2013 Int. Conf. Hum. Comput. Interact. ICHCI 2013*, 2013.

[24] N. Tselios, A. Stoica, M. Maragoudakis, N. Avouris, and V. Komis, "Enhancing user support in open problem solving environments through Bayesian Network inference techniques," *Educ. Technol. Soc.*, vol. 9, no. 4, pp. 150–165, 2006.

[25] P. Bilski, "Data set preprocessing methods for the artificial intelligence-based diagnostic module," *Meas. J. Int. Meas. Confed.*, vol. 54, pp. 180–190, 2014.

[26] G. Castellano, A. Fanelli, and M. Torsello, "LODAP: a LOg DAta Preprocessor for mining Web browsing patterns," *Proc. 6th Conf. 6th WSEAS Int. Conf. Artif. Intell. Knowl. Eng. Data Bases-Volume 6*, no. May 2014, pp. 12–17, 2007.

[27] S. Legg and M. Hutter, "Universal intelligence: A definition of machine intelligence," *Minds Mach.*, vol. 17, no. 4, pp. 391–444, 2007.

[28] S. H. Liao, P. H. Chu, and P. Y. Hsiao, "Data mining techniques and applications - A decade review from 2000 to 2011," *Expert Syst. Appl.*, vol. 39, no. 12, pp. 11303–11311, 2012.

[29] G. S. Cho, "A computer forensic method for detecting timestamp forgery in NTFS," *Comput. Secur.*, vol. 34, pp. 36–46, 2013.

[30] J. Y. Han, "Transaction logfile analysis in health communication research: Challenges and opportunities," *Patient Educ. Couns.*, vol. 82, no. 3, pp. 307–312, 2011.

[31] S. O. Fageeri and R. Ahmad, "An Efficient Log File Analysis Algorithm Using Binary-based Data Structure," *Procedia - Soc. Behav. Sci.*, vol. 129, pp. 518–526, 2014.

[32] H. Sha, T. Liu, P. Qin, Y. Sun, and Q. Liu, “EPLogCleaner: Improving data quality of enterprise proxy logs for efficient web usage mining,” *Procedia Comput. Sci.*, vol. 17, pp. 812–818, 2013.

[33] A. Ambre and N. Shekokar, “Insider threat detection using log analysis and event correlation,” *Procedia Comput. Sci.*, vol. 45, no. C, pp. 436–445, 2015.

[34] H. Jiang and H. Meng, “A Parallel FP-Growth Algorithm Based on GPU,” *2017 IEEE 14th Int. Conf. E-bus. Eng.*, pp. 97–102, 2017.

[35] H. Li, Y. Zhang, N. Zhang, and H. Jia, “A Heuristic Rule Based Approximate Frequent Itemset Mining Algorithm,” *Procedia Comput. Sci.*, vol. 91, no. Itqm, pp. 324–333, 2016.

[36] T. S. Madhulatha, “No TitleMining Inflation Rate Using Predictive and Descriptive Data Mining Technique,” *Int. Conf. Comput. Sustain. Glob. Dev.*, pp. 228–232, 2014.

[37] K. Danjuma and A. Osofisan, “Evaluation of Predictive Data Mining Algorithms in,” no. Cvd.

[38] B. Kawas, M. S. Squillante, D. Subramanian, and K. R. Varshney, “Prescriptive analytics for allocating sales teams to opportunities,” *Proc. - IEEE 13th Int. Conf. Data Min. Work. ICDMW 2013*, pp. 211–218, 2013.

[39] G. P. K. Wu and K. C. C. Chan, “Clustering Driving Trip Trajectory Data Based on Pattern Discovery Techniques,” *3rd Int. Conf. Big Data Anal. IEEE*, pp. 3734–3743, 2018.

[40] I. Mavridis and H. Karatza, “Performance evaluation of cloud-based log file analysis with Apache Hadoop and Apache Spark,” *J. Syst. Softw.*, vol. 125, pp. 133–151, 2017.

[41] J. Yang, H. Huang, and X. Jin, “Mining web access sequence with improved apriori algorithm,” *Proc. - 2017 IEEE Int. Conf. Comput. Sci. Eng. IEEE/IFIP Int. Conf. Embed. Ubiquitous Comput. CSE EUC 2017*, vol. 1, pp. 780–784, 2017.

[42] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design Science in Information Systems Research,” *MIS Q.*, vol. 28, no. 1, pp. 75–105, 2004.

[43] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A Design Science Research Methodology for Information Systems Research,” *J. Manag. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, 2007.

