Generating Knowledge in Maintenance from Experience Feedback

Paula Potes Ruiz, Bernard Kamsu-Foguem, Bernard Grabot

Generating knowledge in maintenance from Experience Feedback

Paula Potes Ruiz, Bernard Kamsu Foguem, Bernard Grabot *

Laboratoire Génie de Production/INP-ENIT, Université de Toulouse, 47, Avenue d'Azereix, BP 1629, 65016 Tarbes Cedex, France

A B S T R A C T

Knowledge is nowadays considered as a significant source of performance improvement, but may be difficult to identify, structure, analyse and reuse properly. A possible source of knowledge is in the data and information stored in various modules of industrial information systems, like CMMS (Computerized Maintenance Management Systems) for maintenance. In that context, the main objective of this paper is to propose a framework allowing to manage and generate knowledge from information on past experiences, in order to improve the decisions related to the maintenance activity. In that purpose, we suggest an original Experience Feedback process dedicated to maintenance, allowing to capitalize on past activities by (i) formalizing the domain knowledge and experiences using a visual knowledge representation formalism with logical foundation (Conceptual Graphs); (ii) extracting new knowledge thanks to association rules mining algorithms, using an innovative interactive approach; and (iii) interpreting and evaluating this new knowledge thanks to the reasoning operations of Conceptual Graphs. The suggested method is illustrated on a case study based on real data dealing with the maintenance of overhead cranes.

1. Introduction

An efficient maintenance is one of the key factors of industrial performance [75]. Various maintenance strategies have been developed in the last decades, including Preventive Maintenance [68], Predictive Maintenance [41], Reliability Centred Maintenance (RCM) [64] or Total Productive Maintenance (TPM) [69], and applied to different industrial domains. Nevertheless, the idea of individualized maintenance strategies, based on a specific knowledge in each company, has recently emerged with the emphasis on "knowledge-based enterprise"; indeed, the global economy is progressively shifting from a manufacturing based value system to a knowledge-based value system [19].

Even if knowledge is the base of human activity, only a part of it ("explicit knowledge") is easily accessible, can be stored in information systems and can be efficiently reused. Making explicit the "implicit knowledge" (sometimes called "tacit knowledge") is the objective of Knowledge Engineering [56,77,83], recently object of an increased attention, especially from large companies. Human experts may directly formalize implicit knowledge in an explicit way, but it often requires a long and complex process [61]. Explicit knowledge may also be extracted (automatically or not) from information related to past experiences stored in the information system of the company: learning from experiences has therefore become a very active field [50] and is the main target of this work.

Maintenance is known as a field where a great mass of data is daily collected [10]: this is due to the generalization of Computerized Maintenance Management Systems (CMMS) in large companies, making available a large amount of information provided by technicians after each maintenance intervention (e.g. date of intervention, concerned equipment, technicians name, type of failure, functional localisation, cause, actions performed, etc.). Nowadays, this information is mainly used for traceability purpose, but it could be processed in order to allow the extraction and formalization of hidden knowledge. This knowledge could potentially be useful to improve the maintenance of the production process, but its extraction can hardly be done manually [89]. Data mining techniques [44] may allow an automated or semi-automated extraction of knowledge from databases. Some experiments have already been conducted in the domain of maintenance: see for instance [60,74,90]. These studies show interesting results, but the use of the suggested methods usually requires a high level of expertise. In that context, we suggest an original approach for facilitating the use of an Experience Feedback process dedicated to maintenance, by using Conceptual Graphs [80] for structuring the experiences, expressing the requirements of the users, then analysing and evaluating the provided knowledge database.

In that purpose, we first suggest to formalize the domain vocabulary and the past experiences included in a CMMS using the Conceptual Graphs formalism. We then describe a method for

---

* Corresponding author. Tel.: +33 5 62 44 27 21.
  E-mail addresses: paula.potesruiz@enit.fr (P. Potes Ruiz), bernard.kamsu-foguem@enit.fr (B. Kamsu Foguem), bernard@enit.fr (B. Grabot).

extracting, interpreting and evaluating meaningful knowledge based on association rules mining algorithms as an iterative and interactive process controlled by the human decision maker, who can express his requests on the type of knowledge produced, and validates the results of the knowledge extraction process.

The rest of the article is organized as follows: Section 2 explores the state of the art on Experience Feedback process, industrial maintenance and knowledge-based maintenance systems. Section 3 suggests a model of Experience Feedback in maintenance, while Section 4 presents an illustrative example based on real data dealing with the maintenance of overhead cranes in the aeronautic industry.

2. State of the art

2.1. Experience Feedback and knowledge management

Knowledge management can be defined as the process of creating value from an organizations intangible assets; it combines notions from several different domains, such as organizational behavior, human resource management, artificial intelligence and information technology [52]. Nevertheless, it can be difficult for experts to describe from scratch a non-contextualized generic knowledge [45]. Therefore, techniques allowing to reuse knowledge contained in past experiences have recently been object of an increasing attention, especially Case-Based Reasoning (CBR) [1], which adapts the solution of an already solved problem for addressing a new one [51]. Nevertheless, the aim of CBR is not to generalize this knowledge by formalizing an implicit knowledge in an explicit way, which is one of the objectives of experience management techniques. Indeed, an experience may be considered as a specialization of knowledge, or as a singular instance (or form) of previous knowledge [85]. Knowledge and experience management both include steps of collection, modeling, storage, evaluation and maintenance [11]. Since it is often easier for operational actors to validate the expertise extracted from lived experiences than to directly structure knowledge, the management of past experiences has become a strategic need for enterprises [25]. Close to experience management, Experience Feedback (EF) can be defined as a structured approach for *capitalization*, *processing* and *exploitation* of knowledge derived from the analysis of positive and/or negative events [73]. Such EF process is consistent with the usual processing stages of experience management [11]: discover, capture and collect, store (*capitalization*), evaluate, adapt, transform experience into knowledge (*processing*), reuse and maintain (*exploitation*).

In the context of maintenance, failures and incidents are part of a "negative EF" that aims at avoiding the repetition of similar errors and improving problem solving. We are therefore especially interested in two critical points of the EF process: experience-knowledge formalization and new knowledge discovery, analysed in next sections.

2.1.1. Experience-knowledge formalization

An experience can be defined using different information slots, e.g. context (in which the event occurred), analysis and solution [42]. The EF process requires a deep reflection on the choice of a knowledge representation formalism in the EF database: this formalism should indeed facilitate the difficult step of identifying the knowledge contained in each experience [11], but a formal knowledge representation should also allow to carry out the reasoning, and may facilitate the explanation and the sharing of this knowledge [19]. As a consequence, we have firstly investigated knowledge representation formalisms that could be used to better structure the stored experiences. These formalisms are mainly:

- *Frames based systems* [62], where concepts represent sets of objects with common properties;
- *Semantic Networks* [72], providing a graphical representation of human knowledge, expressed by semantic relations between concepts;
- *Description Logics* (DLs) [12] and *Conceptual Graphs* (CGs) [80], both coming from semantic networks and formalising knowledge using a first-order logic [47].

A representation formalism should provide both expressiveness and a decidable inference process [58]. This is the case for CGs, allowing both representation and reasoning [80]. Moreover, this formalism is nowadays the only logic-based model that has a corresponding interpretation in graph theory [87]. As a consequence, this formalism has been chosen for the present study.

A CG is composed of two types of nodes: concepts nodes (denoted by rectangles) representing entities, and relation nodes (denoted by ovals) representing relationships between these entities. The ordered set of concept types is denoted TC and called a "hierarchy of concept types", while the set of relations is denoted TR and is represented by a "hierarchy of relation types" [16]. These hierarchies represent the ontological knowledge needed to formalize factual knowledge using CGs. Knowledge representation in CGs is entirely graphical and close to an expression in natural language, facilitating the interpretation by the user. Reasoning is based on graph operations, mainly relying on graph homomorphism. These operations act directly on the represented knowledge, without requiring a logical language. This avoids the problem of explaining subsumptions found in DLs [59]. CGs allow to express various types of knowledge: descriptions, patterns, inference rules and constraints [6] and to structure and contextualize knowledge through nested CGs. Moreover, CGs can be translated into other knowledge representation formalisms, such as semantic web languages [81] or Petri nets [82]. Finally, many links exist between CGs and DLs [22,70], which is the most common language in knowledge based applications.

2.1.2. Knowledge discovery

The context of this study requires to transform experiences stored in databases into new knowledge. This is the objective of Knowledge Discovery from Databases (KDD). KDD is "*the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data*" [28]. Since the vast amount of data stored in databases has great potential as source of new knowledge [39], KDD has become essential in many industrial domains, including product and process design, materials planning, quality control, scheduling and maintenance. This process usually requires several steps, including data mining (DM) [44]. Data mining aims at discovering hidden knowledge (relationships or patterns) in large volumes of data [39]. It is generally considered as the main step in the KDD process, and consists in applying data analysis and discovery algorithms for generating knowledge [44]. DM systems are often classified according to the kind of database mined, the kind of knowledge mined, the kind of technique utilized and the application domain [37]. In an industrial context, DM is frequently partitioned according to the kind of knowledge to be mined [38]. One can distinguish between:

- *Descriptive DM* [24,35], that focuses on the discovery of patterns (or models) in the data. Examples include summarization, clustering, association rules mining and sequence discovery.
- *Predictive DM,* focusing on predicting the behavior of a model, and determining the future values of variables from the information included in the databases [21].

In descriptive DM, association rule mining is often considered as facilitating the understanding of knowledge by the user, since relationships with the form IF (antecedent) THEN (consequent) are often seen as close to human reasoning [46].

2.2. Knowledge based industrial maintenance

2.2.1. Industrial maintenance

Maintenance is an important function that supports the primary process of an organization [4]. According to European Standards [27], maintenance is defined as "*the combination of all technical, administrative and managerial actions performed during the life cycle of an item intended to retain it in, or restore it to, a state in which it can perform the required function*". Following this definition, two main types of actions may be distinguished in maintenance: actions for retaining and actions for restoring a service, i.e. preventive and corrective maintenance operations. The classification presented by the European Standard for maintenance terminology [27] is shown in Fig. 1.

This classification depends on the respective position in time of the maintenance activity and of the failure: an intervention before a failure is a preventive maintenance, carried out at predetermined intervals or when a specific need is identified ("on condition"), while an intervention after failure is a corrective maintenance activity, and can be immediate or postponed.

Nowadays, the global performance of the companies depends to a large extent on their performance in maintenance, but the increasing complexity and level of automation of the industrial equipment make it difficult for the users to operate, diagnose and maintain it efficiently. Maintenance tasks are becoming more and more complex and diverse, involving not only activities on mechanical components, but also on electronic, hydraulic, electro-mechanical systems and software [4]. Managers, supervisors and operators consider that a lack of knowledge on the plant, equipment and process is the main limitation for implementing effective maintenance procedures [23]. Thus, it is important to provide decision support to the actors, based on complementary experience and knowledge, for performing the right maintenance action at the right time.

