To link to this article: DOI:10.1016/j.eswa.2012.08.039
http://dx.doi.org/10.1016/j.eswa.2012.08.039

To cite this version:

Kamsu Foguem, Bernard and Rigal, Fabien and Mauget, Félix *Mining association rules for the quality improvement of the production process.* Expert Systems with Applications, vol. 40 (n° 4). pp. 1034-1045. ISSN 0957-4174

Mining association rules for the quality improvement of the production process

Bernard Kamsu-Foguem *, Fabien Rigal, Félix Mauget

Laboratory of Production Engineering (LGP), EA 1905, ENIT, INPT, University of Toulouse, 47 Avenue d'Azereix, BP 1629, 65016 Tarbes Cedex, France

ARTICLE INFO

Keywords:
Data mining
Association rule mining
Knowledge discovery
Continuous improvement
Drilling product manufacturing
Industrial maintenance

ABSTRACT

Academics and practitioners have a common interest in the continuing development of methods and computer applications that support or perform knowledge-intensive engineering tasks. Operations management dysfunctions and lost production time are problems of enormous magnitude that impact the performance and quality of industrial systems as well as their cost of production. Association rule mining is a data mining technique used to find out useful and invaluable information from huge databases. This work develops a better conceptual base for improving the application of association rule mining methods to extract knowledge on operations and information management. The emphasis of the paper is on the improvement of the operations processes. The application example details an industrial experiment in which association rule mining is used to analyze the manufacturing process of a fully integrated provider of drilling products. The study reports some new interesting results with data mining and knowledge discovery techniques applied to a drill production process. Experiment's results on real-life data sets show that the proposed approach is useful in finding effective knowledge associated to dysfunctions causes.

1. Introduction

Engineering applications of artificial intelligence have attracted substantial consideration from industrial practitioners and researchers because of its ability to learn and comprehend facts and principles in order to acquire knowledge and apply it in practice. Continuous improvement refers to both incremental and breakthrough improvement in organizational performance (Linderman, Schroeder, Zaheer, Liedtke, & Choo, 2004). Improvement can result in such things as improved customer value, reduction of defects and errors, improved productivity, improved cycle time performance safety, and motivation (Evans & Lindsay, 2001). This often occurs through the adherence to a stepwise problem solving approach consisting of number of steps for problem contextualization, problem analysis, solution generation, and lessons learned (Kamsu-Foguem, Coudert, Geneste, & Beler, 2008). The problem-solving approach focuses on a characterization of cognitive processes in reasoning tasks and cognitive considerations deal with knowledge capitalization on certain structural and processing regularities that give strength to generalizations (Patel, Arocha, & Kaufman, 2001). Problem-solving methods play a significant role in knowledge acquisition and engineering, since their abstract knowledge level is valuable to achieve goals of tasks by applying domain knowledge with the sequential process of searching for a solution path. They can be applied, among others, to describe the reasoning process in a structured manner, to guide the knowledge acquisition process and to facilitate knowledge sharing and reuse (Benjamins & Fensel, 1998).

Problem-solving research places a greater emphasis on an evolving process (e.g. analyzing with a set of tools) rather than a fixed selection process, by application of deductive reasoning (i.e. a specific conclusion is arrived at from a general principle) and inductive reasoning (i.e. a general conclusion is arrived at by specific examples) (Newell & Simon, 1972). A stepwise problem solving model presents a systematic analysis by ensuring multiple perspectives of a problem are captured and engaged in formulation of an insightful solution (Gibbons, 2000). Such systemic approach helps in problem comprehension, aids identification of its root causes and impacts knowledge creation (Jabrouni, Kamsu-Foguem, Geneste, & et Vaysse, 2011). In general, problem-solving studies are more operational in formalizing latent sources of error as well as describing the root causes of problems or events. Besides, experienced knowledge differs in important respects from intermediate knowledge and has a qualitatively distinct engagement with differential use of reasoning strategies in problem solving: for example, experts are involved in the process of situation assessment with a data-driven reasoning whereas the novices and intermediates are much more proactive in handling solution options and organizing further investigations with a hypothesis-driven reasoning (Patel, Kaufman, & Arocha, 2002).

Plan Do Check Act (PDCA), Lean, and Six Sigma are three of the common stepwise models of problem solving used in industry today. Each has its rationale with relevant features, and each approach, when deployed accurately, can yield some interesting results and sustain improvement (as described in Table 1 (Chiodo,

* Corresponding author. Tel.: +33 6 24 30 23 37; fax: +33 5 62 44 27 08.
E-mail address: Bernard.Kamsu-Foguem@enit.fr (B. Kamsu-Foguem).

Table 1
Common continuous improvement methodologies (Chiodo et al., 2011).

	PDCA	Lean	Six sigma
Definition	Cyclical product and/or process improvement emphasis on control	Elimination of waste, speed, efficiency	Reduction in defects and variation data driven
Objective	Small incremental improvements, repeat process	Relentless pursuit or perfection by Increasing value-adding activities by eliminating waste	Reduce process variation to near perfect (Six Sigma) levels
Methodology	Deming-Shewhart PDCA cycle • Plan • Do • Check • Act	Value stream mapping: 5S: • Sort • Straighten • Scrub • Systematize • Sustain	DMAIC • Define • Measure • Analyze • Improve • Control

PDCA, plan, do, check act; DMAIC, define, measure, analyze, improve, control.

Rosenhauer, & Worsowicz, 2011). The stepwise fashion of problem solving and the associated continuous improvement methodologies can be used at distinct levels of organization, in service and administrative as well as manufacturing processes. Quality management practices that promote monitoring and experience feedback of information and operations management allow learning and knowledge creation (Choo, Linderman, & Schroeder, 2007). Many quality control and improvement activities (e.g. inspection/screening, quality analysis, process control, quality monitoring) that are related to manufacturing problems utilize data analysis methods to mine huge data sets collected through production processes in manufacturing industry. The ideas for improvement provided by such activities are a key element in the experience feedback process to further corrective or preventive actions (Foster, 2008). The generated ideas can be tested through the use of data analysis techniques that link continuous improvement to knowledge creation processes.

Knowledge Discovery in Databases (KDD) has become one of the fastest growing research topics in mathematics and computer science, because the ability to continually change and acquire new understanding is a driving force for its applications (Liao, Chu, & Hsiao, 2012; Washio, 2007). For example data mining have served in the search and retrieval of computer-aided design elements (Liu, McMahon, Ramani, & Schaefer, 2011). The KDD process, specifically data mining techniques, is used to characteristically discover knowledge from data (Zhu & Davidson, 2007). The data mining process extracts knowledge from an existing data set and transforms it into a human-understandable structure for further use (Witten, Frank, & Hall, 2011). Data mining techniques are required to help in identification of model characteristics important to capture and document in an enhancement context of the safety and reliability of complex engineering systems (Saitta, Raphael, & Smith, 2005). Data mining applications are very suitable for quality improvement programs (e.g. Kaizen-PDCA, 9-Steps, 8D, 7-Step, PDCA, Six Sigma-DMAICS) in manufacturing (Köksal, Batmaz, & Testik, 2011), due to advances in data collection systems, analysis tools and interpretation methods (Alzghoul & Löfstrand, 2011). However, there are some factors influencing the adoption of data mining tools (DMTs), primarily the task-oriented dimension (job relevance, output quality, result demonstrability, response time, and format) (Huang, Liu, & Chang, 2012). So, it is decisive to ensure good means of promoting, efficient and effective information access, processing, and use by people and organizations (Detlor, 2010). Data mining involves six common classes of tasks (Kantardzic, 2011) (Ngai, Hu, Wong, Chen, & Sun, 2011):