Appendix A

A.1. Database log table model

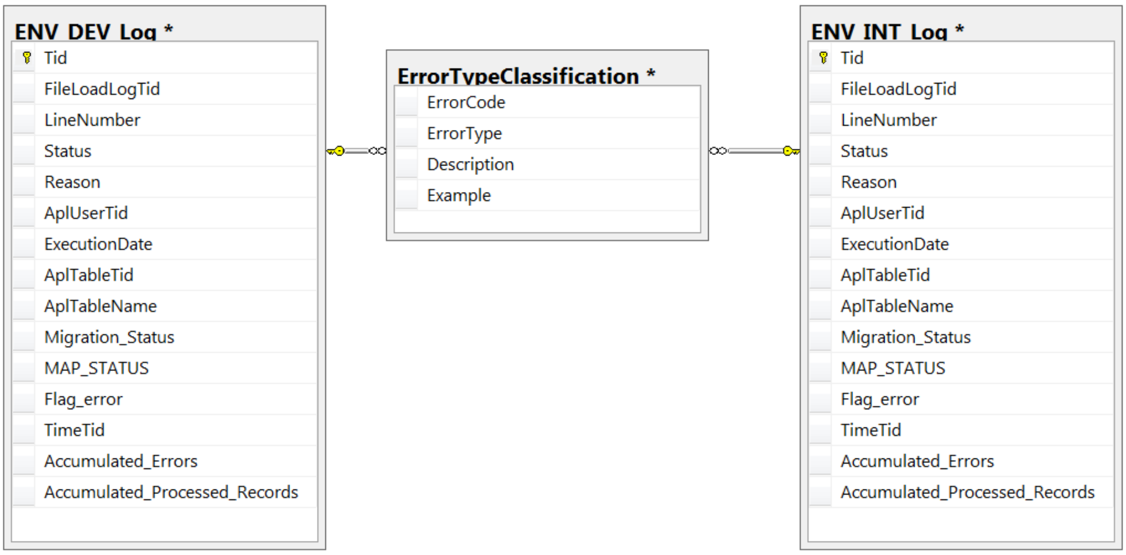

Figure 19 - Database model of log tables from migration project

A.2 KMS database model

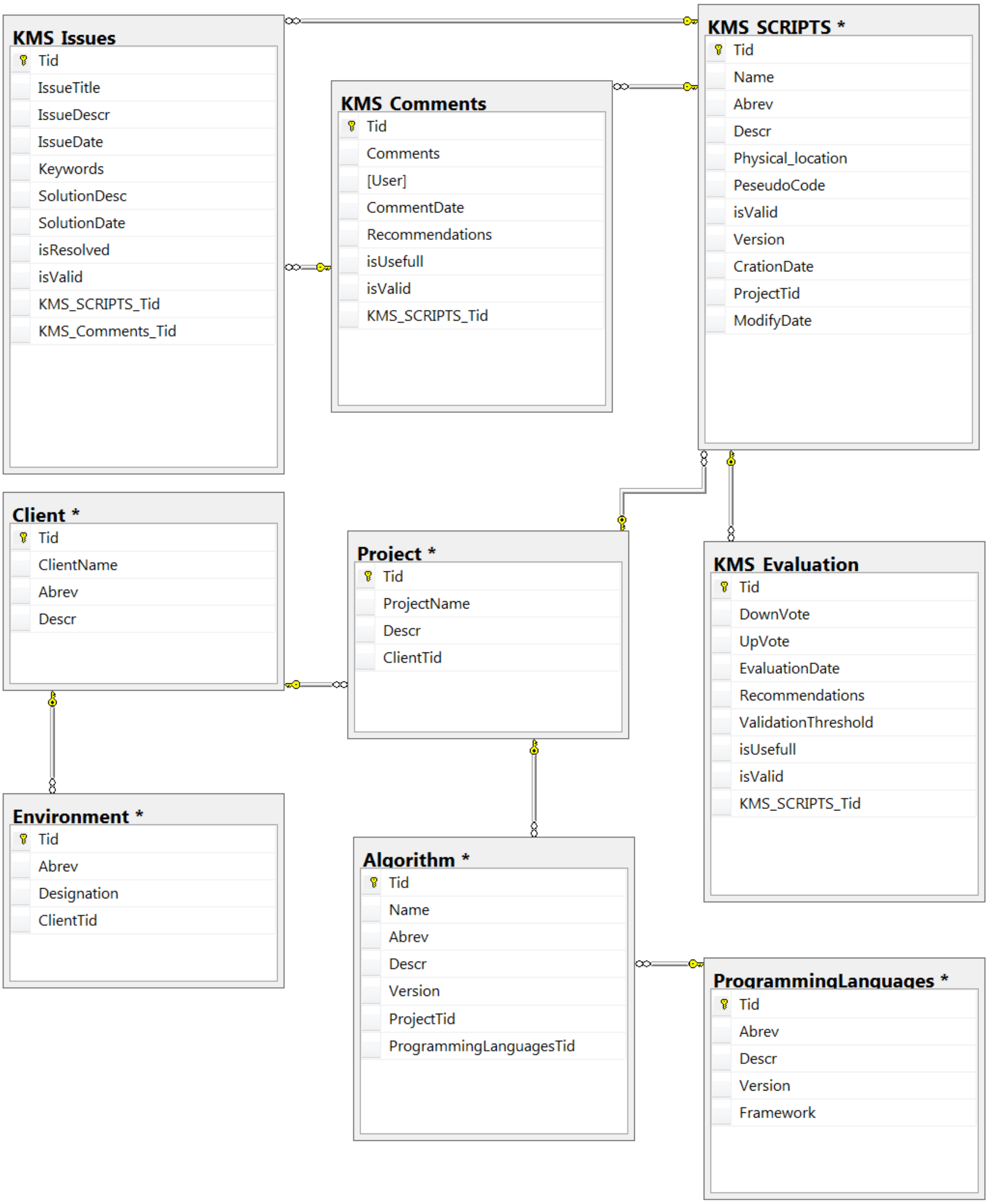

Figure 20 - Database model for KMS

Appendix B

B.1. Development environment

Table 6 - Sample of development environment table log

Line Number	Status	Reason	Execution Date	Migration Status	MAP STATUS	Flag
523	OK	Successful migrated	2018-05-21 10:48:51.633	1	0	0
524	OK	Successful migrated	2018-05-21 10:48:51.637	1	0	0
...	...	...	...	...	...	...
3011	NOT_OK	Cannot insert the value NULL into column	2018-05-21 13:24:50.153	-3	-20	1
3012	NOT_OK	Cannot insert the value NULL into column	2018-05-21 13:24:50.153	-3	-20	1
...	...	...	...	...	...	...
1055	NOT_OK	String or binary data would be truncated.	2018-05-21 17:01:01.340	-1	-3	1
1275	NOT_OK	String or binary data would be truncated.	2018-05-21 17:01:01.343	-1	-3	1
...	...	...	...	...	...	...