2.2.2. Knowledge-based maintenance systems

It is now commonly considered that the knowledge gained during the maintenance activities could be reused to improve the next interventions [9]. In this context, and considering the difficulties for implementing strategies like TPM and RCM (see Section 1), organizations try to use their internal knowledge more efficiently [40].

Several knowledge-based systems, usually based on expert knowledge, have been developed to support maintenance decisions, with objectives including design of strategies, scheduling of tasks or diagnosis of machines. A method for knowledge capitalization in maintenance, aiming at developing a decision support system (DSS) for the diagnosis and repair of an equipment

by using past experiences, is for instance described in [74]. Detect, preserve, capitalize and actualize the strategic knowledge are its main steps, based on Case-Based Reasoning. The main difficulty of this method is that the used knowledge representation techniques do not have a reasoning mechanism. MAIC [71] is a knowledge-based DSS also using CBR for the maintenance of a chemical plant, based on an adaptation of past experiences, taking into consideration economic variables. EXPERT-MM [8] suggests maintenance strategies on the base of expert knowledge stored in a knowledge database. Chassiakos et al. [14] describes a knowledge-based system aiming at building the maintenance planning of a bridge.

According to [21], only 8% of the studies on data mining application in manufacturing are related to the maintenance domain, including descriptive and predictive DM: this is consistent with [39], denoting that even if this domain was the first area of manufacturing taking advantage of data mining-based solutions, only a few reports on the use of data mining were found in maintenance applications. In Létourneau et al. [55], a DSS for predicting the failure of a component is described, based on data collected from sensors, with the goal to improve the preventive maintenance. Several data mining approaches are used, like decision trees, naive Bayesian networks, regressions and neural networks. In Shen et al. [78], the rough-set theory is applied to the diagnostic of faults and is used to extract rules predicting the occurrence of failures (predictive DM).

In our context, some studies suggest to use association rules mining (descriptive DM) for improving the maintenance process. Others do not explicitly target such improvement, but provide results that may be useful in that purpose (see for instance [18,54,60,63]). The main characteristics of these studies are summarized in Table 1, namely the algorithm used for rule mining, the goal of the study, the application domain and the source of data.

A first comment on Table 1 is that the Apriori algorithm [3], well known for its simplicity and efficiency, is often used. It can also be noticed that the objectives of these studies are often rather precise. This facilitates the interpretation/validation of the obtained rules, all based on the same itemsets (symptoms-causes for a data mining activity oriented on diagnosis for instance). As a consequence, few of these works are interested in getting all the possible types of rules that an algorithm may identify from a database, which is our objective. In this case, helping the user to understand the possible links between extracted rules of different types becomes of great interest. Supporting him not only in diagnostic activities but also in problem solving, database structuration, prognostic activities, etc. is also necessary.

In the studies described in Table 1, the interaction with the users is not detailed, except in Mirabadi and Sharifian [63] where the use of the Generalized Rule Induction algorithm allows the user to specify how many rules she/he expects.

In comparison with these studies, all dealing with association rules mining in maintenance, our approach has several original aspects:

- the goal of the data mining activity is not to solve a specific problem (diagnosis for instance) but to explore all the links between itemsets identified by the rule extraction algorithm;
- since rules of different types will be mined, it may be beneficial to firstly show to the user the relationships between the different types of mined rules. One of our objectives is therefore to provide tools allowing the user to analyse the structure of the produced knowledge;
- for the same reason, it becomes interesting to provide the user with tools allowing to filter the set of rules according to different points of view or expectations; and
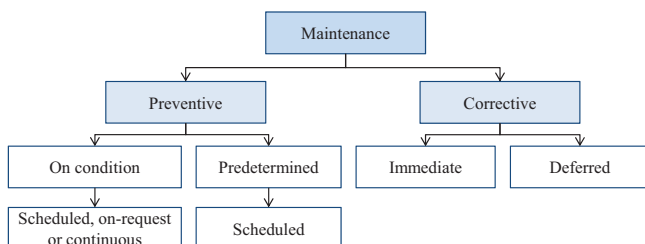

Fig. 1. Maintenance types according to [27].

Table 1
Association rules mining for maintenance.

Ref.	Algorithm/tool	Description	Application case	Data source
Chen et al. [18]	Apriori/	Find correlations between machine and defective products	Semiconductor manufacturing	Follow-up system
Zhang and Yang [92]	Apriori/	Identify man-made mistakes for improving aircraft maintenance	Military aero-transport	Information system
Meseroll et al. [60]	/ThinkAnalytics	Analyse results of built in tests for identifying discrepancies	Aircraft industry	Built-in-tests
Young et al. [90]	Apriori/Clementine	Link failures, diagnoses and repair actions for enhancing maintenance practices	Aircraft industry	Maintenance reports
Mirabadi and Sharifian [63]	GRI (Generalized Rule Induction)/Clementine	Analysing the causes of accidents in a railway network	Railway	Reports on accidents
Baohui et al. [7]	Apriori/	Find association between fault symptoms and corrective actions	Aircraft industry	Maintenance reports
Liu et al. [54]	SQL-Based Apriori Algorithm/	Find links between failures in railway tunnel condition monitoring	Railway	Reports on failures
Maquee et al. [57]	Clustering + Apriori/	Analyse efficiency of maintenance activities by clustering	Bus maintenance	Maintenance reports
Sammouri et al. [76]	T-patterns/	Temporal associations between alarms and occurrence of severe failures	Railway	Floating train diagnosis system
Kamsu-Foguem et al. [43]	FP-Growth/	Improve production processes (including maintenance)	Manufacturing	Follow-up system

- finally, interesting things may be learnt not only from the rules that are present, but also from the fact that expected rules are absent. We shall show that comparing present and absent rules may for instance help to identify problems in the way an experience is structured in the database, or to define more precisely the operating characteristics of the maintenance processes. To our knowledge, there is no work addressing this subject.

Reaching these objectives may be easier if a specific formalism is chosen for modeling the extracted knowledge at a conceptual and operational level: we have chosen Conceptual Graphs in that purpose. Their use will necessitate to define first the domain vocabulary (so-called *support* or ontological knowledge). This representation formalism allows therefore to contextualize the different concepts or items contained in the extracted rules and to use reasoning mechanisms taking into account the specialization of each concept in the support.

3. An experience feedback process for maintenance

3.1. Methodology

A CMMS allows to plan and follow maintenance activities, by supporting information management on the workforce, spare-parts, operation schedules and equipment histories [86]. A CMMS may contain several modules: we are more specifically interested in the equipment management module, allowing to locate the maintenance operations, and in the maintenance management module, providing the history of the past operations.

Formalizing procedural knowledge in the form of association rules can be useful for two reasons [58]: (*i*) the model of the extracted patterns is simple and understandable for a non-specialist user and (*ii*) the workload of the human expert during the process (data mining) remains limited. In order to generate these rules, rough information will firstly be extracted from the CMMS database. It will then be "cleaned" to be exploitable, and finally processed for formalising knowledge and finding rules. As a consequence, we suggest to distinguish two different levels in the EF database: the *experiences database* and the *rules database*.

The phases of the suggested EF process coupled to a knowledge discovery process are summarized in Fig. 2. They are described with more details in Sections 3.2 and 3.3, together with the tools used in each step.

We first select the data of interest from the considered CMMS. A data-cleaning step is then necessary before the processing phase of EF (this first part relates therefore to the *capitalization* phase of the EF process, but it is also the first phase of the knowledge discovery process, which will be explained in more details in Section 3.3). This phase aims at improving data quality, especially because data from the "real world" is usually incomplete (lacking values, lacking attributes of interest, etc.), noisy (containing errors or duplicates) or inconsistent (containing discrepancies). This is done by using techniques like data cleaning, data reduction or discretization [44]. Indeed, a poor quality of the data would result in a poor quality of the mining results [37].

As already introduced earlier, the *processing* phase of the EF process addresses two main challenges:

(i) *"experiences database"* structuration (left path in Fig. 2) for future reuse, using CGs built from a domain vocabulary. An experience being a singular instance of knowledge, this knowledge may remain partially implicit in the experiences database. However, a better formalization of these experiences, here using CGs and their reasoning tools, may be a first support for decision making on the maintenance domain.

(ii) *"rules database"* generation (right path in Fig. 2) from the analysis of past experiences, in order to incorporate more generic knowledge in the industrial maintenance process. We will use in that purpose the well-known "Apriori" algorithm [3] to extract several types of association rules, and the projection operation in CGs [66] to allow the user to evaluate and interpret the extracted rules before validation. The last phase of the EF process, namely the *exploitation* phase, will only be addressed in this article by showing the possible use of the obtained rules for improving the maintenance process.

In what follows, we present in more details the processing phase of the EF process: the experiences database structuration (Section 3.2) and the rules database generation (Section 3.3).