- *Anomaly detection* (outlier/change/deviation detection) – Anomaly detection is engaged to identify the unusual data records and to detect data objects that are unacceptably different from or inconsistent with the remaining data set. A system protection method can be applied for detecting anomalies in user patterns, with the purpose to provide guidance for facilitating the reconfiguration of collaboration systems (Lee, Ryu, Shin, & Cho, 2012).
- *Association rule mining* (association rule learning) – Association rule learning is employed to discover interesting relations between variables in large databases. This dependency modeling analyses strong rules discovered in databases using different measures of interestingness. The use of association rules mining in frequent patterns captured from industrial processes can provide useful knowledge to explain industrial failures (Martínez-de-Pisón, Sanz, Martínez-de-Pisón, Jiménez, & Conti, 2012).
- *Clustering* – Clustering serves to partition objects into conceptually meaningful groups (clusters), such that similar objects are in the same group, while dissimilar objects are in different groups. Clustering is an unsupervised learning problem where one is only given the unlabeled data and the goal is to learn the underlying structure. A graph clustering algorithm approach for manufacturing cell formation can be used to makes an improvement in the number of intercell moves (Oliveira, Ribeiro, & Seok, 2009).
- *Classification* – Classification is the procedure of assigning labels to objects such that objects' labels within the same categories will match previously labeled objects from a training set, by generalizing known structure. Classification is traditionally a type of supervised learning problem that tries to learn a function from the data in order to predict the categorical labels of unknown objects to differentiate between objects of different classes. Classification procedure can be employed to assist decision makers to classify alternatives into multiple groups, reduce the number of misclassifications and lessen the impact of outliers (Ma, 2012).
- *Regression* – Regression is a statistical methodology for modeling and analyzing several variables and is used to understand which among the independent variables are related to the dependent variable, and to explore the forms of these relationships. Most commonly, regression analysis attempts to find a function of the independent variables that models the data with a method of estimation. In the work of Alzghoul and his colleagues, different data-stream-based linear regression prediction methods have been tested and compared within a newly developed fault detection system (Alzghoul, Löfstrand, & Backe, 2012).
- *Summarization* – Summarization is related to the effortlessly understandable presentation of data and to methodology that converts intricate data characteristics into explicit patterns that can make sense to users. It provides a more concise and intelligible representation of the data set, including visualization and report generation in digest form. A visual data mining approach can be suitable for building knowledge base in shop floor control systems of semiconductor wafer fabrication (Shiue, Guh, & Tseng, 2012).

Since we are interested in knowledge discovery techniques that can help overcome the challenge of defining procedural knowledge, the association rule learning is chosen as knowledge discovery technique in our framework. This technique is domain independent, and can be potentially applied for modeling procedural knowledge in any domain characterized by the fact that for a given task there might be many alternative solution-strategies with an extensive range of practical solutions (Nkambou, Fournier-Viger, & Mephu Nguifo, 2011). Therefore, our approach aims to use this knowledge discovery technique in authoring solutions for the industrial monitoring process.

Particularly, our work focused on the approach of association rule mining, which extracts knowledge from data sets and the knowledge discovered is represented by rules. At a very abstract level, knowledge can be represented by links between items, whereas items are facts or events. These links of items will be referred as rules. These rules can permit a system to order and organize its interaction with its environment, giving the possibilities for reasoning such as predicting events, and other analyses. Agrawal et al. first presented the concept of strong rules, where association rules are used to discover regularities between products (modeled by sets of items) in large-scale databases (Agrawal, Imielinski, & Swami, 1993).

The article is structured as follows: in Section 2 we present the proposed context of knowledge discovery with association rule mining (Section 2.1) and the algorithms associated with processing mechanisms (Section 2.3). Section 3 is devoted to the description of our methodology and the principles proposed to assist the quality analysis of the studied industrial process monitoring (Section 3.2). Finally, a conclusion is provided to illustrate the lessons learned and prospective work (Section 4).

2. Knowledge discovery with association rule mining

The procedure to set up an artificial intelligence is complex, highly dependent on its functional organization as well as on its environment. Our study focuses on expert system, data mining and rule extraction. The procedure to set up such a system is linked to Knowledge Discovery in Databases, to setting up a model and formalism, to execute an appropriately chosen Algorithm with suited parameters.

2.1. Knowledge Discovery in Databases (KDD) process

Data mining is a very important analysis activity of the Knowledge Discovery in Databases (KDD) process, which is an interdisciplinary field of computer science; this refers to a very broad process of finding knowledge in a large database. In order to find knowledge, a standard process has been developed, "The Knowledge Discovery in Databases process" (Fayyad, Piatetsky-Shapiro, & Smyth, 1996):

As seen on Fig. 1, the KDD process extracts knowledge from data in four different steps. The first step, selection, develops the understanding of the application domain, of the prior knowledge and the goals of the end-user. A target data is created; the selection of data in which the discovery will be performed. During the pre-processing step, the data is cleaned from noise and outliers; the necessary modeling information is collected. The data is transformed following the modeling information and a data-mining task is attributed (whether a classification, clustering, association rule mining, etc.). The data modeling is complete and a data-mining algorithm can be executed to discover patterns in large data sets. The resulting patterns are represented as rules, trees, or clustering. Mined patterns are interpreted in a user goal focus and knowledge is extracted.

More formally, the problem of ***association rule mining*** is stated as follows (Agrawal et al., 1993).

Let $I = \{a_1, a_2, \ldots, a_n\}$ be a finite set of items. A transaction database is a set of transactions $T = \{t_1, t_2, \ldots, t_m\}$ where each transaction $tj \subseteq I$ $(1 \leqslant j \leqslant m)$ represents a set of items. An itemset is a set of items $X \subseteq I$. The support of an itemset X is denoted as $sup(X)$ and is defined as the number of transactions that contain X. An association rule $X \rightarrow Y$ is a relationship between two itemsets X, Y such that X, $Y \subseteq I$ and $X \cap Y = \emptyset$. The support of a rule $X \rightarrow Y$ is defined as $sup(X \rightarrow Y) = sup(X \cup Y)/|T|$. The confidence of a rule $X \rightarrow Y$ is defined as $conf(X \rightarrow Y) = sup(X \cup Y)/sup(X)$. The *problem of mining association rules* is to find all association rules in a database having a support no less than a user-defined threshold *minsup* and a confidence no less than a user-defined threshold *minconf*. The problem of rule mining can be decomposed in two steps: Step 1 is to determine all frequent itemsets in the database (itemsets being present in at least $minsup \times |T|$ transactions). Step 2 is to discover association rules by using the frequent itemsets found in step 1. For each frequent itemset X, pairs of frequent itemsets P and $Q = X - P$ are carefully chosen to engender rules of the form $P \rightarrow Q$. For each such rule $P \rightarrow Q$, if $sup(P \rightarrow Q) \geqslant minsup$ *and* $conf(P \rightarrow Q) \geqslant minconf$, the rule is output.

A subset of the problem of ***association rule mining*** is the problem of ***mining sequential rules*** common to several sequences as follows (Fournier-Viger, Faghihi, Nkambou, & Mephu Nguifo, 2012). A *sequence database* SD is a set of sequences $S = \{s_1, s_2, \ldots, s_n\}$ and a set of items $I = \{i_1, i_2, \ldots, i_n\}$, where each sequence s_x is an ordered list of itemsets $s_x = \{X_1, X_2, \ldots, X_n\}$ such that $X_1, X_2, \ldots, X_n \subseteq I$. An item x is said to occur before another item y in a sequence $s_x = \{X_1, X_2, \ldots, X_n\}$ if there exists integers $k < m$ such that $x \in X_k$ and $y \in X_m$. A *sequential rule* $X \Rightarrow Y$ is defined as a relationship between two itemsets X, $Y \subseteq I$ such that $X \cap Y = \emptyset$ and X, Y are not empty. The interpretation of a *sequential rule* $X \Rightarrow Y$ is that if the items of X occur in some itemsets of a sequence, the items in Y will occur in some itemsets afterward in the same sequence. The *problem of mining sequential rules common to several sequences* is to find all sequential rules from a sequence database such that their support and confidence are respectively higher or equal to some user-defined thresholds *minSup* and *minConf*.