Table 7 - Categorizing error type of migrated data for development environment

Error Type	Number of Records	Description
NULL_ERROR	1308	Cannot insert the value NULL into column
SQL_ERROR	6	Error reported from SQL
NO_ERROR	23210	There were no errors. A DML command was perform without errors

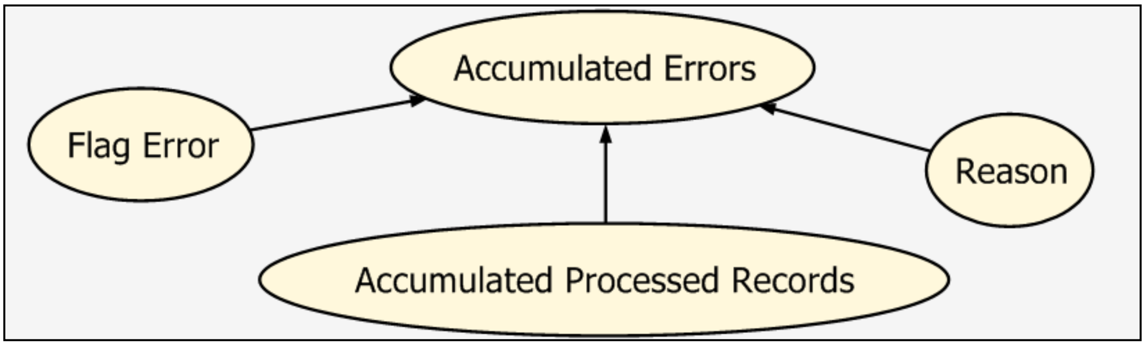

Figure 21 - Developing environment dependencies model in SSAS

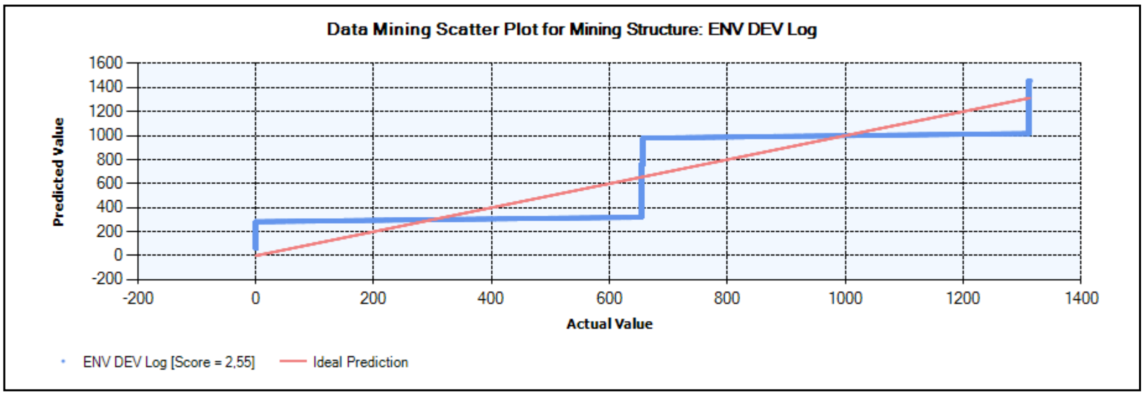

Figure 22 - Error predictions in development environment from log table

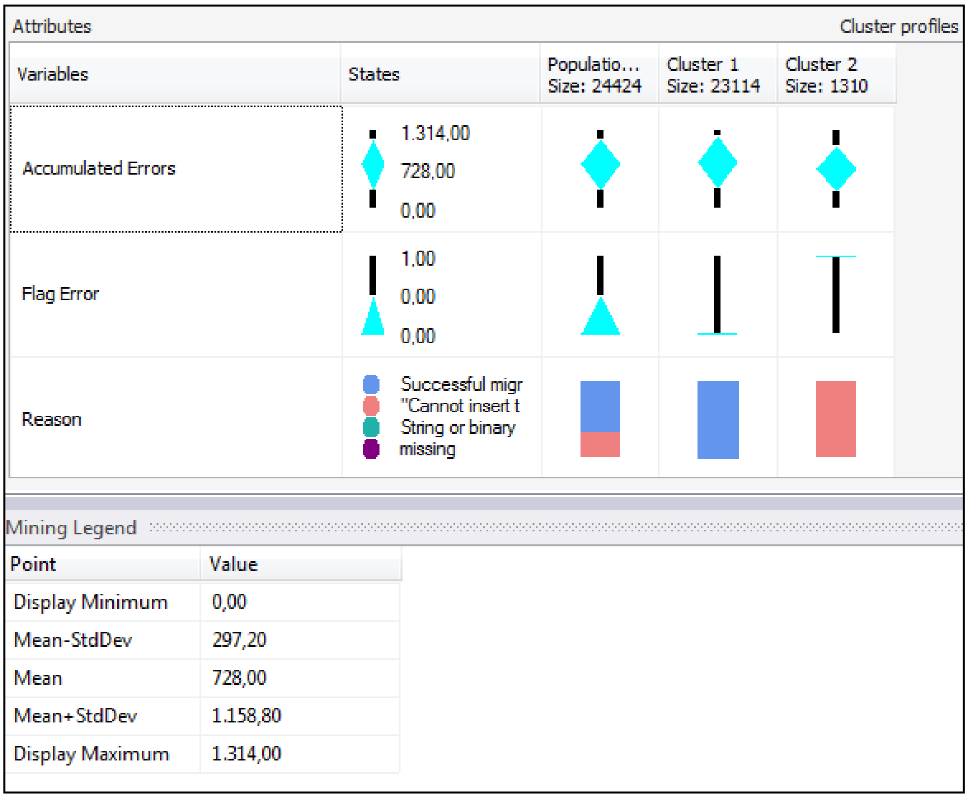

Figure 23 - Cluster mining related development environment

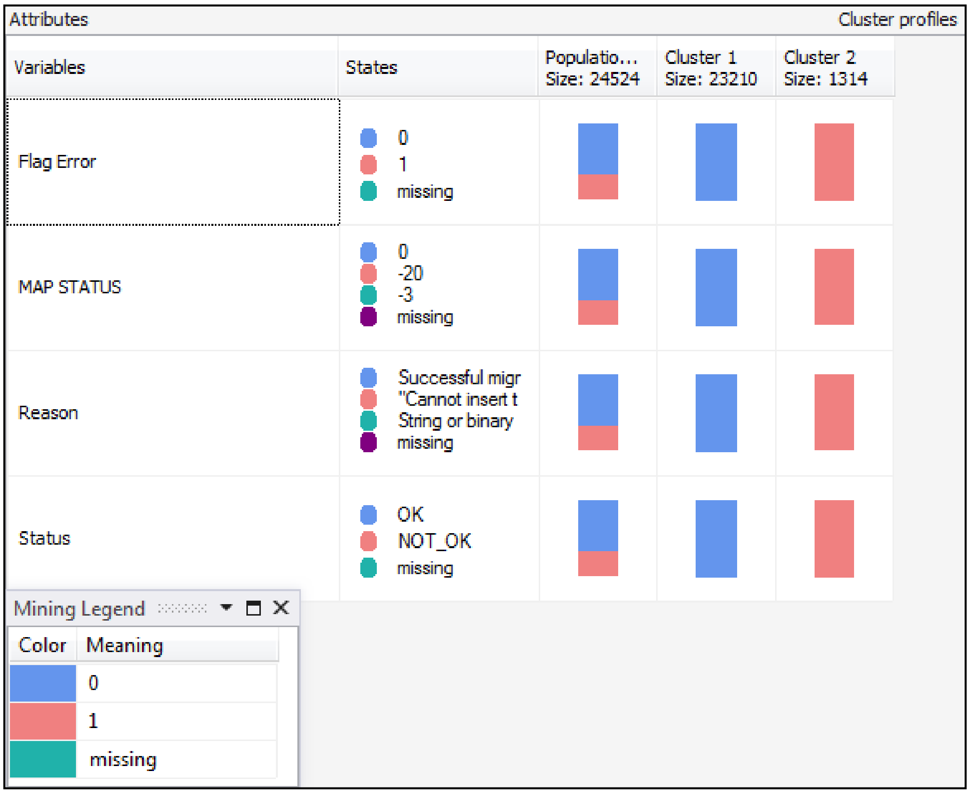

Figure 24 - Error classification from log table of developing environment in SSAS

B.2. Integration environment

Table 8 - Sample of integration environment table log

Line Number	Status	Reason	Execution Date	Migration Status	MAP STATUS	Flag
1	OK	Successful migrated	2018-05-24 21:06:16.170	0	0	0
2	OK	Successful migrated	2018-05-24 21:06:16.180	0	0	0
3	OK	Successful migrated	2018-05-24 21:06:16.187	0	0	0
…	…	…	…	…	…	…
855	NOT_OK	A name already exists in referred table	2018-05-24 21:06:20.433	-1	-51	1
…	…	…	…	…	…	…
880	NOT_OK	A id document number is mandatory	2018-05-24 21:06:20.457	-1	-59	1
…	…	…	…	…	…	…

Table 9 - Categorizing error type of migrated data for integration environment

Error Type	Number of Records	Description
ID_DOC_NUMBER	68	A id document number is mandatory
REPEATED_NAME	265	A name already exists in referenced table
NO_ERROR	24275	There were no errors. A DML command was perform without errors
UNIQUE_KEY_CONSTRAINT	436	Violation of UNIQUE KEY CONSTRAINT