3.2. Experiences database structuration by formalizing experience-based knowledge with Conceptual Graphs

For knowledge representation with CGs, [15] proposes first to define a "support", which provides the domain vocabulary. This
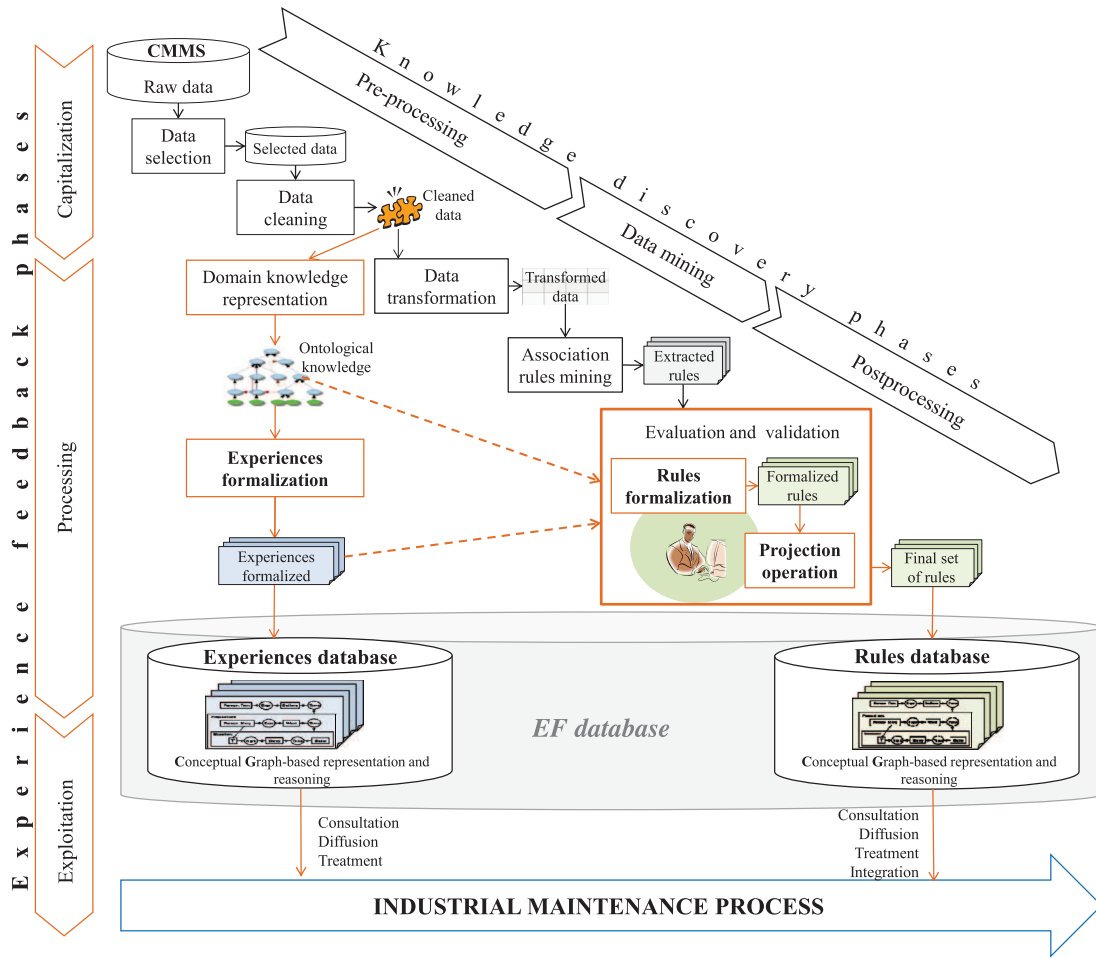
Fig. 2. General scheme of the suggested EF process for maintenance.

support (S), or basic ontological knowledge, can be considered as a rudimentary ontology [16] and is essentially defined as a couple (TC, TR), representing respectively the hierarchy of concept types and the hierarchy of relation types.

In our context, this support corresponds to a high level and generic knowledge on the domain, specifying the vocabulary of the maintenance domain and the semantics of this conceptual vocabulary [31]. Since maintenance is a matter of communication between operators, maintenance actors and experts of various fields, a specific attention has been drawn on ontologies, ensuring that information/knowledge exchanged by different actors is meaningful, and that all the stakeholders interpret it in the same way [88]. An ontology is defined in [36] as "*a formal, explicit specification of a shared conceptualisation*". The notion of "conceptualisation" refers to an abstract model of the concerned domain (in our case, the maintenance domain). The fact that the conceptualisation has to be "shared" implies a consensual knowledge, accepted by the group; "explicit specification" requires that the concepts identified and the constraints that link them are explicitly defined. The "formal" aspect allows to guarantee that the ontology is machine-readable [84]. The components of an ontology should thus allow to formalize the experience-knowledge in a specific domain.

This basic ontological knowledge will be encoded in a simplified general model (see Fig. 3) developed for the EF process in maintenance. TC is described in the left part of Fig. 3, while TR is shown in the right side of the same figure. These hierarchies provide the support (S) and are the basis of a knowledge-oriented representation

of the experiences. It mainly allows to model the equipment and the maintenance interventions according to the three main components of an experience considered in this study: *context*, *analysis* and *solution*. In TC, the "context" part describes the general situation in which the event has occurred (i.e. Work Order (WO), functional localisation of equipment involved, failure, technician); the "analysis" part presents the cause(s) of the problem; finally, the "solution" describes the type of intervention and the actions that have been performed for solving this problem (i.e. selected maintenance activities). TR expresses the basic relations of generic ontologies that will be used here, like "temporal" relation (i.e. before, after, parallel), "spatial" relation (i.e. in, out), "logic" relation (e.g. implies), "usual" relation (i.e. object, agent, involve, etc.) [13], as well as other specific relations of the domain of study, such as "experience relation" (i.e. generates, requires) or "element of". These relations allow to link the different concepts types in the representation of an experience.

At the operational level, CGs will represent the knowledge confined in each experience, but also in each extracted rule (see Section 3.3). They will for instance show which semantic axioms are required to use ontological knowledge in an operational way [30]. Therefore, they allow to address the constraint to integrate knowledge in a way that facilitates sharing and reuse. They will also provide reasoning tools that facilitate the visualization and the verification of the modeled knowledge by end-users [26].

From the support (S) described in Fig. 3, we have built the generic model of an experience (Fig. 4), represented by CGs. The concept nodes are defined by a label and an individual marker, which
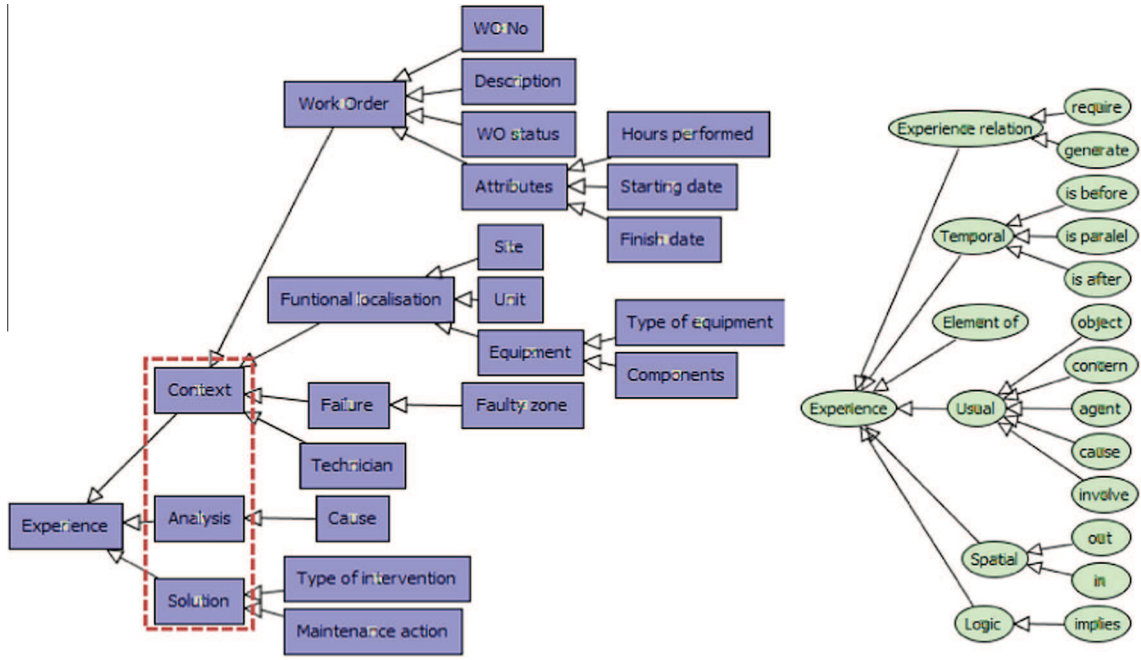
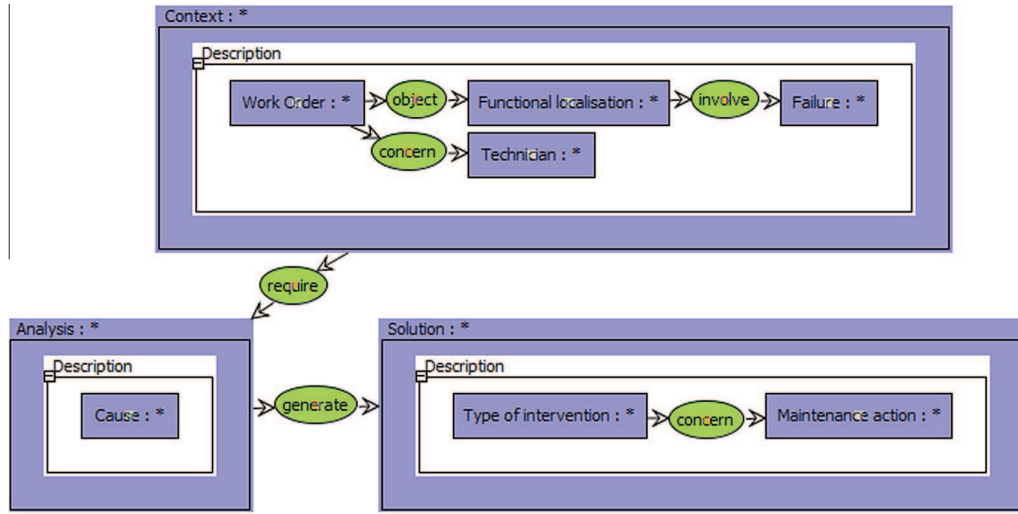**Fig. 3.** Hierarchies of concept and relation types (TC, TR).

Fig. 4. A generic model of an experience.

identify the considered instance (the "∗" denotes a generic marker or undefined instance) [31]. This model should be adapted according to the constraints and restrictions of each application, for instance in order to take into account the type of information provided by the considered CMMS.

The use of CGs for the formalization and evaluation of the extracted knowledge, stored in a *rules database,* is presented in the next section.

3.3. Rules database generation from a knowledge discovery process

Our goal is here to find a way to extract generic knowledge from an analysis of past experiences, then to interpret and evaluate the results obtained with the user, in order to provide a database of validated rules allowing to support decision making.

In [79], the authors detail how the user may be included in the process, especially through three types of discovery process:

automatic, semi-automatic and manual. In our approach, the process of extraction of association rules involves obviously the intervention of the user to guide the process and validate the results. This is thus a semi-automatic process.

In Fig. 5 are presented the main steps of the suggested knowledge discovery process. The *pre-processing phase* aims at preparing the data and organizes them in an adequate format for preparing the *data mining phase,* in which rules will be mined. The last phase, *post-processing phase*, covers the interpretation, evaluation and validation of the results obtained.

3.3.1. Data pre-processing

In order to prepare the data in an adequate format for the mining operation, we build here an appropriate "formal context" for past experiences, based on the ontological knowledge (see Section 3.2). This formal context is defined as a triplet $D = (O, I, R)$, in which D is the database, O is a set of objects or transactions (i.e.
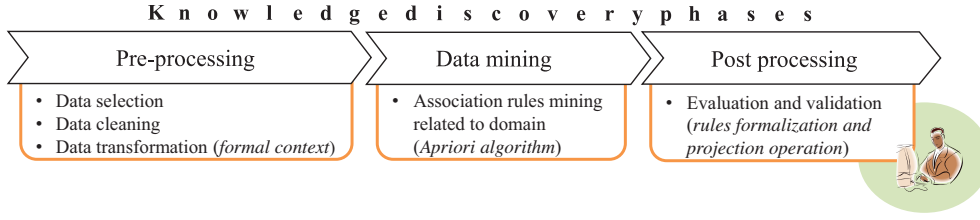
Fig. 5. Main phases of the knowledge discovery process.

each maintenance intervention), I is a set of attributes or items (i.e. concepts defined in the support) and $R \subseteq O \times I$ is a binary relation between O and I. Thus, each maintenance activity or transaction O in D represents a set of concepts or items contained in I.