More generally, *frequent patterns* are itemsets, subsequences, or substructures that appear in a data set with frequency no less than a user-specified threshold. A substructure can refer to various structural forms, such as subgraphs, subtrees, or sublattices, which may be combined with itemsets or subsequences (Han, Cheng, Xin, & Yan, 2007). Frequent pattern mining plays an essential role in association rule mining. For instance, the design knowledge concerning a given task can be specified through frequent pattern mining used to search for frequently occurring design diagrams that are represented as attributed hierarchical layout hypergraphs encoding knowledge engaged for reasoning about design features

Fig. 1. The Knowledge Discovery in Databases (KDD) process (Fayyad et al., 1996).

(Strug & Ślusarczyk, 2009). Sequential pattern mining algorithms allows product and quality engineers to extract hidden knowledge from a large industrial database, since significant patterns provide knowledge of one or more product/process failures that leads to future product/process fault(s) (Buddhakulsomsiri & Zakarian, 2009).

2.2. The importance of formalism

The main issue and difficulties related to data mining come from the formalization of the input data and rule organization. In order to extract knowledge from an environment, information must be translated from a real context (real input data) to an abstract context (processed input data). The goal of this translation is to represent real events and facts through abstract events. These abstract events are represented in the database by entities that are understood by the processing platform these can be symbols or any other binary expression. In any environment this formalization is needed, it is highly dependent on the goals of the system. It is of most importance that the environment boundaries and formalism complexity are defined with the goals of the system.

The success of the system relies on the formal system. False definitions and data transformations can result in a data loss (or in the opposite too much data is selected). Some rules may not appear or be false, unwished knowledge may also be extracted.

2.3. Choosing an appropriate algorithm

Once the formalism is set up and input data is pre-processed, the algorithm can be executed. Based on the definition of mining association rules, most studies take frequent pattern mining as the first step which precedes the second step generating rules from frequent itemsets. However, this first step is computationally expensive process and not all the association rules so generated are interesting. As a result, several algorithms have been developed over time (a review of frequent pattern mining algorithms is described in (Tiwari, Gupta, & Agrawal, 2010)). Much has been written about the advantages and disadvantages of their programs. The main existing algorithms are the *Apriori*, *Eclat* and *FP-Growth* algorithms (or lightweight and hybrid variants thereof). They mine frequent itemsets. Besides, these algorithms have two parameters, support and confidence. They are specified by the user and enable result filtering by the algorithm. These parameters when well defined filter only important association rules from the system.

- *The Apriori algorithm*:

The Apriori algorithm (Agrawal et al., 1993) finds frequent itemsets from databases by iteration. At each iteration i the algorithm attempts to determine the set of frequent patterns with i items and this set is engaged to generate the set of candidate itemsets of the next iteration. The iteration is repetitively performed until no candidate patterns can be discovered. It uses a bottom-up approach, where frequent subsets are extended one item at a time. In the input datasets are referred as sequences composed of more or less items. The output of Apriori is a set of rules explaining the links these items have in their sets.

- *The Eclat algorithm*:

The *Eclat* algorithm (Zaki, 2000) uses a depth-first search and finds links between itemsets (between sequences). It is recursively structured and uses item intersection to compute the support of an itemset avoiding the generation of non-existing item patterns. The three main ideas behind the Eclat algorithm are specially: (i) generation of every possible 2-itemset whether or not it occurs in the database, (ii) search space partitioning using equivalence classes, which is very convenient for sake of enumerating itemsets with a particular item, and (iii) the vertical dataset layout approach to achieve support counting, which is more suitable to lazy rule generation approach.

- *The FP-Growth algorithm*:

The *FP-Growth* (frequent pattern growth) algorithm (Han, Pei, Yin, & Mao, 2004) uses a prefix-tree (FP) data structure to store compressed and crucial information about frequent items of the database. The FP-Growth algorithm recursively establishes conditional parameters and from the FP-tree structure and uses them to generate the full set of frequent patterns. The mining task as well as the database are decomposed using a divide and conquer system and finally it uses a fragment pattern method to avoid the costly process of candidate generation and testing opposed to the Apriori algorithm.

- *Comparison of three algorithms*:

These three algorithms are used all over the world on different applications, and are well known. Apart from its FP-tree, the FP-growth algorithm is very analogous to *Eclat*, but it uses some additional steps to maintain the FP-tree structure during the recursion steps, while *Eclat* only needs to maintain the covers of all generated itemsets. The simple difference between *Eclat* and *FP-growth* is the way they count the support of every candidate itemset and how they represent and maintain the *i*-projected database. As a comparison, *Eclat* basically generates candidate itemsets using only the join step from *Apriori*, since the itemsets necessary for the prune step are not available. If the transaction database contains a lot of large transactions of frequent items, such that *Apriori* needs to generate all its subsets of size 2, *Eclat* still outperforms *Apriori*. For very low support thresholds or sparse datasets, *Eclat* clearly outperforms all other algorithms. The main advantage FP-*growth* has over *Eclat* is that each linked list, starting from an item in the header table representing the cover of that item, is stored in a compressed form. The *Apriori* and *FP-Growth* Algorithms extract rules from a database but use two different approaches, where *Apriori* computes all possibilities; FP-Growth uses a prefix-tree structure to simplify computing. The heavy algorithm *Apriori* may give interesting results, but FP-growth is about an order of magnitude faster than *Apriori*, specifically with a dense data set (containing many patterns) and/or with long frequent patterns (Goethals, 2010, chap. 16). It is important while implementing an association rule learning system to study performance indicators. These algorithms are complex and the overall data-mining task is heavy in computing and memory consumption. The execution speed and the memory consumption are two performance indicators and should always be calculated.

Unlike other algorithms, the approach that uses the FP-tree structure to discover sequential rules is more efficient and scalable on both synthetic data and real-life data (Hu & Chen, 2006). Especially, for the problem of ***mining sequential rules*** common to several sequences, the Pattern-Growth approach could be particularly valuable in managing complex tasks such as monitoring the state and quality of materials resources in industrial operational processes. For that reason, we use the RuleGrowth algorithm (Fournier-Viger, Nkambou, & Tseng, 2011) relying on a Pattern-Growth, in order to discover a more general form of sequential rules such that items in the antecedent and in the consequent of each rule are unordered. This form of sequential rules conveys more information and it is not discovered by other approaches stating that items of the left part or the right part of a rule have to appear with exactly the same ordering in a sequence (Lo, Khoo, & Wong, 2009). RuleGrowth first find rules between two items and then recursively grow them by scanning the database for single items that could expand their left or right parts.

The association rule learning procedure can be applied to any already existing large database or any real time event stream. In that case near future events can be predicted. The system analyses a live data stream (returned by a translation unit who processes from a captor environment) and can detect rule occurrences and predict its consequence. This type of system can be used as a problem detecting system and can enable preventive actions.

3. A sequential rule mining approach for industrial process monitoring

3.1. Introduction to our manufacturing example

The manufacturing example takes place at Vam Drilling (a part of the oil and gas division of Vallourec & Mannesmann Tubes, which is a subsidiary of the Vallourec Group) at Tarbes (south-west of France). VAM Drilling manufactures drill pipe and associated drill-stem products (e.g. drilling tubulars, drilling tool joints, drill collars) for oil and gas extraction. VAM Drilling continually reshapes itself to provide the right expertise in the right place, by emphasizing the engagement of quality drilling products. At VAM Drilling, the sales, marketing, and production teams continually work together to understand client expectations and to adapt the production process and services to fulfill requirements: (i) providing safe products with high performance values, (ii) having efficient and timely production processes with quick delivery times and (iii) demonstrating stringent regulatory compliance to standards.

3.1.1. Quick process introduction

This diagram from Fig. 2 shows us the manufacturing process of the company. The first operation is to cut steel bars of 9 m long to small plots that must meet a predetermined weight (determined by the planning department) so the material is optimized for forging. Then the slugs are heated at 1150 °C to be in a semi-malleable state, enabling to forge them. Presses of 1500 tons and 320 tons drill and conform parts. The heat treatment allows us to obtain the required mechanical characteristics. Finally the machining activity makes the final form.

The use of quality monitoring and industrial engineering techniques for continuous improvement and manufacturing process controls form the basis of the Vam Drilling management system. For example, throughout its assembly, drill pipe can undergo as many as 20 levels of inspection and testing to guarantee products meet performance requirements (e.g. dimensional, visual, magnetic particle, ultrasonic and mechanical evaluations).