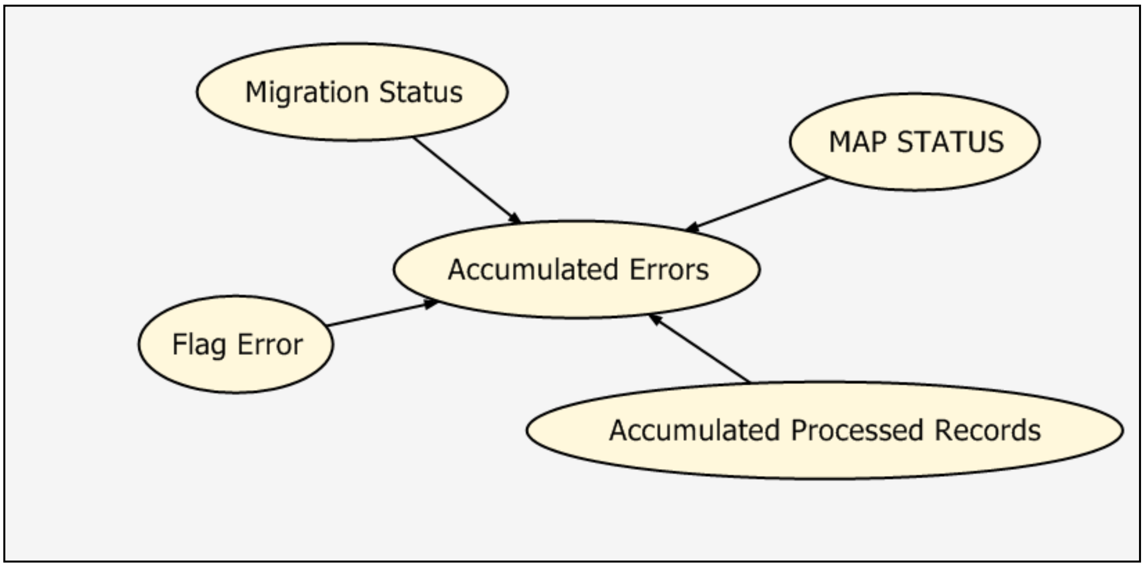

Figure 25 - Integration environment dependencies model in SSAS

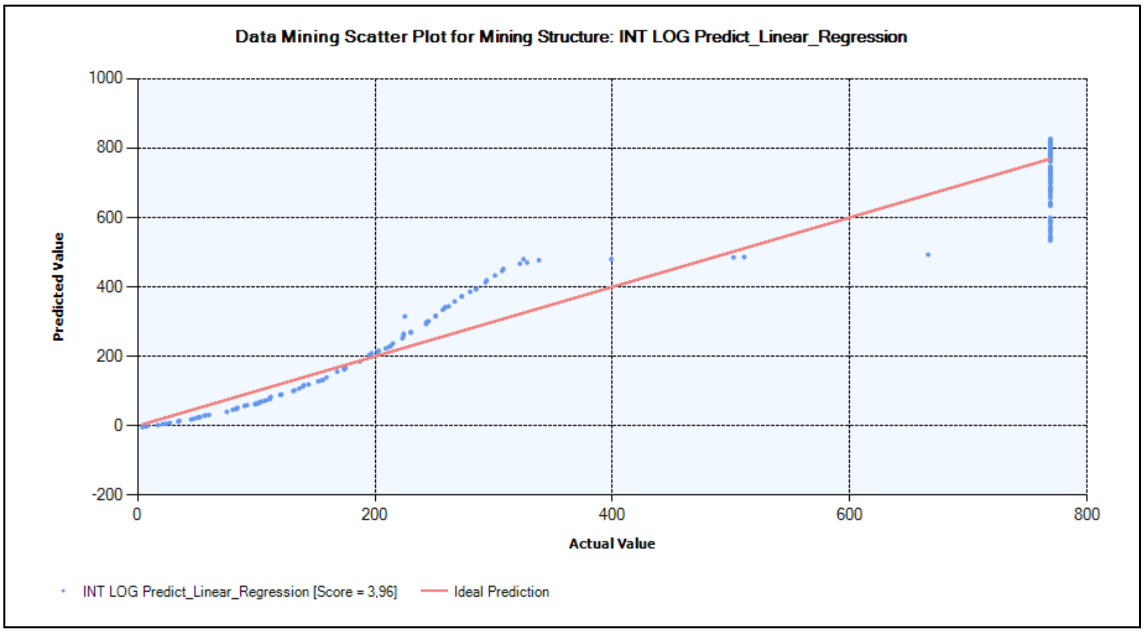

Figure 26 - Error predictions in development environment from log table