3.3.2. Data mining

3.3.2.1. Definition of an association rule. An association rule is formally defined as a relation between two itemsets (antecedent and consequent) of a database D. It represents the regularities of a database through relations of the form "IF X THEN Y", denoted as $X \to Y$, where X and Y are proper subsets of I ($X, Y \in I$), and X and Y are mutually exclusive ($X \cap Y = \emptyset$). X is usually called *antecedent*, and Y *consequent*. Let us underline that in spite of this vocabulary, such rule does not denote a "cause-consequence" link between two events X and Y, but only that the occurrence of a given item in a record would imply the occurrence of another item in the same record [2].

The support (sup) of a particular association rule $X \to Y$ is the proportion of transactions in D that contain both X and Y, i.e. the frequency of occurrence of the rule (see Eq. (1)).

$$\text{Support (sup)} = P(X \cap Y)$$
$$= \frac{\text{number of transactions containing both } X \text{ and } Y}{\text{total number of transactions}}$$
$$(1)$$

The confidence (conf) of the association rule $X \to Y$ is a measure of the accuracy of the rule, determined by the percentage of transactions in D containing X that also contains Y [48]. Confidence may be considered as a measure of the "strength" of a rule, or conditional probability $P(Y|X)$ (see Eq. (2)).

$$\text{Confidence (conf)} = P(Y|X) = \frac{P(X \cap Y)}{P(X)}$$
$$= \frac{\text{number of transactions containing both } X \text{ and } Y}{\text{number of transactions containing } X}$$
$$(2)$$

The number of rules, and therefore their genericity, will be managed by defining predefined thresholds *minsup* and *minconf* of (resp.) the support and confidence [33]. Choosing these parameters is not an easy task: it has been often underlined that the user needs some expertise in order to find the *minsup* and *minconf* that will lead to the best rules [32]. An algorithm has recently been proposed to mine the "top-k" association rules; this becomes for example useful when the user wants to control the number of extracted rules [29].

3.3.2.2. Mining of association rules. The goal is here to derive association rules [2] from past experiences, these rules linking the concepts of the modeled domain. This can be done through rule mining, extracting relevant relationships, correlations, frequent patterns or associations among sets of items in database.

Two steps are necessary for mining association rules [3]: (*i*) discover the itemsets satisfying the user-specified minimum support from a given dataset (finding "frequent itemsets") and (*ii*) generate robust rules satisfying the confidence required by the user from all frequent itemsets found (generating association rules). Many algorithms exist for discovering association rules, the best known being the Apriori algorithm [3]. Unlike the other algorithms, Apriori finds all association rules between frequent itemsets by adding to large sets, and pruning small sets [20]. This is consistent with our objective to extract all possible types of rules for later evaluation of the results by the user.

For the general problem of mining the association rules, m items potentially lead to 2^m frequent itemsets. To address this problem, the Apriori algorithm uses an estimation procedure in order to determine the itemsets that should be measured at each iteration. Thus, an itemset X of length k is frequent if and only if every subset of X, having length $k - 1$, is also frequent; i.e., if an itemset of size k is a frequent itemset, then all the itemsets below $(k - 1)$ size must also be frequent itemsets. This consideration permits a significant reduction of the search space, and allows rule discovery in a computationally reasonable time [67].

Let k-itemset be an itemset having k items, L_k be a set of large k-itemsets and C_k a set of candidate k-itemsets. The Apriori algorithm [3] is summarized in Fig. 6.

As described in [3], the first iteration of the algorithm counts item occurrences to determine the large 1-itemsets. During the following iterations, the large itemsets L_{k-1} found in the $(k - 1)$th iteration are used to generate the candidate itemsets C_k, using the Apriori-gen function (see Fig. 6), which includes two phases taking as argument L_{k-1}: union and pruning. In the union phase, all k-itemsets candidates are generated. Then, in the pruning phase, all candidates generated in the union phase with some non-frequent $(k - 1)$-itemset are removed [49].

Improvements of the Apriori algorithm have been suggested in order to decrease the execution time, the memory consumption and to improve its efficiency [17,91]. An analysis of different

Input: Database D

```
1)  L₁ = {large 1-itemsets};
2)  for (k = 2; L_{k-1} ≠ ∅; k++) do begin
3)          C_k = apriori-gen (L_{k-1}); // New candidates
4)              forall transactions t ∈ D do begin
5)                      C_t = subset (C_k, t); // Candidates contained in t
6)                          for all candidates c ∈ C_t do
7)                                  c.count++;
8)              end
9)              L_k = {c ∈ C_k | c.count ≥ minsup}
10) end
11) Answer = ∪_k L_k;

apriori-gen (L_{k-1})

12) forall itemsets c ∈ C_k do begin
13)             forall (k-1)-subsets s of c do begin
14)                     if (s ∉ L_{k-1}) then
15)                             delete c from C_k
16)             endfor
17) endfor
```

Fig. 6. The Apriori algorithm.

propositions developed in the literature for improving this association rule mining process is for instance presented in [58]. Nevertheless, because of its popularity and good performance, we have chosen the classical Apriori algorithm for extracting association rules in our application example. A post data-mining phase, which has in our opinion a specific interest and should not be automated, is so required to evaluate and filter the extracted rules.

3.3.3. Assessment and interpretation of the rules

The post-processing phase is the assessment of the utility and reliability of the mined rules, then the interpretation of the discovered knowledge [35]. The main objective of this phase is to help the user to discover useful knowledge in the set of extracted association rules. There are several ways to evaluate the association rules in a knowledge discovery process. An interesting classification has been presented in [34], suggesting: (*i*) an "objective evaluation" (based on the thresholds defined by the user, so already taken into account in Apriori), (*ii*) a "semantic evaluation" (based on the domain knowledge), and (*iii*) a "subjective evaluation" (based on the objectives and beliefs of the user). The suggested post-mining approach is based on this classification.

3.3.3.1. Objective evaluation. The *minsup* and *minconf* thresholds chosen by the user allow a first evaluation of the extracted rules. Choosing the optimal levels of *minsup* and *minconf* is a difficult task:

If the *minsup* is high, only experiences often combining the same itemsets will be considered. As a limit, if *minsup* = 1, a single rule would be generated under condition that all the experiences contain the same itemset. If *minsup* = 0, each experience would be expressed by a different rule: no generalization is in that case performed. Therefore, the *minsup* directly controls the number of rules extracted, and as a consequence the degree of generalization of the knowledge contained in these rules. In practice, efficiently mining the experiences requires to test different thresholds since rare rules may be more interesting than frequent ones. In the case of maintenance for instance, the "urgency" of a corrective maintenance activity may be considered, but very urgent maintenance orders are usually rare. Defining a high *minsup* would prevent from generating knowledge concerning these rare, but very critical experiences.

The *minconf* has a different interest: it shows the validity of a rule, i.e. up to what point the consequent part is linked to the antecedent part. A high *minconf* allows the generation of very robust rules, but in practice, these rules are usually well known. On the opposite, rules with a low confidence may be inconsistent (and may denote typing errors for instance), but may also express unusual but interesting situations.

3.3.3.2. Semantic evaluation. Semantic evaluation allows to validate the interest of a rule according to external knowledge on the domain. Since the knowledge on a large domain like maintenance has multiple facets and may be complex, it is difficult to give generic guidelines here, except on an original point: combining objective and semantic evaluation may allow to diagnose some aspects of the consistency and coherence of the database.

In that purpose, we propose to use the following step-by-step approach (illustrated in the case study) as a methodology to understand and interpret the extracted rules:

- analyse "elementary" rules (i.e. involving two items),
- express each attribute by a question,
- express the problem addressed by the rule by combining the questions involved,
- interpret the support and confidence of each rule,

- analyse the possible use of the rule for improving the maintenance activities,
- check whether the reverse rule is, and should be, present,
- consider the more complex rules by comparison with the elementary ones by mean of three logical operations, denoted here as extension (of the antecedent or consequent part of the rules), permutation (of items between the antecedent and consequent parts of the rules), and junction (of the antecedent or consequent parts of the rules),[1] and then using the same steps above,
- define, using CGs, the structure of the rules database (relations between identified elementary and complex rules) that allow a visual exploration of the mined rule set, and
- finally, formalise a "metarule" to generalize a rule-set and provide a new abstraction level grouping the rules of the rules database.

Let us consider an example for showing the interest of assessing present and absent rules, using two attributes of an experience: "cause" of a failure and "maintenance action" performed. In many real cases, a given cause of failure should result in a given maintenance action; conversely, this given maintenance action is only performed if this cause occurred. A consequence is that, in such case, if a rule of the type "IF (cause of failure A) THEN (maintenance action B)" is generated, the inverse rule should also be generated, since it has the same support and confidence. In some practical cases, we have noticed the absence of such reverse rule (see Section 4). This may show a difference of granularity in the way the attributes are considered by the designer of the database (maintenance manager) and the maintenance actor. The cause of failure could for instance have sub-causes for the maintenance actor, leading to different maintenance actions.

As a consequence, the analysis of the rules (not only the present ones but also the absent ones), considering their *minsup* and *minconf,* may allow to identify some inconsistencies in the databases (data typing errors, entry mistakes or defaults in the definition of the attributes).

3.3.3.3. Subjective evaluation: checking the consistence of the rules with the user's expectations. Even if algorithms like Apriori may mine various types of rules, a user in a given situation has usually an idea on the type of rule that she/he expects. For instance, the user is usually interested in rules linking "symptoms" to "causes" in a diagnosis phase, or "causes" to "maintenance actions" in the phase of search for solution.

Let X be the set of discovered association rules. According to a user query Q regarding the structure of a rule, [53] suggests to distinguish between four sets of rules:

- *Conforming rules* a discovered rule $X_i \in X$ is conform to the user query Q if both antecedent and consequent parts of X_i can be derived from Q, i.e. the conforming rules are the derived rules having both antecedent and consequent parts consistent with the user's expectation.
- *Unexpected consequence rules* a discovered rule $X_i \in X$ has an unexpected consequence with respect to Q if the antecedent of X_i is a projection of Q, but not the consequent part. Unexpected consequence rules show discovered rules that may be inconsistent with the existing knowledge. We can also find in this category the rules including the user query Q, but the consequent part of these rules provides more information.