3.1.2. Know difficulties and problems

Among the different elements of the manufacturing process the forge is the most important one because it's the bottleneck of the manufacturing process. The forge process can't be subcontracted. This is why a quality management procedure is applied to the forge process in order to improve it. The first part of the procedure is to check the forge's performance. This is why they asked an intern to implement some performance indicators. These indicators measure performance and show the dysfunctions. Finally the intern, showing the most important delay causes, generated a Pareto of causes.

3.2. Setting up goals

We want to use an AI system to sort the dysfunctions causes. This means we want to prove this system can do the same work as an engineer but with some constraints. To do this we need to

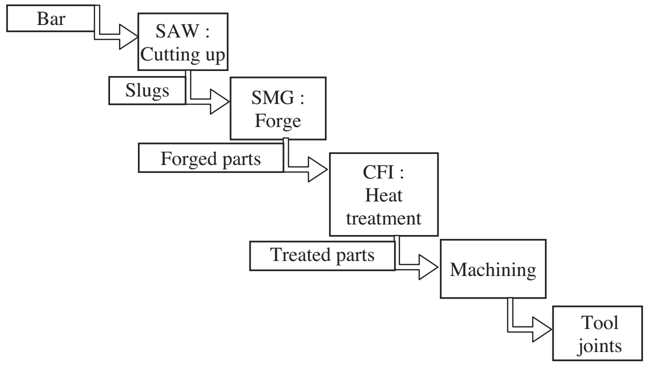

Fig. 2. Manufacturing process of the studied case.

first set up an environment, assign a formalism and create input data. Then we will execute an algorithm with the input data. We will choose a *minsup* (minimum support) and *minconf* (minimum confidence) to have good rule ranking conditions.

The goal is to set up the system so its results will be as close as possible to the internship results. We will focus on the consequences of dysfunctions, mainly on the amount of lost manufacturing time as well as the dysfunction appearance frequency. Indicator measurements from the internship give the information linking dysfunctions and their consequences, but such information must be translated into our formalism.

3.3. Creating a formalism

The graph from Fig. 3 shows us various part changes, with their tool change times and start-up times during the week 21 of 2011. We will focus on the start-up times (blue bar)[1] because they are the main manufacturing lost time cause. We will also analyze the dysfunction causes (written under the blue bars).

We need to translate this information into input data. We decided that each sequence in the input data would be the appearance of one start-up time. These sequences of events appear for each start-up (blue bar) and contain information such as the part family, the duration of the eventual dysfunction and its causes. All of this is represented as various events. The events representing dysfunctions can be separated in part problems and maintenance problems.

3.3.1. Overall organization

First we create a formalism structure with event families. For our application we have developed three event families:

- "Part Size". This event family translates the manufactured part size into size events (as shown in Table 2):
- "Dysfunctions Occurrence". This event family translates dysfunctions into dysfunctions events (as shown in Table 3). There are 23 retained dysfunctions translated into 23 events:
- "Start-up Delays". This event family translates real time delays into delay events. We have three different delay events (as shown in Table 4):

3.4. Creating the input data

The usage of data mining for analyzing industrial processes is also enlarged thanks to the set up focusing on real industrial data. Data are translated automatically into event families using the

[1] For interpretation of color in Fig. 3, the reader is referred to the web version of this article.

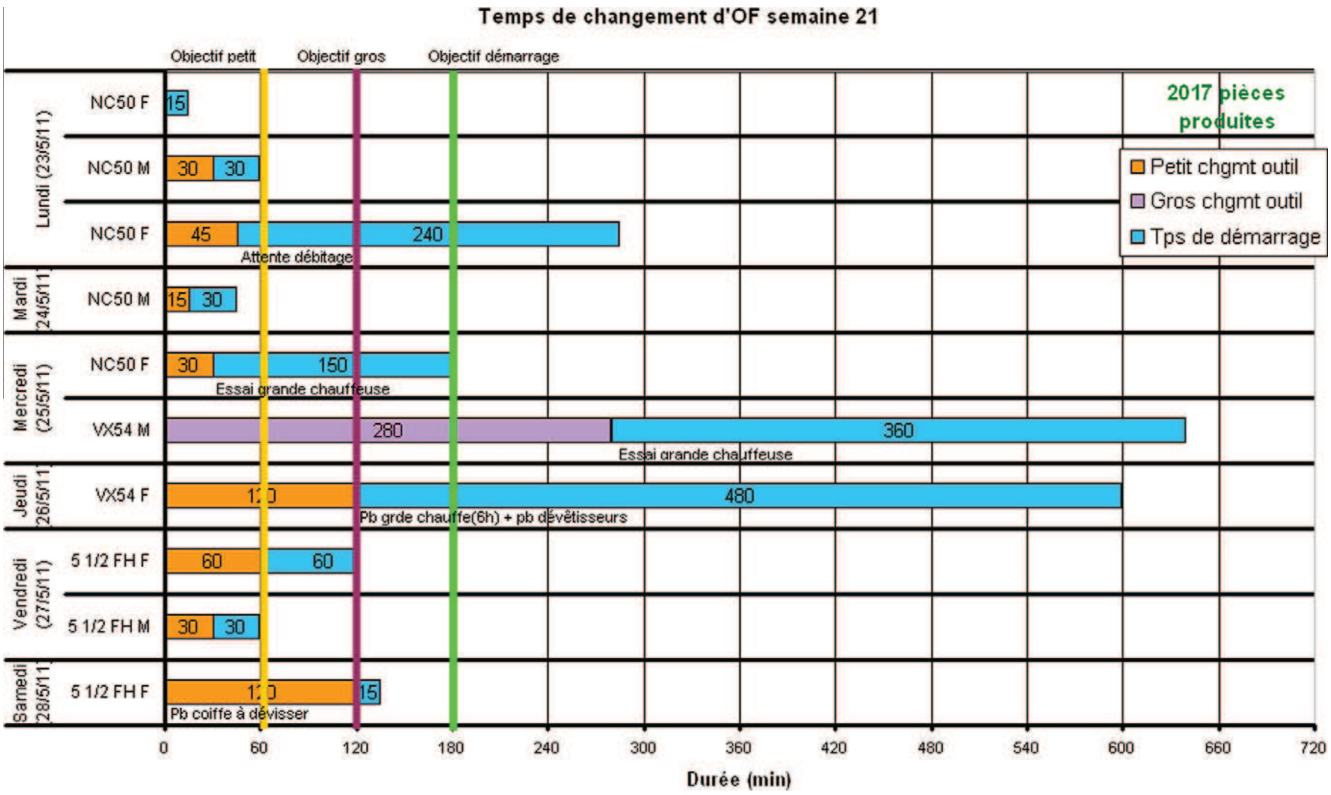

Fig. 3. Forge start-up time indicators during a week.

Table 2
Part size event family.

Number	Event	Explication
1	Small	Small parts are being manufactured
2	Medium	Medium parts are being manufactured
3	Big	Big parts are being manufactured

Table 3
Dysfunction event family.

Number	Event	Explication
1	Dys1	Induction furnace dysfunction
2	Dys2	Stripper dysfunction
3	Dys3	Scrap sticking to the tool
4	Dys4	Tool off-center
5	Dys5	Robot manipulator dysfunction
6	Dys6	Press 320 tons dysfunction
7	Dys7	Waiting cutting up

requested structure for data files and record format. Out of 13 weeks of real data, we translated 118 sequences of event with approximately nine sequences per week.

Table 5 represents the input data translated from the real data from Fig. 3.

3.5. Result description

3.5.1. Non-interesting and exploitable results

We can now process the data input with an algorithm and extract rules. We did this with RuleGrowth that mines sequential rules common to several sequences by FP-Growth (frequent pattern growth), with the parameters $minSup = 0.06$ and $minConf = 1$ (minimum 6% frequency and 100% reliability).

In Table 6, the rules in blue are linked to the formalism. As an example the first line shows the rule: if we have at least 10 h delay

Table 4
Delay event family.