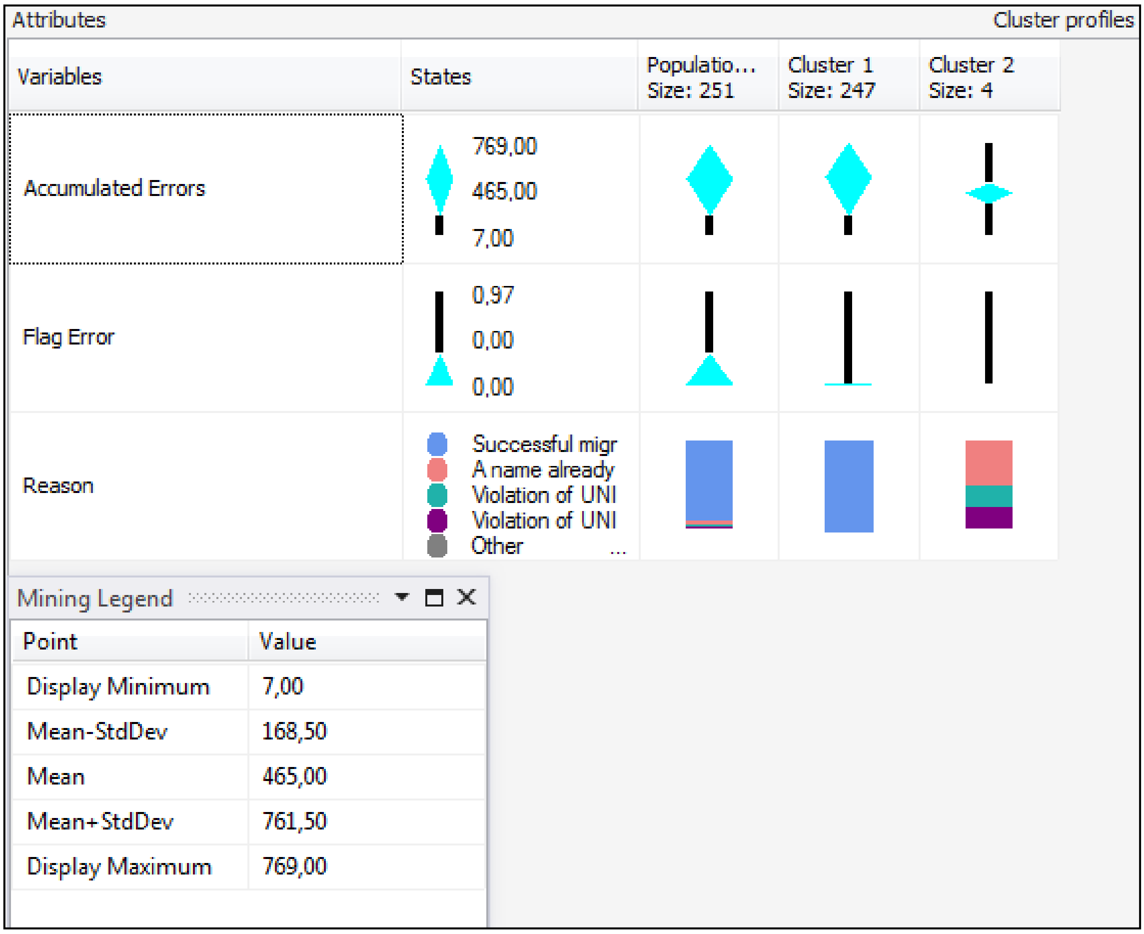

Figure 27 - Cluster mining related integration environment

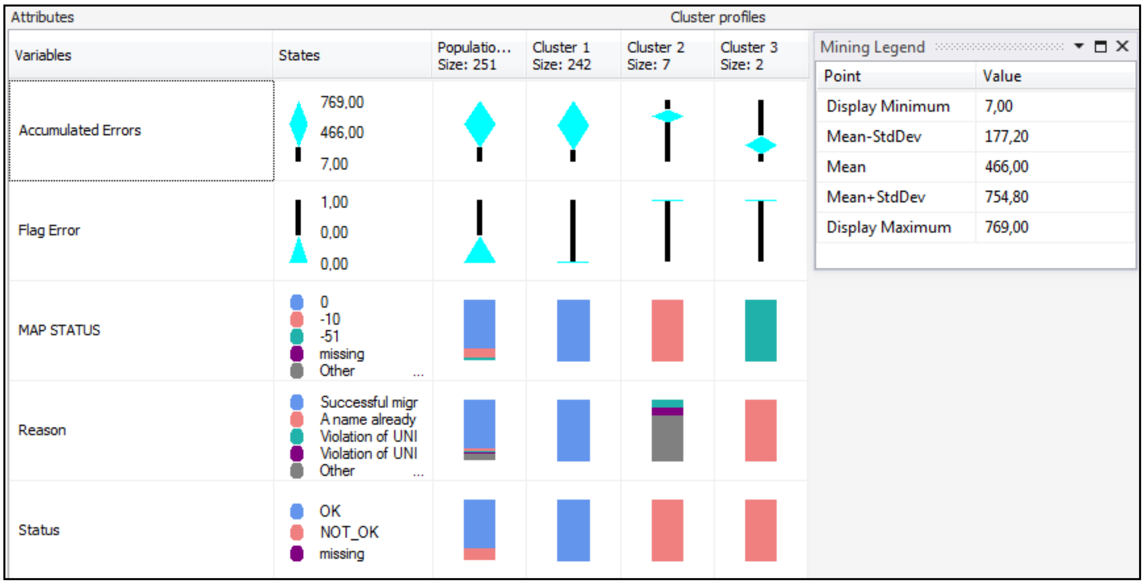

Figure 28 - Error classification from log table of developing environment in SSAS