---

[1] These operations are semantically well defined by the graph operations of specialization, generalization and equivalence: the "extension" is a form of "generalization" in CGs, the "junction" a form of "join" in GCs and "permutation" is a light form of equivalence in CGs.

- *Unexpected antecedent rules* a discovered rule $X_i \in X$ has an unexpected antecedent with respect to Q if the consequent of X_i is a projection of Q, but not the antecedent part. Unexpected antecedent rules can show other antecedents that can lead to the same result. We can also find in this category other rules including the user query Q, in which the antecedent part provides more information, leading to the same consequent part.
- *Both-side unexpected rules* a discovered rule $X_i \in X$ is both-side unexpected with respect to Q if both the antecedent and consequent part of the rule X_i are not the projections of Q. Both-side unexpected rules may be relevant but are not a priori known by the user or are not mentioned in its expectations.

The visual representation of the mined association rules by CGs makes remarkably easy the process of modeling a query, then finding the rules consistent with this query: an example of an user's expectation is presented in Fig. 7, where the user searches among the extracted rules those of the form: Context → Analysis. This rule evaluation is subjective, since it depends on the user through the type of rule that she/he wants to find.

Finding the rules that meet an expectation is easy using the "projection" operation of graphs defined by the CGs formalism. The fact that the same language (CGs) can be used at the interface (query) and operational (processing) levels makes transparent the logical structure of information, facilitating the understanding and interpretation of the results by the user [66].

Given two graphs G and H, $H \leqslant G$ (G is said to "subsume" H) if H can be obtained from G by a global operation (projection), which is essentially a homomorphism of graph. Thus, $H \leqslant G$ implies the existence of a projection from G to H [66]. More specifically, the relations of specialization/generalization are fundamental notions for reasoning with graphs. H is a specialization of G ($H \leqslant G$) if H can be derived from G by specialization operations, or G is a generalization of H ($G \geqslant H$) if G can be derived from H by generalization operations [16].

Thus, the subsumption relation defines which items are classified by which concepts i.e., whether the description of a given concept is more general than the description of another concept. The semantically significant implication of the subsumption relation is the inheritance of properties from the parent (subsuming) concept to the child (subsumed) concept. The projection operation is the key computational notion for reasoning on CGs, since it corresponds to the logical subsumption when considering the logical formulas associated with CGs.

To perform the projection, we first represent the extracted rules based on the context, analysis and solution of the generic model of an experience (Fig. 4) using CGs, in order to facilitate their interpretation. We also represent the user's expectations using CGs. The query/answering mechanism will search projections between the "query" graph (user's expectation) and "response" graphs
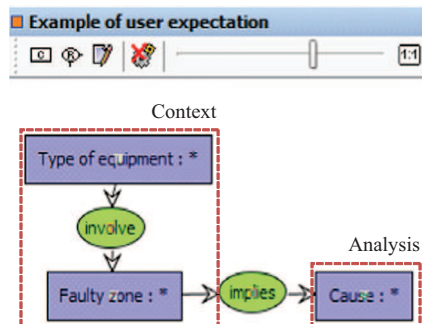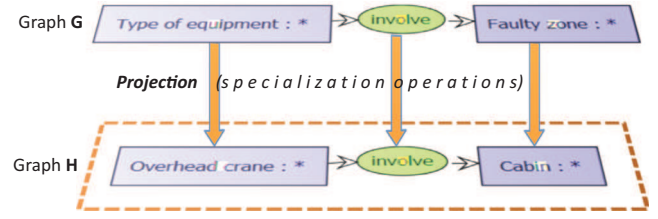

Fig. 8. Example of the projection operation in CGs.

(extracted rules), to select a final set of rules relevant for the user. The existence of a projection from a Conceptual Graph G on a Conceptual Graph H means that the knowledge represented by H is deducible from the knowledge represented by G (called query graph). In Fig. 8, we illustrate a projection operation: the concept "Overhead crane" of the graph H is a specialization of the concept "Type of equipment" of the graph G, and the concept "Cabin" is a specialization of the concept "Faulty zone" according a support previously defined.

It is then easy if needed to classify the rules according to the categories defined in [53]: conforming rules, unexpected consequence rules, unexpected antecedent rules, both sides unexpected rules (see Section 4).

From an algorithmic point of view, the computational problem of determining whether a given graph can be projected in another graph has a nondeterministic polynomial time [65]. Some polynomial cases are obtained by restricting the form of the graphs in practical applications, especially with a polynomial that depends on the choice of query graph [6]. For instance, the projection of an acyclic CG into another graph is polynomial [16].

We shall see in next section how this generic approach may be instantiated on a real case.

4. Application example

We consider here a real set of reports on maintenance operations performed on overhead cranes, used to assemble different sections of aircrafts in a large company of the aeronautical sector.

After extraction from the SAP ERP Production Maintenance module and a cleaning phase, the starting point is an Excel© sheet with 693 maintenance reports. The main fields (attributes) of this sheet are shown in Fig. 9: work order number, type of equipment (here hidden), faulty zone, cause, type of intervention and maintenance action performed (the name of the technician is mentioned but is also hidden).

After cleaning, we formalize the past experiences and look for association rules using the knowledge discovery process presented in Section 3 (see Fig. 2). Therefore, we begin by building a support of the domain vocabulary in CGs.

4.1. Definition of the support: ontological knowledge

Several platforms and implementation tools for CGs have been proposed [5], allowing to define a support and to build the corresponding graphs. We have chosen the CoGui platform for this implementation: the CoGui editor[2] is a free graph-based visual tool, developed in Java, allowing to build intuitive visual structures with reasoning capabilities. Essentially, this tool allows to build a support (TC, TR) and a set of CGs representing assertions, usually called "facts", but in our context denoted as "experiences" (stored in the *experience database*), and "rules" (stored in the *rules database*).

Fig. 7. Example of user's expectation expressed by a CG.

---

[2] http://www2.lirmm.fr/cogui/.

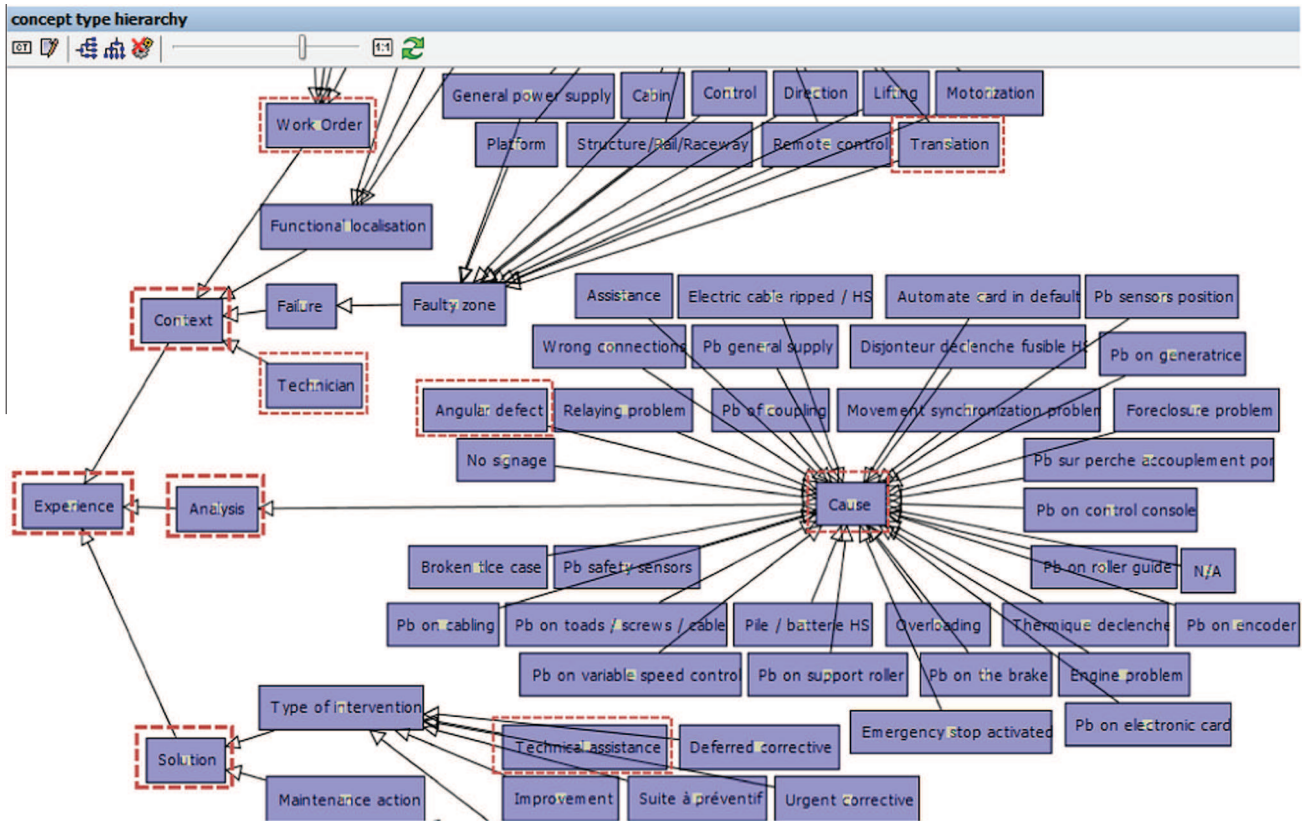**Fig. 9.** Maintenance work reports on overhead cranes.

Fig. 10. A part of the hierarchy of concept types (TC).

In Fig. 10 is shown a part of the basic support (hierarchy of concept types) used for the application example. In this example, we consider the same hierarchy of relation types than in the right side of Fig. 3. Among the main concept types, the "experience" concept is described by a "context", an "analysis" and a "solution". We consider here that the "context" includes the "work order (WO)", the "functional localisation", the "failure", the "technician", etc. The "analysis" contains the different "causes" of intervention while the "solution" describes the "type of intervention" and the "maintenance action".