Number	Event	Explication
1	Delay1 h	At least 1 h longer than expected
2	Delay4 h	At least 4 h longer than expected
3	Delay10 h	At least 10 h longer than expected

then we have at least 4 h delay (**Delay10 h** → **Delay4 h**). These rules depend on the chosen formalism and have little interest as results expect to validate our formalism and to show delay event frequency (through the support). For example the rule **Delay10 h** → **Delay4 h** has a support of 6%, this shows us the frequency of the event: **Delay10 h** (at least 10 h delay). The same reasoning is applied to the rule **Delay4 h** → **Delay1 h** and indicates that the event **Delay4 h** appears 23% of the time when **Delay1 h** occurred. These rules are called Non-Interesting because they are not directly related to the goal.

In Table 6, the green rules translate knowledge extracted from the environment. For example the third line or sequence translates the following knowledge: each time we have the cause **Dys5** production is delayed of 1 h (**Delay1 h**). The algorithm's execution with $minSup = 0.06$ and $minConf = 1$ as parameters, gave us four exploitable rules, this is judged as insufficient.

We re-executed the algorithm with other support and confidence parameters to see how they influence our results. We chose a minimum support of 0,05 and a minimum confidence of 0,8, so the algorithm will seek rules that are true 80% of the time and appear 5% of the time. The results are presented on Table 7.

This new algorithm execution gave us 13 exploitable rules, allowing us to sort the causes more efficiently because another level of delay appears in the output data. On line 6, a delay of at least 4 h appears with the dysfunction **Dys1** 82% time. This causes the longest delay.

Table 5
Extract of the input data.

Number	Sequence	Explication
1	Medium	No dysfunction occur, no delay
2	Medium, Dys7, Delay1 h	Between one and 4 h of delays caused by the waiting cutting up
3	Medium	No dysfunction occur, no delay
4	Medium, Dys1, Delay1 h	Between one and 4 h of delays caused by the induction furnace dysfunction
5	Medium, Dys1, Delay1 h, Delay4 h	Between 4 and 10 h of delays caused by the induction furnace dysfunction
6	Medium, Dys1, Dys2, Delay1 h, Delay4 h	Between 4 and 10 h of delays caused by the induction furnace dysfunction and the stripper dysfunction
7	Medium	No dysfunction occur, no delay
8	Medium	No dysfunction occur, no delay
9	Medium	No dysfunction occur, no delay

Table 6
Algorithm results with *minSup* = 0.06 and *minConf* = 1.

Number	Rules	Support	Confidence
1	Delay10h → Delay4h	0.065	1
2	Delay10h → Delay1h	0.065	1
3	Dys5 → Delay1h	0.065	1
4	Dys3 → Delay1h	0.083	1
5	Dys4 → Delay1h	0.093	1
6	Dys1 → Delay1h	0.157	1
7	Delay4h → Delay1h	0.231	1
8	Delay10h, Delay4h → Delay1h	0.064	1
9	Delay10h, Delay1h → Delay4h	0.064	1
10	Delay10h → Delay1h, Delay4h	0.064	1
11	Dys1, Delay4h → Delay1h	0.130	1
12	Medium, Dys1 → Delay1h	0.083	1
13	Big, Delay4h → Delay1h	0.102	1
14	Medium, Delay4h → Delay1h	0.102	1

Table 7
Results with *minConf* = 0.8 and *minSup* = 0.05.

Number	Rules	Support	Confidence
1	Delay10h → Delay4h	0.065	1
2	Delay10h → Delay1h	0.065	1
3	Dys5 → Delay1h	0.065	1
4	Dys3 → Delay1h	0.083	1
5	Dys4 → Delay1h	0.093	1
6	Dys1 → Delay4h	0.130	0.82
7	Dys1 → Delay1h	0.157	1
8	Delay4h → Delay1h	0.231	1
9	Delay10h, Delay4h → Delay1h	0.064	1
10	Delay10h, Delay1h → Delay4h	0.064	1
11	Delay10h → Delay1h, Delay4h	0.064	1
12	Medium, Dys4 → Delay1h	0.056	1
13	Big, Dys1 → Delay1h	0.056	1
14	Big, Dys1 → Delay4h	0.056	1
15	Dys1, Delay4h → Delay1h	0.130	1
16	Dys1, Delay1h → Delay4h	0.130	0.82
17	Dys1 → Delay1h, Delay4h	0.130	0.82
18	Medium, Dys1 → Delay1h	0.083	1
19	Big, Delay4h → Delay1h	0.102	1
20	Medium, Delay4h → Delay1h	0.102	1
21	Big, Dys1, Delay4h → Delay1h	0.056	1
22	Big, Dys1, Delay1h → Delay4h	0.056	1
23	Big, Dys1 → Delay1h, Delay4h	0.056	1
24	Medium, Dys1, Delay4h → Delay1h	0.056	1

Table 8
Results with *minsup* = 0.04 and *minconf* = 0.6.

Number	Rules	Support	Confidence
1	Dys6 → Delay1h	0.046	1
2	Delay10h → Delay4h	0.065	1
3	Delay10h → Delay1h	0.065	1
4	Dys5 → Delay1h	0.065	1
5	Dys3 → Delay1h	0.083	1
6	Dys4 → Medium	0.056	0.6
7	Dys4 → Delay1h	0.093	1
8	Dys1 → Delay4h	0.130	0.82
9	Dys1 → Delay1h	0.157	1
10	Delay4h → Delay1h	0.231	1
11	Big → Delay1h	0.167	0.67
12	Delay10h, Delay4h → Delay1h	0.065	1
13	Delay10h, Delay1h → Delay4h	0.065	1
14	Delay10h → Delay1h, Delay4h	0.065	1
15	Dys3, Delay4h → Delay1h	0.046	1
16	Medium, Dys3 → Delay1h	0.046	1
17	Medium, Dys4 → Delay1h	0.056	1
18	Dys4, Delay1h → Medium	0.056	0.6
19	Dys4 → Medium, Delay1h	0.056	0.6
20	Big, Dys1 → Delay1h	0.056	1
21	Big, Dys1 → Delay4h	0.056	1
22	Medium, Dys1 → Delay4h	0.056	0.67
23	Dys1, Delay4h → Delay1h	0.130	1
24	Dys1, Delay1h → Delay4h	0.130	0.82
25	Dys1 → Delay1h, Delay4h	0.130	0.82
26	Medium, Dys1 → Delay1h	0.083	1
27	Big, Delay1h → Delay4h	0.102	0.6
28	Big, Delay4h → Delay1h	0.102	1
29	Medium, Delay4h → Delay1h	0.102	1
30	Big, Dys1, Delay4h → Delay1h	0.056	1
31	Big, Dys1, Delay1h → Delay4h	0.056	1
32	Big, Dys1 → Delay1h, Delay4h	0.056	1
33	Medium, Dys1, Delay1h → Delay4h	0.056	0.67
34	Medium, Dys1, Delay4h → Delay1h	0.056	1
35	Medium, Dys1 → Delay1h, Delay4h	0.056	0.67

Table 9
Results after deductive reasoning.

Dysfunction	Occurrences (%)	Delay1 h occurrences because of dysfunction (%)	Delay1 h occurrences when the dysfunction occurs (%)	Delay4 h occurrences because of dysfunction (%)	Delay4 h occurrences when the dysfunction occurs (%)	Delay10 h occurrences because of dysfunction (%)	Delay10 h occurrences when the dysfunction occurs (%)
Dys1	15.7	2.8	18	12.9	82	0	0
Dys3	8.3	3.7	45	4.6	55	0	0
Dys4	9.2	9.2	100	0	0	0	0
Dys5	6.4	6.4	100	0	0	0	0
Dys6	4.6	4.6	100	0	0	0	0
Total	44.2	26.7	60.4	17.5	39.6	0	0

Table 10
Simplified results after deductive reasoning.