Appendix C

C.1. Sample of KMS tables

Table 10 - KMS script table

Name	Abrev	Descr	Physical location	Creation Date
SQL_FC	SQL_FC	Transfer financial company to destination database	C:\Migration\SQL_FC_transf.sql	2018-01-10
SQL_Customer	SQL_Customer	Transfer customers to destination database	C:\Migration\SQL_Customer_transf.sql	2018-01-10
SQL_Marketer	SQL_Marketer	Transfer Marketer to destination database	C:\Migration\SQL_Marketer_transf.sql	2018-01-10
SQL_Insurance	SQL_Insurance	Transfer insurance to destination database	C:\Migration\SQL_Insurance_transf.sql	2018-01-10
SQL_Mediator	SQL_Mediator	Transfer Mediator to destination database	C:\Migration\SQL_Mediator_transf.sql	2018-01-10
Correcting mediator code.sql	Mediator code	Correcting to code 100 of the Mediator	C:\Migration\MediatorCodeCorrection.sql	2018-06-04
Correcting insurance data	Insurance data	Correcting to insurance data	C:\Migration\CorrectionInsuranceData.sql	2018-06-04
Correcting customer address	customer address	Correcting customer address	C:\Migration\CorrectionsCustomerAddress.sql	2018-06-04
Correcting financial company	Financial Company	Correcting the connections between financial company and custumer	C:\Migration\CorrectionsFinancialCompany.sql	2018-06-04

Table 11 - Sample of comments table

Tid	Comments	Comment Date	Recommendations	isUsefull	isValid
1	This script corrected the issue	05-06-2018	None	1	1
2	The migration of data was done with success	04-06-2018	None	1	1

Table 12 - Sample of evolution table

Tid	Down Vote	Up Vote	Evaluation Date	Recommendations	Validation Threshold	isUsefull	isValid
1	0	1	2018-06-04	None	5	1	1
2	0	1	2018-06-04	None	5	1	1

Table 13 - Sample of issue table

Issue Title	Issue Descr	Issue Date	Keywords	Solution Desc	Solution Date
Financial company address is missing.	For financial company, the address is missing.	04-06-2018	address, financial, company, missing	update field address from table Address with a proper street name	2018-06-05
Insurance relation to Marketer	The insurance relation to Marketer is not properly related.	05-06-2018	insurance, Marketer, relation	insert a new relation between insurance and marketer	2018-06-05

C.2. Sample of report tables

Table 14 - KMS report

Issue Solution Descr	Issue Keywords	Issue Descr	Issue Title	Script Descr	Script Name
insert a new relation between insurance and marketer	insurance ,Marketer ,relation	The insurance relation to Marketer is not properly related.	Insurance relation to Marketer	Correcting the connections between financial company and customer	Correcting financial company
insert a new relation between insurance and marketer	insurance ,Marketer ,relation	The insurance relation to Marketer is not properly related.	Insurance relation to Marketer	Correcting the connections between financial company and customer	Correcting financial company

Table 15 - Customize report from log tables

Log Table Name	Environment	MAP STATUS	Migration Status	Execution Date	Reason	Status	Line Number	File Load LogTid
Env_DEV_log	Dev	0	1	2018-05-21 10:48:51.627	Successful migrated	OK	521	12
Env_DEV_log	Dev	0	1	2018-05-21 10:48:51.633	Successful migrated	OK	523	12
Env_DEV_log	Dev	0	1	2018-05-21 10:48:51.637	Successful migrated	OK	524	12
…	…	…	…	…	…	…	…	…
Env_INT_log	Int	-59	-1	2018-05-24 21:44:46.593	A id document number is mandatory	NOT_OK	422	6
Env_INT_log	Int	-51	-1	2018-05-24 21:44:46.610	A name already exists in refered table	NOT_OK	447	6
Env_INT_log	Int	-10	-1	2018-05-24 22:47:20.040	Violation of UNIQUE KEY constraint 'ENTITY_AK'. Cannot insert duplicate key in object 'dbo.Entity'. The duplicate key value is (48745940, 1).	NOT_OK	1	2
…	…	…	…	…	…	…	…	…

Table 16 - Report queries

Added Date	Report Name	Abrev	Description	Query
22-06-2018	Error report from integration log table	Error report	Report that relate errors with description	select err.ErrorType,err.ErrorCode,err.Description,[FileLoadLogTid], [LineNumber], [Reason], [ExecutionDate], [Migration_Status], [MAP_STATUS], 'Int' as ENV, 'Env_INT_log' as LogTableName from Env_INT_log l, [ErrorTypeClassification] err WHERE Status = 'NOT_OK' and err.errorcode = l.map_status
22-06-2018	Error report from integration log table	Error report	Report that relate errors with description	select err.ErrorType,err.ErrorCode,err.Description,[FileLoadLogTid], [LineNumber], [Reason], [ExecutionDate], [Migration_Status], [MAP_STATUS], 'Int' as ENV, 'Env_INT_log' as LogTableName from Env_INT_log l, [ErrorTypeClassification] err WHERE Status = 'NOT_OK' and err.errorcode = l.map_status
22-06-2018	Success report from developing log table	Success report	Report that list the success migrated records	select [FileLoadLogTid], [LineNumber], [Status], [Reason], [ExecutionDate], [Migration_Status], [MAP_STATUS], 'Dev', 'Env_DEV_log'from Env_DEV_log l WHERE Status = 'OK'
22-06-2018	Success report from integration log table	Success report	Report that list the success migrated records	select [FileLoadLogTid], [LineNumber], [Status], [Reason], [ExecutionDate], [Migration_Status], [MAP_STATUS], 'Dev', 'Env_DEV_log' from Env_INT_log l WHERE Status = 'OK'
22-06-2018	error report from developing log table	Success report	Report that list the success migrated records	select [FileLoadLogTid], [LineNumber], [Status], [Reason], [ExecutionDate], [Migration_Status], [MAP_STATUS], 'Int', 'Env_INT_log' from Env_DEV_log l WHERE Status = 'NOT_OK'
22-06-2018	error report from integration log table	Error report	Report that list the success migrated records	select [FileLoadLogTid], [LineNumber], [Status], [Reason], [ExecutionDate], [Migration_Status], [MAP_STATUS], 'Int', 'Env_INT_log' from Env_INT_log l WHERE Status = 'NOT_OK'