4.2. Formalizing experience-knowledge with Conceptual Graphs

In Fig. 11, an event on a bridge crane is the basis of an experience. The CG for an experience is built according to the support defined in Fig. 10, in which the used concepts are denoted by doted lines. The graph of Fig. 11 can be interpreted as follows: in the considered experience, Context C1 requires Analysis A1, which generates Solution S1. More specifically, the context is described by the Work Order No 698188 for equipment POMC02002. We distinguish here the object of the maintenance action, the failure (on
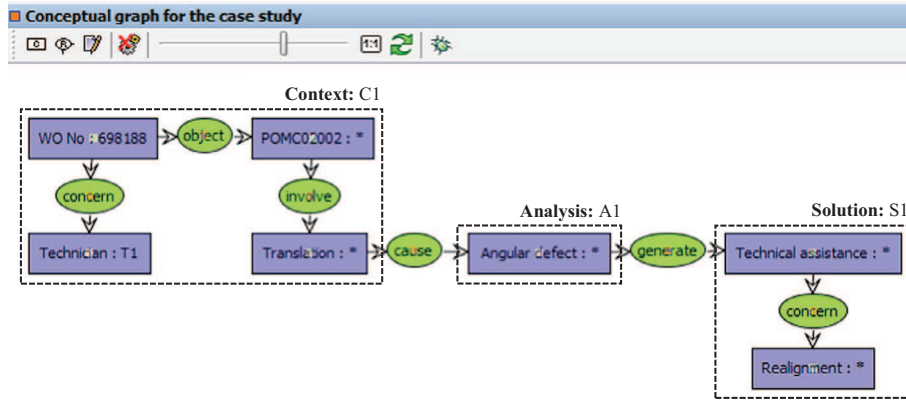
Fig. 11. Conceptual Graph for an experience of the case study.

Table 2
Results obtained when support and confidence vary.

Minsup (%)	10	10	20	20	30	30
Minconf (%)	100	50	100	50	100	50
Frequent itemsets	51	51	21	21	9	9
Extracted rules	4	80	3	44	0	12

the translational movement) and additional data used to locate the equipment. In the analysis step, we seek for the primary cause of intervention (in this case, an angular defect of the equipment). Finally, the description of the solution concerns the type of intervention carried out and the actions performed (in this case, a technical assistance consisting in a realignment of the instrument).

4.3. Discovering association rules

The objective in this step is to discover association rules on the past experiences. As an example, we look for rules linking type of equipment, faulty zone, cause of intervention, type of intervention and maintenance action, which are therefore the five main concepts of the support used in the data mining process.

During a data pre-processing phase, we have built a "formal context" organizing the data in an appropriate way, O being a set of experiences (O = {*Experience 1, Experience 2, Experience 3, ...*}), I a set of main concepts of support (I = {*type of equipment, faulty zone, cause of intervention, ...*}) and R the binary relations between facts.

We have chosen the SPMF[3] software (Sequential Pattern Mining Framework), which is an open-source Data Mining software written in java, for association rules mining from a formal context. The Apriori algorithm (Fig. 6) [3], allowing to mine association rules from a *minsup*, a *minconf* and a *formal context*, is included in the package. In Table 2, we present some results obtained when the support and the confidence defined by the user vary. As expected, it can be seen that the combination of a low *minsup* and a high *minconf* leads to many frequent itemsets, but few rules. A high *minconf* leads to fewer but more robust rules, i.e. rules with a high conditional probability, antecedent and consequent being almost always linked.

For finding a good compromise between number of rules and robustness, we have empirically chosen *minsup* = 20% and *minconf* = 90%, leading to the extraction of 21 frequent itemsets and 16 rules (Table 3). The choice of these parameters depends on the characteristics of the database. In our case, a low support

was required for finding a substantial amount of rules (the number of obtained rules decreases considerably when *minsup* increases).

4.4. Interpretation of the extracted rules

Understanding the rules listed in Table 3 is possible, but requires an effort and some methodology. In that purpose, we suggest to use the step-by-step approach proposed in the semantic evaluation section (Section 3.3.3).

The first four rules of Table 3 are the "elementary" ones established by the algorithm (they only link two items). They involve four attributes denoting answers to the following questions:

Attribute	Question
type of intervention:	in which conditions?
maintenance action:	what?
cause:	why?
faulty zone:	where?

Let us now analyse each rule.

R1: ***Maintenance action*** = Reset − acknowledgement → ***Type of intervention*** = Urgent corrective
Question answered: link between "what" and "in which conditions".
Interpretation: these two items are often present in the database (support = 0.29) and the "reset-acknowledgment" action is often considered as an "urgent corrective" operation (confidence 91%).
Usefulness: this rule allows to know that a given action (reset-acknowledgment) is often performed urgently in this context. It would therefore be of interest to check that the required materials and instructions are immediately available for the maintenance actors, and that the actors are perfectly trained for this operation.

R2: ***Maintenance action*** = Realignment → ***Cause*** = Angular defect
Question answered: link between "what" and "why".
Interpretation: this rule should allow to link a given maintenance action and its cause. Realignment and angular defect are often linked (support = 33%).
Usefulness: this rule could allow to improve the diagnosis but the absence of the reverse rule denotes that an "angular defect" may be addressed by other maintenance actions (the confidence threshold has not been reached by the reverse rule). This means that the definition of an experience adopted in this company (denoted by an ontological knowledge) does not allow to

Rule ID	Rule	Sup	Conf
R1	*Maintenance action* = Reset − acknowledgement → *Type of intervention* = Urgent corrective	0.29	0.91
R2	*Maintenance action* = Realignment → *Cause* = Angular defect	0.33	0.97
R3	Maintenance action = Realignment → Faulty zone = Translation	0.34	0.99
R4	*Cause* = Angular defect → *Faulty zone* = Translation	0.42	0.99
R5	*Type of intervention* = Technical assistance, *Maintenance action* = Realignment → *Cause* = Angular defect	0.20	0.99
R6	*Type of intervention* = Technical assistance, *Cause* = Angular defect → *Maintenance action* = Realignment	0.20	1
R7	Type of intervention = Technical assistance, Maintenance action = Realignment → Faulty zone = Translation	0.20	1
R8	*Type of intervention* = Technical assistance, *Cause* = Angular defect → *Faulty zone* = Translation	0.22	0.97
R9	*Maintenance action* = Realignment, *Faulty zone* = Translation → *Cause* = Angular defect	0.33	1
R10	*Maintenance action* = Realignment, *Cause* = Angular defect → *Faulty zone* = Translation	0.33	0.97
R11	*Maintenance action* = Realignment → *Faulty zone* = Translation, *Cause* = Angular defect	0.33	0.99
R12	*Type of intervention* = Technical assistance, *Maintenance action* = Realignment, *Faulty zone* = Translation → *Cause* = Angular defect	0.20	1
R13	*Type of intervention* = Technical assistance, *Maintenance action* = Realignment, *Cause* = Angular defect → *Faulty zone* = Translation	0.20	0.90
R14	*Type of intervention* = Technical assistance, *Cause* = Angular defect, *Faulty zone* = Translation → *Maintenance action* = Realignment	0.20	0.90
R15	*Type of intervention* = Technical assistance, *Maintenance action* = Realignment → *Faulty zone* = Translation, *Cause* = Angular defect	0.20	0.91
R16	*Type of intervention* = Technical assistance, *Cause* = Angular defect → *Faulty zone* = Translation, *Maintenance action* = Realignment	0.20	0.93

define a clear link between cause and corrective action, causes being probably not enough precisely described. This statement should lead to improve the definition of the experiences.

R3: ***Maintenance action*** = Realignment → ***Faulty zone*** = Translation
Question answered: link between "what" and "where".
Interpretation: the maintenance action "realignment" performed in the "translation" sub-system of the overhead crane is often present in the database (support = 34%) and this action is almost only performed in this zone (confidence = 99%). It would be interesting to investigate the very rare cases in which this maintenance action has been performed in another zone (such exceptional event may denote a typing error).
Usefulness: since this maintenance action often occurs in this zone, its practical feasibility should be checked (access of the actors to the zone with the material required by the activity (intervention) could be checked/improved, etc.).

R4: ***Cause*** = Angular defect → ***Faulty zone*** = Translation
Question answered: link between "why" and "where".
Interpretation: "angular defect" is often the cause of interventions in the zone "translation" (confidence = 99%), but the reverse rule is not present: we might therefore deduce that many other problems may happen in that zone.
Usefulness: if specific materials or spare parts are required to fix this problem, they could be positioned close to this zone. If the faulty zone is critical, actions should be performed to avoid the occurrence of this cause.

The following rules may be considered as variants of the three basic ones R2–R3–R4 by mean of the three logical operations defined in Section 3.3.3, denoted here as extension, permutation and junction:

- **extension** of the antecedent part of the rules: new items are added to the antecedent part of the rule, meaning that the rule becomes more specific. For instance, R5 is an extension of R2: a "type of intervention" (technical assistance) has been added in the antecedent part. In this case, the support of the extended rule is lower (meaning that other types of intervention than "technical assistance" are associated with "realignment" in the past experiences) but the confidence in the rule remains high. This new rule does not completely overlaps R2, and we consider that both can be kept, even if R5 does not add much to R2. Similarly, R7 is an extension of R3 (same addition of item) and R10 is also an extension of R3 (addition of a cause "angular

defect"). R8 is an extension of R4 (addition of the type of intervention "technical assistance"). R13 is an extension of R8 and R14 an extension of R6.

The case of R9 is different: it is an extension of R2 (addition of a faulty zone "translation") but the support is the same, meaning that the added item is present in all the experiences that have allowed to create R2, the combination of "realignment" and "translation" allowing to conclude that the cause is an "angular defect" (confidence in R9 = 1). Such univocal link between maintenance action and cause is not possible on other faulty zones, since the confidence in R2 is only 0.97.

R12 is similarly an extension of R5, having the same support and a higher confidence, it can replace it because it is a more complete rule.

- **permutation**: the Apriori algorithm also tests permutations of items between the antecedent and consequent parts of the rules (the support is identical but the resulting confidence may be different). R6 is a permutation of R5, the cause "angular defect", which was in the consequent part of R5, and the maintenance action "realignment", which was in the antecedent part, have been permuted. The result is a full confidence in R6 (confidence = 1), meaning that the type of intervention "technical assistance" and cause "angular defect" allow to conclude without ambiguity on the maintenance action performed "realignment", whereas the type of intervention and maintenance action may (seldom) be linked to other causes (R5).
- **junction** of the consequent parts: several rules combine the consequent parts of other rules that have the same antecedent part. It is the case of R11, combining the consequences of R2 and R3, of R15 (R5 and R7) and of R16 (R6 and R8).

R11 links a maintenance action "realignment" and a couple (faulty zone, cause), namely "translation" and "angular defect". The good support and confidence of the combined rule, close to the ones of the two "elementary" rules, shows that there is probably a high correlation between the two items present on the consequent part of R11. This is confirmed by R4 showing an association between these two items.

This is quite similar for R15, but its confidence is much lower than its two "elementary" rules (R5 and R7). We can also notice that R15 is an extension of R11, R5 being an extension of R2 and R7 of R3. Nevertheless, R11 is much better than R15 (respective supports: 0.33 and 0.2; respective confidence: 0.99 and 0.91);

other types of intervention than "technical assistance" are possible.

R16 is an extension of R6 and R8, with a comparable support but lower confidence, even if the association between the two items of its consequent part (faulty zone "translation" and maintenance action "realignment") is also denoted in the elementary rule R3.