Dysfunction	Criticality	% Of criticality
Dys1	54.4 (=2.8 + 4 ∗ 12.9)	56.3% (=54.4/96.7)
Dys3	22.1 (=3.7 + 4 ∗ 4.6)	22.9%
Dys4	9.2	9.5%
Dys5	6.4	6.6%
Dys6	4.6	4.8%
Total	96.7	100%

3.5.2. Influence of the min support and confidence

Decreasing the minimum support and confidence will increase the number of exploitable rules. As a matter of fact the support represents the frequency of appearance of the causes of a rule and the confidence is the ratio of appearance of the consequence of a rule after the causes appeared. For example in Table 7, the knowledge represented on line 6 is the same as line 16 and 17. This is also true for lines 14, 22 and 23 whereas line 13 is not interesting. This information repetition is due to the formalism (**Delay10 h** → **Delay1 h**, **Delay4 h**; **Delay4 h** → **Delay1 h**) and generates non-interesting rules.

We processed the data again with *minsup* = 0.04 and a *minconf* = 0.6 as parameters, results appear on Table 8.

This new execution extracted additional knowledge, line 15 shows that the rule **Dys3**, **Delay4 h** → **Delay1 h** has a support of 0.046 which means that the rule **Dys3** → **Delay4 h** occurs 4.6% of the time. Line 5 shows that the rule **Dys3** → **Delay1 h** has a support of 0.083 (so it occurs 8.3% of the time), we can conclude from this that when the dysfunction **Dys3** occurs, it will cause a delay of at least 4 h in 55.4% of the time (0.046/0.083 = 0.554).

The choice of the thresholds *minsup* and *minconf* which clearly influences various points of the resolution and the quality of the rules generated by algorithms:

- If set too high, then algorithms generate too few results, omitting valuable information,
- If set too low, then algorithms can generate an extremely large amount of results and can become very slow.

One of the main difficulties users encounter, is to set up the algorithms parameters (thresholds *minsup* and *minconf*) in a way where there is a desired amount of rules. To propose a solution to these difficulties (Fournier-Viger, Wu, & Tseng, 2012) developed the TopKRules algorithm. This algorithm takes two parameters (*k* the number of rules to be generated and *minconf*), employs a rule expansion approach and provides the top-*k* association rules in which users have considerable interest. The rule expansion approach finds larger rules by recursively scanning the database for adding a single item at a time to the left or right part of each rule (these processes are called left and right expansions). The main idea is to always find first the most promising rules with higher support and then we can raise *minsup* more quickly and prune the search space. Finally, the algorithm mines the top-*k* rules using a user hidden support calculation and verifies that the found rules respect the user giving confidence. Experimental results show that the top-*k* association rule algorithm has excellent performance and scalability (execution time linearly increases with *k*), and that it is an advantageous alternative to classical association rule mining algorithms when the user wants to control the number of association rules generated. However when the user knows the size and structure of the database and therefore knows the optimal minimum support and confidence, the TopKRules algorithm is slower than a classic data mining algorithm.

3.6. Analyzing results and extracting knowledge

In order to fully extract knowledge from these results we need an additional reasoning step. This step links the obtained rules to our goals through a deductive reasoning. This step should be executed by another algorithm; in this study we did it manually.

Five main dysfunctions appear in our results (**Dys1**, **Dys3**, **Dys4**, **Dys5** and **Dys6**). First we extract the frequency of occurrences of

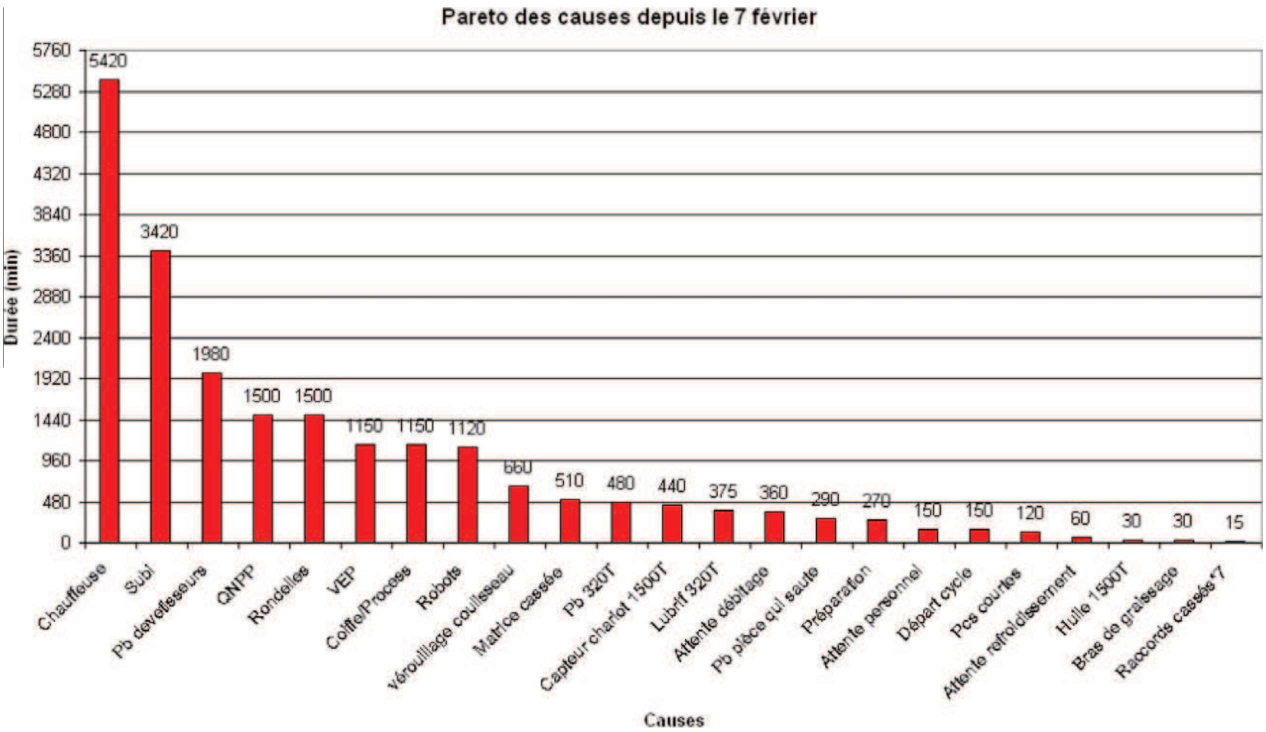

Fig. 4. Pareto of root causes of operational underperformance.

these dysfunctions, then the frequency of the different delay events they cause.

This deductive reasoning will be explained on event **Dys1** (dysfunction of the induction furnace):

- *Frequency of occurrences of Dys1*: We use the support the rule **Dys1 → Delay1 h** (=15.7%).
- *Frequency of the Delay1* **h** *only when dysfunction 4 occurs*: We use the support of the rules **Dys1 → Delay1 h**, **Dys1 → Delay4 h**, **Dys1 → Delay10 h** (=15.7%; =12.9%; =0%), 15.7–12.9–0 = 2.8%.
- *Frequency of the Delay4* **h** *only when Dys1 occurs*: We use the support of the rules **Dys1 → Delay4 h** and **Dys1 → Delay10 h** (=12.9%; =0%), 12.9–0 = 12.9%.
- *Frequency of the Delay10* **h** *only when Dys1 occurs*: Doesn't occur because rule **Dys1 → Delay10h** has a support of 0% (it does not appear).

We can also extract that when **Dys1** occurs, 12.9/15.7 = 82.16% of the time it will cause at least 4 h delay (**Delay4 h)**.

The results in Table 9 show that one or more of these five dysfunctions will occur in 44.2% of the time, if they occur they cause a delay of 1–4 h in 60.4% of the time and 39.6% of the time they cause between 4 and 10 h delay. A delay between 1 and 4 h will occur 26.7% of the time and a delay between 4 and 10 h will occur 17.5% of the time.

In our industrial example, if we consider delay 4 h to be four times more problematic than delay 1 h, we can simplify the results by using criticality parameter, including additive costs associated to these delays.