Appendix D

D.1. Project planning

Project planning

Process of selection for supervisor position of the Master Dissertation	**7 days**	**Mon 07-08-17**	**Tue 15-08-17**
Proposal submission	4 days	Mon 07-08-17	Thu 10-08-17
Guidance / counseling negotiation	3 days	Thu 10-08-17	Mon 14-08-17
Completion of the advisor selection	0 days	Tue 15-08-17	Tue 15-08-17
Initial proposal to the supervisor	**8 days**	**Tue 12-09-17**	**Thu 21-09-17**
Beginning of Drafts of the Master degree dissertation	7 days	Tue 12-09-17	Wed 20-09-17
Consolidation of ideas for the development of thesis	7 days	Tue 12-09-17	Thu 21-09-17
Draft Conclusion	0 days	Thu 21-09-17	Thu 21-09-17
Thesis guidance	**65 days**	**Mon 09-10-17**	**Fri 05-01-18**
Exchange of emails with the advisor	**40 days**	**Mon 13-11-17**	**Fri 05-01-18**
Meeting with the Advisor	**38 days**	**Mon 13-11-17**	**Wed 03-01-18**
Writing the review of the systematic literature and methodology	**0 days**	**Fri 05-01-18**	**Fri 05-01-18**
Systematic Review Completion	0 days	**Fri 05-01-18**	**Fri 05-01-18**
Final Dissertation Document Final Format	**25 days**	**Sat 06-01-18**	**Thu 08-02-18**
Exchange of emails with the advisor	**23 days**	**Sat 06-01-18**	**Tue 06-02-18**
Meeting with the Advisor	**23 days**	**Sat 06-01-18**	**Tue 06-02-18**
Thesis Document Delivery	0 days	Fri 09-02-18	Fri 09-02-18
Define model Architecture	**16 days**	**Mon 26-02-18**	**Mon 19-03-18**
Meeting with the Advisor	**16 days**	**Mon 26-02-18**	**Mon 19-03-18**
Writing the review of the systematic literature and methodology	**16 days**	**Mon 26-02-18**	**Mon 19-03-18**
Model Engineering for parsing log data	**16 days**	**Mon 05-03-18**	**Sat 24-03-18**
Meeting with the Advisor	**16 days**	**Mon 05-03-18**	**Sat 24-03-18**
Writing the review of the systematic literature and methodology	**16 days**	**Mon 05-03-18**	**Sat 24-03-18**

Figure 29 - Project planning first section

Model Engineering Data Mining	11 days	Mon 19-03-18	Sat 31-03-18
Meeting with the Advisor	11 days	Mon 19-03-18	Sat 31-03-18
Writing the review of the systematic literature and methodology	11 days	Mon 19-03-18	Sat 31-03-18
Model Engineering Knowledge Management	11 days	Mon 02-04-18	Sat 14-04-18
Meeting with the Advisor	10 days	Tue 03-04-18	Sat 14-04-18
Writing the review of the systematic literature and methodology	10 days	Tue 03-04-18	Sat 14-04-18
Model Engineering Artificial Intelegence	11 days	Mon 16-04-18	Sat 28-04-18
Meeting with the Advisor	6 days	Tue 17-04-18	Tue 24-04-18
Meeting with the Advisor 1	1 day	Tue 17-04-18	Tue 17-04-18
Meeting with the Advisor 2	1 day	Tue 24-04-18	Tue 24-04-18
Writing the review of the systematic literature and methodology	11 days	Mon 16-04-18	Sat 28-04-18
Wrting Systematic Review	11 days	Mon 30-04-18	Sat 12-05-18
Meeting with the Advisor	6 days	Tue 01-05-18	Tue 08-05-18
Writing the review of the systematic literature and methodology	5 days	Tue 08-05-18	Sat 12-05-18
Evaluated the Architecture	11 days	Mon 14-05-18	Sat 26-05-18
Meeting with the Advisor	6 days	Tue 15-05-18	Tue 22-05-18
Writing the review of the systematic literature and methodology	11 days	Mon 14-05-18	Sat 26-05-18
Evaluated each Model Engineered	6 days	Mon 28-05-18	Sat 02-06-18
Meeting with the Advisor	5 days	Tue 29-05-18	Sat 02-06-18
Writing the review of the systematic literature and methodology	6 days	Mon 28-05-18	Sat 02-06-18

Figure 30 - Project planning second section

Results	12 days	Sat 02-06-18	Sat 16-06-18
Meeting with the Advisor	11 days	Mon 04-06-18	Sat 16-06-18
Writing the review of the systematic literature and methodology	12 days	Sat 02-06-18	Sat 16-06-18
Conclusions and Future work	6 days	Mon 18-06-18	Sat 23-06-18
Meeting with the Advisor	5 days	Tue 19-06-18	Sat 23-06-18
Writing the review of the systematic literature and methodology	6 days	Mon 18-06-18	Sat 23-06-18
Entrega Final da Tese	0 days	Tue 26-06-18	Tue 26-06-18

Figure 31 - Project planning third section