Up to that point, it is clear that understanding the rule database extracted by a data mining algorithm requires to "map" the links between the identified rules. The links between rules of the case study are summarized in Fig. 12. As shown above, such analysis is important since new knowledge may be created by correlating the interpretations of the rules that are close.

Finally, CGs can again be used for formalising a "metarule" generalizing and grouping all the rules of the rule database (see Fig. 13) (except rule R1, remaining in some sense "unique").

4.5. Looking for specific types of rules

We have shown in Section 3.3.3 that CGs can be used to represent *facts* that match the extracted rules and *queries* that correspond to the user's expectations.

A limitation of the CoGui editor is that only simple Conceptual Graphs can be used (CGs without negation and without nesting graphs) to perform the projection operation, because the software does not allow an efficient projection with nested graphs. Thus, to formalize the rules extracted in the case study, we build a simple CG as a "fact" in CoGui that connects the antecedent and consequent parts by the relation "implies", always taking into account the generic model of an experience presented in Fig. 4 (i.e. context, analysis, solution). In the rule representation, let us consider the context part composed of the type of equipment and the faulty zone, the analysis part composed of the principal cause of intervention, and the solution part composed by the type of intervention and the maintenance action. We find in the following figures these main concepts (i.e. "type of equipment", "failure zone", "cause", etc.) or the more specific concepts or specializations of these concepts in the support (i.e. "technical assistance", "realignment", etc.) according to the hierarchy of concepts types (see Fig. 10). In Fig. 14, we show the representation of R5 using a simple CG.
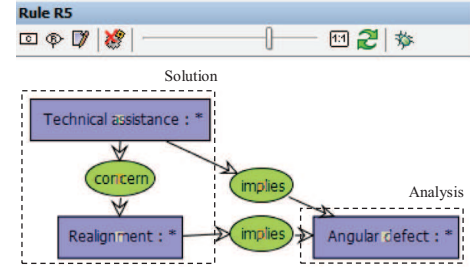

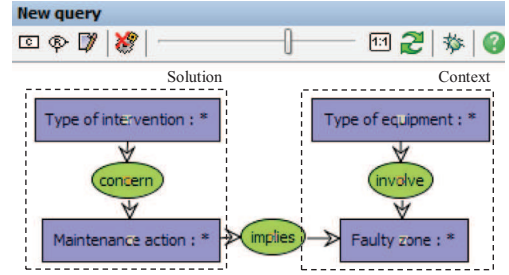**Fig. 14.** Model of the extracted rule R5.

Fig. 15. User's expectation (query Q).

Let us consider a temporary database composed of the 16 extracted rules. A query (Q) is expressed through a CG in order to model the user's expectation. Therefore, elements answering Q (i.e. projections) are defined as elements in the database that are specializations of Q, or in other words, elements that are subsumed by Q [16]. We show in Fig. 15 a user's expectation ("new query") that corresponds to query Q: it is the "model" of rule that the user expects. Thus, the software will search in the extracted rules those of the form [Solution] → (implies) → [Context].

The CoGui interface puts in black color the projection found. The results of these projections have been classified in conforming rules, unexpected consequence rules, unexpected antecedent rules and both-side unexpected rules and are presented below.
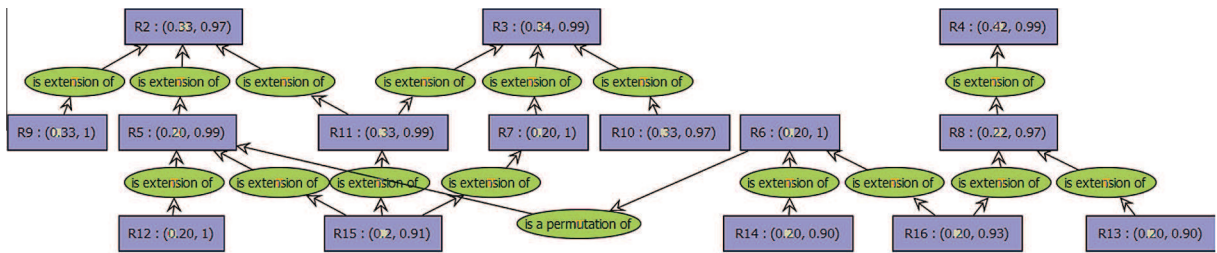

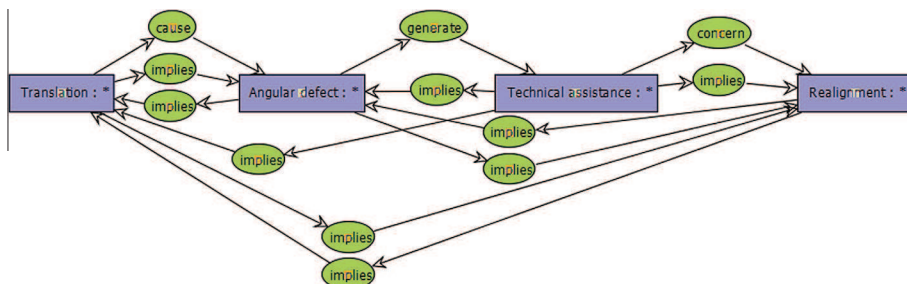**Fig. 12.** Links between extracted rules (sup,conf).

Fig. 13. Metarule grouping R2 to R16.

4.5.1. Conforming rules

R3 (left side of Fig. 16) and R7 (right part of Fig. 16) are the only rules conform to the user query given that the antecedent and consequent parts of R3 and R7 can be derived from query Q. In other terms, the concepts of R3 and R7 are specializations of concepts of Q according to the support (i.e. $R3 \leqslant Q$ and $R7 \leqslant Q$). A small difference exists between these two rules: R3 does not specify the type of intervention as in rule R7, where the type of intervention corresponds to a technical assistance ("technical assistance" is a specialization of concept "type of intervention" (see hierarchy of concept types in Fig. 10)).

4.5.2. Unexpected consequence rules

R11 (left part of Fig. 17) and R15 (right part of Fig. 17) are rules including the user query, but their consequent part provides more information; the type of intervention (technical assistance) and the

maintenance action (realignment) give also the cause of intervention (i.e. angular defect), which is not specified in the query. Compared to R11, R15 specifies the type of intervention (technical assistance). This is the same for R2 and R5, which also have unexpected consequence according to the user query.

4.5.3. Unexpected antecedent rules

Rules R10 (left part of Fig. 18), R13 (middle part of Fig. 18) and R8 (right part of Fig. 18) include other antecedents than the query (i.e. the cause of intervention), leading to the same consequent part. The difference between these three rules is in the solution part, i.e. the type of intervention and associated maintenance action. R10 specify the maintenance action (realignment), R13 the type of intervention (technical assistance) and the maintenance action (realignment), and R8 only specify the type of intervention (technical assistance). The support and confidence of these rules
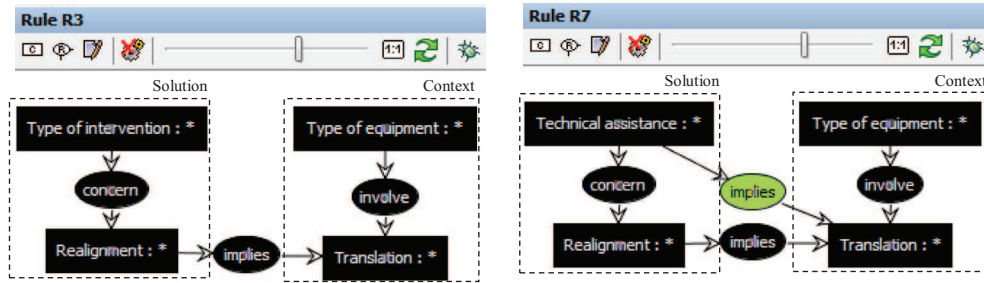

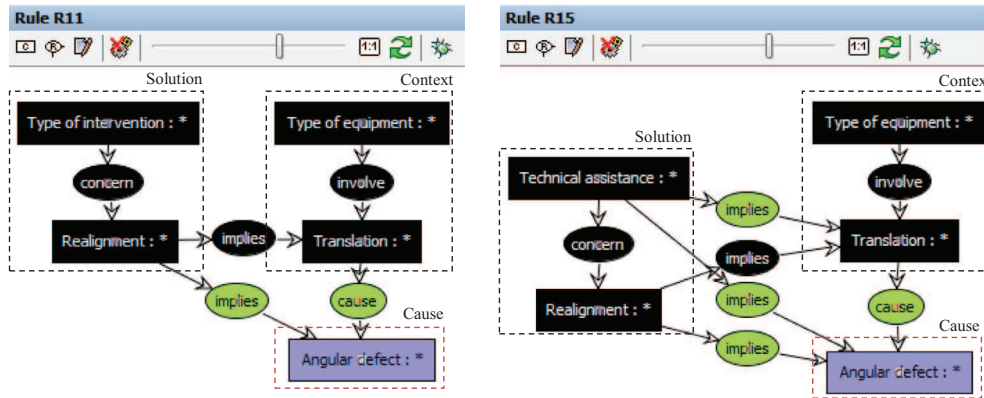**Fig. 16.** Conforming rules with the user query: R3 and R7.

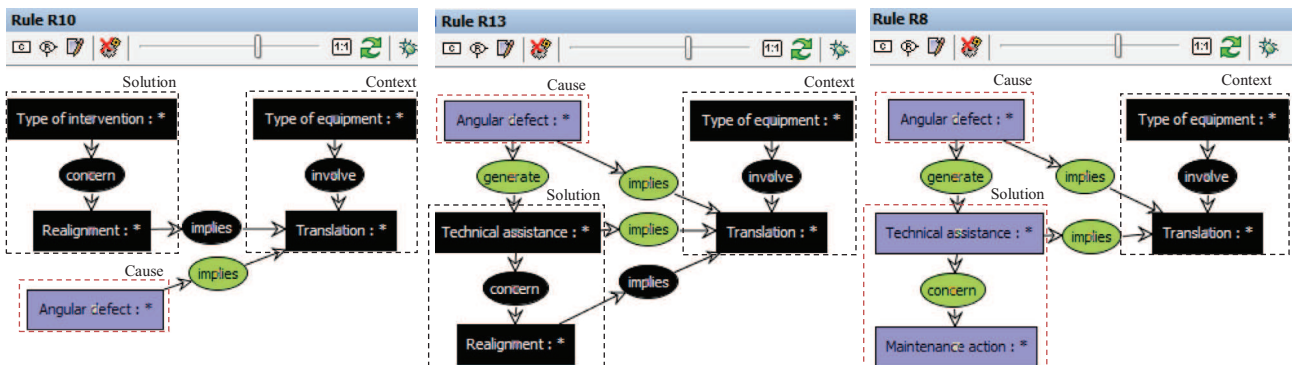**Fig. 17.** Unexpected consequence rules: R11 and R15.