The results on Table 10 show that the dysfunction **Dys1** represents 56.3% of the delays' costs and is therefore the main cost cause. It is now possible using the chart on Table 10 to obtain a Pareto of causes (Fig. 4). These results are only estimations due to the presence of a minimum support and confidence; nevertheless these minimum support and confidence are needed to limit the number of rules. Association rule mining can have more impact because the user is engaged and interested, ready for results and willing to move those results into practice because they are of direct relevance to their day-to-day lives. Other than this effective association rule mining method, the general principle of greatest interest for industrial process monitoring is that on sustainable continuous improvement.

Further analysis of the data for monitoring quality of operations in Fig. 4 shows that the generator is the first cause for exceeding the maximum time in starting phase. This needs to be fixed for further verification, after it is repaired and improved. One of the main causes of defects was identified: after slugs passing through the generator, superficial oxides or calamine stands and stagnates in the heating inductors. A good protective coating of ferrous material is commonly obtained after removing the calamine and offering a clean surface. Successful corrosion management processes require appropriate tools (risk-based assessments, mitigation/corrosion control/inspection/monitoring, and data collection/interpretation (Dawson, 2010)). The second major problem is the sudden changes. These often result in changes either big or small to manufacturing activities and resources in ways that are both obvious and subtle. Sudden changes in management approaches can have very serious impacts on the quality monitoring of manufacturing processes. The third major problem is the lack of effectiveness of metal strippers, which must equipped so that the piece being machined can be placed and guided in safety. It is likely some connection or pressure problems related to the metal stripper settings. In addition the strippers were worn beyond manufacturer's specifications and thereby the jaws of the strippers should be standardized.

Quality is a fundamental part of VAM Drilling's process and requires avoiding the recurrence of high delays, so it is important to ensure that effective maintenance capacity continues to match the growing quality expectations. It is a good idea to perform routine maintenance tasks to ensure the manufacturing process is reliable and in good repair by using quality drilling products. Preventative and routine maintenance are crucial to prevent material resource downtime and time-consuming cleaning and corrective maintenance work in drilling product manufacturing facilities. Regardless of the respective legal requirements, regular safety inspections guarantee compliance with safety and quality standards, serve as precautionary maintenance measures and consequently help to reduce undesirable material resource downtimes to a minimum. One of the challenges is in utilizing existing and novel methods or technologies to achieve an appropriate level of manufacturing process control while maintaining the desired product quality attributes. Lessons learned from knowledgeable and experienced production teams can help to design, install and carry out planned routine maintenance programmes and plant operation which leave the human resources free to focus on their core activities. Accordingly, taking all of the above factors into account, in the context of the continuous improvement the data mining for quality control is useful to optimize the industrial process and reduce economic cost (Ferreiro, Sierra, Irigoien, & Gorritxategi, 2011).

4. Conclusion

Our application focused on data mining and knowledge discovery, these Artificial Intelligence (AI) sub-fields are actually one of the closest to an industrial application. Through existing studies, application and algorithms we acquired an AI system implementing procedure. We applied this implementing procedure to an industrial case on real data, in order to really understand the possibilities and limitations of our approach. The strategy addresses not only the building and upgrading of association rule mining facilities but also includes effective manufacturing process quality monitoring combined and sustainable operation and continuous improvement processes. In addition, the proposal will benefit fully from innovative new execution approaches using the reliability enhancement program, a continuous improvement program based on the analysis of completed projects.

The reasoning behind our approach is divided in two steps: first an inductive reasoning were knowledge is extracted from the input data through the data mining procedure, and then a deductive reasoning that uses the extracted knowledge and links it to the system goals. By comparing the results of our approach and the real results from the internship we estimated the capability of mining association rules in such an example. While there is still a need for improvement, we note that the quality of information being provided has improved. At the same time, the system still needs other processing tools, such a rule filtering and processing units to execute the deductive reasoning part. Our conclusions indicate that an approach can be implemented in various industrial applications through suitable contextual adaptations.

Despite such positive developments, it is important to have a better mechanism to identify complementarities and build stronger working relationships with domain experts. Expert elicitation yields expert knowledge, but also it incorporates the analyst as assessor knowledge that engages a judgmental process of selective querying of acquired knowledge in risk assessment (Aven & Guikema, 2011). An intelligent architecture of interactive data mining method can be used as a powerful tool for emulating cognitive process of human analysts (Shu, 2007). Also, it is interesting to incorporate methodological commonalities in intelligent production research for adaptive control optimization of production processes (Kruger, Shih, Hattingh, & van Niekerk, 2011). In fact, expert knowledge and data mining discovered knowledge can cooperate and complement each other in the investigation, analysis and further problem solving activities of complex situations (Kamsu-Foguem et al., 2012). Several types of cooperation between them are possible (Alonso, Martínez, Pérez, & Valente, 2012):

- Expert functions remove incorrect tests, eliminate incorrect extensions and remove noise before applying the numerical data mining for pattern discovery.
- Expert knowledge is used to select and validate the relevant patterns from candidate patterns discovered by the numerical data mining system.
- Expert knowledge is engaged for guidance at the beginning of the reference model generation by selecting the population to be manipulated.

Indeed, the aptitude of learning routinely procedural knowledge could facilitate the formalization of problem spaces (task model, executive knowledge and practices, etc.) and moderate the need for domain experts (Kamsu-Foguem, 2012). Furthermore, ontologies can take action as a semantically rich knowledge base (Mikroyannidis & Theodoulidis, 2010) and they give additional information useful to guide the selection of procedural knowledge, in order to present only those which are of interest to the domain experts (Mansingh, Osei-Bryson, & Reichgelt, 2011).

A more effective interaction would be highlighted for defining association rule templates (that describe "flavors" of interesting and uninteresting rules) and facilitating a collaborative interpretation of results (Brossette & Hymel, 2008). Besides, a user oriented description and multiple criteria decision aid can be incorporated into the recommendation process of one or more user-adapted interestingness measures for association rules (Lenca, Meyer, Vaillant, & Lallich, 2008). A standardization of the environment structure, of the formalisms and of the goals would certainly provide an easier implementation by simplifying preliminary work and would allow other industrial applications.

References

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining associations between sets of items in large databases. In *ACM SIGMOD int'l conf. on management of data* (pp. 207–216), Washington D.C., May 1993.

Alonso, F., Martínez, L., Pérez, A., & Valente, J. P. (2012). Cooperation between expert knowledge and datamining discovered knowledge: Lessons learned. *Expert Systems with Applications, 39*(8), 7524–7535.

Alzghoul, A., & Löfstrand, M. (2011). Increasing availability of industrial systems through data stream mining. *Computers & Industrial Engineering, 60*(2), 195–205.

Alzghoul, A., Löfstrand, M., & Backe, B. (2012). Data stream forecasting for system fault prediction. *Computers & Industrial Engineering, 62*(4), 972–978.

Aven, T., & Guikema, S. (2011). Whose uncertainty assessments (probability distributions) does a risk assessment report: The analysts' or the experts'? *Reliability Engineering & System Safety, 96*(10), 1257–1262.

Benjamins, V. R., & Fensel, D. (1998). Editorial: Problem-solving methods. *International Journal of Human-Computer Studies, 49*(4), 305–313.

Brossette, S. E., & Hymel, P. A. Jr., (2008). Data mining and infection control. *Clinics in Laboratory Medicine, 28*(1), 119–126.

Buddhakulsomsiri, J., & Zakarian, A. (2009). Sequential pattern mining algorithm for automotive warranty data. *Computers & Industrial Engineering, 57*(1), 137–147.

Chiodo, A., Rosenhauer, E., & Worsowicz, G. (2011). Performance improvement methods: Not just for maintenance of certification. *PM&R, 3*(2), 163–167.

Choo, A. S., Linderman, K. W., & Schroeder, R. G. (2007). Method and context perspectives on learning and knowledge creation in quality management. *Journal of Operations Management, 25*(4), 918–931.

Dawson, J. L. (2010). Corrosion management overview. *Shreir's Corrosion, 4*(Part VI), 3001–3039.

Detlor, B. (2010). Information management. *International Journal of Information Management, 30*(2), 103–108.

Evans, J. R., & Lindsay, W. M. (2001). *The management and control of quality* (5th ed.). Cincinnati, OH: South-Western.

Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine, 17*(3), 37–54.

Ferreiro, S., Sierra, B., Irigoien, I., & Gorritxategi, E. (2011). Data mining for quality control: Burr detection in the drilling process. *Computers & Industrial Engineering, 60*(4), 801–810.

Foster, S. T. Jr., (2008). Towards an understanding of supply chain quality management. *Journal of Operations Management, 26*(4), 461–467.

Fournier-Viger, P., Nkambou, R., & Tseng, V. S. (2011). RuleGrowth: Mining sequential rules common to several sequences by pattern-growth. In: Proceedings of the 26th symposium on applied computing (ACM SAC 2011). ACM Press, pp. 954–959.

Fournier-Viger, P., Faghihi, U., Nkambou, R., & Mephu Nguifo, E. (2012). CMRules: Mining sequential rules common to several sequences. *Knowledge-Based Systems, 25*(1), 63–76.

Fournier-Viger, P., Wu, C. -W., & Tseng, V. S. (2012). Mining top-K association rules. In: Proceedings of the 25th Canadian conf. on artificial intelligence (AI 2012), Springer, LNAI 7310, pp. 61–73.

Gibbons, S. (2000). Down to a system: Keeping employee morale and retention high. *Journal for Quality and Participation, 23*(2), 20–22.

Goethals, B. (2010). Frequent set mining. In O. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook* (2nd ed., pp. 321–338). New York (NY), USA: Springer.

Han, J., Cheng, H., Xin, D., & Yan, X. (2007). Frequent pattern mining: Current status and future directions. *Data Mining and Knowledge Discovery, 15*(1), 55–86.

Han, J., Pei, J., Yin, Y., & Mao, R. (2004). Mining frequent patterns without candidate generation. *Data Mining and Knowledge Discovery, 8*, 53–87.

Hu, Y.-H., & Chen, Y.-L. (2006). Mining association rules with multiple minimum supports: A new mining algorithm and a support tuning mechanism. *Decision Support Systems, 42*(1), 1–24.

Huang, T. C.-K., Liu, C.-C., & Chang, D.-C. (2012). An empirical investigation of factors influencing the adoption of data mining tools. *International Journal of Information Management, 32*(3), 257–270.

Jabrouni, H., Kamsu-Foguem, B., Geneste, L., & et Vaysse, C. (2011). Continuous improvement through knowledge-guided analysis in experience feedback. *Engineering Applications of Artificial Intelligence, 24*(8), 1419–1431.

Kamsu-Foguem, B. (2012). Knowledge-based support in Non-Destructive Testing for health monitoring of aircraft structures. *Advanced Engineering Informatics.* http://dx.doi.org/10.1016/j.aei.2012.06.006.

Kamsu-Foguem, B., Coudert, T., Geneste, L., & Beler, C. (2008). Knowledge formalization in experience feedback processes: An ontology-based approach. *Computers in Industry, 59*(7), 694–710.

Kamsu-Foguem, B., Tchuenté-Foguem, G., Allart, L., Zennir, Y., Vilhelm, C., Mehdaoui, H., et al. (2012). User-centered Visual Analysis using a Hybrid Reasoning Architecture for Intensive Care Units. *Decision Support Systems.* http://dx.doi.org/10.1016/j.dss.2012.06.009.

Kantardzic, M. (2011). Data mining: Concepts, models, methods, and algorithms. 2nd revised ed. Wiley-Blackwell (an imprint of John Wiley & Sons Ltd). ISBN-10: 0470890452.

Köksal, G., Batmaz, İ., & Testik, M. C. (2011). A review of data mining applications for quality improvement in manufacturing industry. *Expert Systems with Applications, 38*(10), 13448–13467.

Kruger, G. H., Shih, A. J., Hattingh, D. G., & van Niekerk, T. I. (2011). Intelligent machine agent architecture for adaptive control optimization of manufacturing processes. *Advanced Engineering Informatics, 25*(4), 783–796.

Lee, S., Ryu, K., Shin, M., & Cho, G.-S. (2012). Function and service pattern analysis for facilitating the reconfiguration of collaboration systems. *Computers & Industrial Engineering, 62*(3), 794–800.

Lenca, P., Meyer, P., Vaillant, B., & Lallich, S. (2008). On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid. *European Journal of Operational Research, 184*(2), 610–626.

Liao, S-H., Chu, P-H., & Hsiao, P-Y. (2012). Data mining techniques and applications – A decade review from 2000 to 2011. *Expert Systems with Applications, 39*(12), 11303–11311.

Linderman, K., Schroeder, R. G., Zaheer, S., Liedtke, C., & Choo, A. S. (2004). Integrating quality management practices with knowledge creation processes. *Journal of Operations Management, 22*(6), 589–607.

Liu, Y., McMahon, C., Ramani, K., & Schaefer, D. (2011). Editorial for the special issue of information mining and retrieval in design. *Advanced Engineering Informatics, 25*(2), 117–118.

Lo, D., Khoo, S.-C., & Wong, L. (2009). Non-redundant sequential rules – Theory and algorithm. *Information Systems, 34*(4–5), 438–453.

Ma, L.-C. (2012). A two-phase case-based distance approach for multiple-group classification problems. *Computers & Industrial Engineering, 63*(1), 89–97.

Mansingh, G., Osei-Bryson, K-M., & Reichgelt, H. (2011). Using ontologies to facilitate post-processing of association rules by domain experts. *Information Sciences, 181*(3), 419–434.

Martínez-de-Pisón, F. J., Sanz, A., Martínez-de-Pisón, E., Jiménez, E., & Conti, D. (2012). Mining association rules from time series to explain failures in a hot-dip galvanizing steel line. *Computers & Industrial Engineering, 63*(1), 22–36.

Mikroyannidis, A., & Theodoulidis, B. (2010). Ontology management and evolution for business intelligence. *International Journal of Information Management, 30*(6), 559–566.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems, 50*(3), 559–569.

Nkambou, R., Fournier-Viger, P., & Mephu Nguifo, E. (2011). Learning task models in ill-defined domain using an hybrid knowledge discovery framework. *Knowledge-Based Systems, 24*(1), 176–185.

Oliveira, S., Ribeiro, J. F. F., & Seok, S. C. (2009). A spectral clustering algorithm for manufacturing cell formation. *Computers & Industrial Engineering, 57*(3), 1008–1014.

Patel, V. L., Arocha, J. F., & Kaufman, D. R. (2001). A primer on aspects of cognition for medical informatics. *Journal of the American Medical Informatics Association, 8*(4), 324–343.

Patel, V. L., Kaufman, D. R., & Arocha, J. F. (2002). Emerging paradigms of cognition in medical decision-making. *Journal of Biomedical Informatics, 35*(1), 52–75.

Saitta, S., Raphael, B., & Smith, I. F. C. (2005). Data mining techniques for improving the reliability of system identification. *Advanced Engineering Informatics, 19*(4), 289–298.

Shiue, Y.-R., Guh, R.-S., & Tseng, T.-Y. (2012). Study on shop floor control system in semiconductor fabrication by self-organizing map-based intelligent multi-controller. *Computers & Industrial Engineering, 62*(4), 1119–1129.

Shu, Y. (2007). Inference of power plant quake-proof information based on interactive data mining approach. *Advanced Engineering Informatics, 21*(3), 257–267.

Strug, B., & Ślusarczyk, G. (2009). Reasoning about designs through frequent patterns mining. *Advanced Engineering Informatics, 23*(4), 361–369.

Tiwari, A., Gupta, R. K., & Agrawal, D. P. (2010). A survey on frequent pattern mining: Current status and challenging issues. *Information Technology Journal, 9*, 1278–1293.

Washio, T. (2007). Applications eligible for data mining. *Advanced Engineering Informatics, 21*(3), 241–242.

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data mining: Practical machine learning tools and techniques* (3 ed.). Elsevier. ISBN 978-0-12-374856-0.

Zaki, M. J. (2000). Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering, 12*(3), 372–390.

Zhu, X., & Davidson, I. (2007). *Knowledge discovery and data mining: Challenges and realities*. New York: Hershey. ISBN 978-1-59904-252-7.