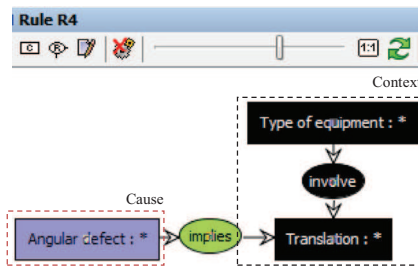**Fig. 18.** Unexpected antecedent rules: R10, R13 and R8.

Fig. 19. Unexpected antecedent rule: R4.

are quite different (see Table 3). On the other hand, R4 (Fig. 19) has an unexpected antecedent according to the user query, i.e. only the consequent part of R4 is a projection of Q.

4.5.4. Both-side unexpected rules

The rest of the extracted rules are both side unexpected rules according to the considered query, i.e. rules R16, R12, R9, R1, R6, R14. Neither the antecedent nor the consequent parts of these rules are projections of query Q.

Other queries can be built on other classical associations, depending on the user objectives and expectation in each situation, for example:

- [Context] → (implies) → [Solution]
- [Analysis] → (implies) → [Solution]
- [Context] [Analysis] → (implies) → [Solution]

Rules fitting with these queries may help to efficiently orientate human decision making when an urgent maintenance is needed in a given recognized situation, but may also help to identify recurrent problems on some machines, which could be solved by corresponding preventive maintenance activities.

5. Conclusion

This paper describes a framework for the development of an Experience Feedback process in maintenance, taking benefit of the potential of CMMS for providing a huge volume of information on past experiences that can be translated into new useful knowledge. The use of Conceptual Graphs at various steps of the process is one of the originalities of our study: we have shown that CGs may be useful (*i*) to model the past experiences and extracted rules, according to a support (ontological knowledge) defined for each application; (*ii*) to help the user for modeling his expectations regarding the extracted knowledge; (*iii*) to filter the mined rules according to the user's expectations; (*iv*) to represent the links between the identified rules; (*v*) to provide a metamodel generalizing a rule-set and providing a new abstraction of the extracted knowledge.

A specific emphasis has been set on the semantic interpretation of the extracted rules: we have suggested a step-by-step systematic approach for facilitating this interpretation, considering first "elementary" rules, then complex ones. We have also shown that additional knowledge may come from the identification of the links between mined rules, and from a comparison between present and absent rules. On the base of a limited but real case study, we have also shown that the extracted rules may not only help to structure knowledge on how the maintenance activities should be performed, but may in some cases detect problems in the way the database is structured, or in the way the information is entered.

Future investigations firstly aim at improving the process of association rules mining, by an in depth analysis of the influence of the support and confidence chosen by the user. In order to explore both frequent and rare data (which have different interests), the objective is to analyse the interest of generating several sets of rules, based on different support and confidence thresholds, then to compare the obtained knowledge databases. We shall also consider the use of other mining algorithms in order to obtain other type of rules, and in order to optimize memory and time used on large databases. Three new applications are in progress, in the aeronautical, pharmaceutical and petro-chemical domains, using more complex databases with more cases and much more attributes for describing an experience. This complexity will be the base of new opportunities for improving and optimizing the method, especially concerning the interpretation of the formalized knowledge.

References

[1] A. Aamodt, E. Plaza, Case-based reasoning: foundational issues, methodological variations, and system approaches, AI Commun. 7 (1) (1994) 39–52.

[2] R. Agrawal, T. Imielinski, A. Swami, Mining association rules between sets of items in large databases, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD'93, May 26–28, ACM, Washington, DC, USA, 1993, pp. 207–216.

[3] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, Proceedings of the 20th International Conference on Very Large Data Bases, VLDB'94, September 12–15, Vol. 1215, Morgan Kaufmann Publishers Inc., 1994, pp. 487–499.

[4] I. Alsyouf, Maintenance practices in Swedish industries: survey results, Int. J. Product. Econ. 121 (1) (2009) 212–223.

[5] J.F. Baget, O. Corby, R. Dieng-Kuntz, C. Faron-Zucker, F. Gandon, A. Giboin, A. Gutierrez, M. Leclère, M.L. Mugnier, R. Thomopoulos, Griwes: generic model and preliminary specifications for a graph-based knowledge representation toolkit, in: P. Eklund, O. Haemmerle (Eds.), Knowledge Visualization and Reasoning, Lecture Notes in Artificial Intelligence, vol. 5113, Springer-Verlag, Berlin, 2008, pp. 297–310.

[6] J.F. Baget, M.L. Mugnier, Extension of simple conceptual graphs: the complexity of rules and constraints, J. Artif. Intell. Res. 16 (2002) 425–465.

[7] J. Baohui, W. Yuxin, Y. Zheng-qing, The research of data mining in ahm technology based on association rule, in: Prognostics and System Health Management Conference (PHM), May 24–25, IEEE, Shenzhen, 2011, pp. 1–8.

[8] D. Batanov, N. Nagarur, P. Nitikhunkasem, Expert-mm: a knowledge-based system for maintenance management, Artif. Intell. Eng. 8 (4) (1993) 283–291.

[9] M. Ben-Daya, S.O. Duffuaa, A. Raouf, J. Knezevic, D. Ait-Kadi, Handbook of Maintenance Management and Engineering, vol. 1, Springer, 2009.

[10] S. Benomrane, M.B. Ayed, A.M. Alimi, An agent-based knowledge discovery from databases applied in healthcare domain, in: International Conference on Advanced Logistics and Transport', ICALT 2013, May 29–31, Tunisia, 2013.

[11] R. Bergmann, Experience Management: Foundations, Development Methodology, and Internet-Based Applications, Springer-Verlag, Berlin, 2002.

[12] A. Borgida, On the relative expressiveness of description logics and predicate logics, Artif. Intell. 82 (1/2) (1996) 353–367.

[13] J. Breuker, A cognitive science perspective on knowledge acquisition, Int. J. Hum.–Comp. Stud. 71 (2) (2013) 177–183.

[14] A.P. Chassiakos, P. Vagiotas, D.D. Theodorakopoulos, A knowledge-based system for maintenance planning of highway concrete bridges, Advan. Eng. Softw. 36 (11–12) (2005) 740–749.

[15] M. Chein, M.-L. Mugnier, Conceptual graphs: fundamental notions, Revue d'Intelligence Artificielle 6 (4) (1992) 365–406.

[16] M. Chein, M.L. Mugnier, Graph-Based Knowledge Representation: Computational Foundations of Conceptual Graphs, Springer Publishing Company, Incorporated, 2008.

[17] E. Chen, H. Cao, Q. Li, T. Qian, Efficient strategies for tough aggregate constraint-based sequential pattern mining, Inform. Sci. 178 (6) (2008) 1498–1518.

[18] W.C. Chen, S.S. Tseng, C.Y. Wang, A novel manufacturing defect detection method using association rule mining techniques, Expert Syst. Appl. 29 (4) (2005) 807–815.

[19] Y.J. Chen, Development of a method for ontology-based empirical knowledge representation and reasoning, Dec. Supp. Syst. 50 (1) (2010) 1–20.

[20] F. Chiang, R.J. Miller, Discovering data quality rules, Proc. VLDB Endown. 1 (1) (2008) 1166–1177.

[21] A.K. Choudhary, J.A. Harding, M.K. Tiwari, Data mining in manufacturing: a review based on the kind of knowledge, J. Intell. Manufact. 20 (5) (2009) 501–521.

[22] P. Coupey, C. Faron, Towards correspondences between conceptual graphs and description logics, in: M.L. Mugnier, M. Chein (Eds.), Conceptual Structures:

Theory, Tools and Applications, Lecture Notes in Computer Science, vol. 1453, Springer, Berlin Heidelberg, 1998, pp. 165–178.

[23] A. Crespo Marquez, J.N.D. Gupta, Contemporary maintenance management: process, framework and supporting pillars, Omega – Int. J. Manage. Sci. 34 (3) (2006) 313–326.

[24] T. Dasu, T. Johnson, Exploratory Data Mining and Data Cleaning, vol. 479, Wiley.com, 2003.

[25] L. Delange, R. Vogin, 'La croissance de sûreté de fonctionnement par le retour d'expérience dans le domaine technique et industriel', Performances Humaines et Techniques 69 (1994).

[26] R. Dieng-Kuntz, O. Corby, Conceptual graphs for semantic web applications, in: F. Dau, M.L. Mugnier, G. Stumme (Eds.), Conceptual Structures: Common Semantics for Sharing Knowledge, Lecture Notes in Artificial Intelligence, vol. 3596, Springer-Verlag, Berlin, 2005, pp. 19–50.

[27] EN13306:2001, Maintenance Terminology – European Standard. CEN (European Committee for Standardization), Brussels, 2001.

[28] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, Advances in Knowledge Discovery and Data Mining, MIT Press, 1996.

[29] P. Fournier-Viger, C.W. Wu, V.S. Tseng, Mining top-k association rules, in: L. Kosseim, D. Inkpen (Eds.), Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol. 7310, Springer, Berlin Heidelberg, 2012, pp. 61–73.

[30] F. Fürst, M. Leclère, F. Trichet, Ontological engineering and mathematical knowledge management: a formalization of projective geometry, Ann. Math. Artif. Intell. 38 (1) (2003) 65–89.

[31] F. Fürst, F. Trichet, Axiom-based ontology matching, Expert Syst. 26 (2) (2009) 218–246.

[32] E. Garcia, C. Romero, S. Ventura, C. de Castro, An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering, User Model. User-Adap. Interact. 19 (1–2) (2009) 99–132.

[33] G. Gasmi, S. Ben Yahia, E. Mephu Nguifo, Y. Slimani, 'Igb: une nouvelle base générique informative des règles d'association', Revue I3 (Inform. Interact. Intell.) 6 (1) (2006) 31–67.

[34] L. Geng, H.J. Hamilton, Interestingness measures for data mining: a survey, ACM Comput. Surv. 38 (3) (2006). Article 9.

[35] P. Giudici, Applied Data Mining: Statistical Methods for Business and Industry, Wiley, 2003.

[36] T.R. Gruber, Toward principles for the design of ontologies used for knowledge sharing, Int. J. Hum.–Comp. Stud. 43 (5–6) (1995) 907–928.

[37] J. Han, M. Kamber, Data Mining Concept and Technology, Publishing House of Mechanism Industry, 2001. pp. 70–72.

[38] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2006.

[39] J.A. Harding, M. Shahbaz, Srinivas, A. Kusiak, Data mining in manufacturing: a review, J. Manuf. Sci. Eng. – Trans. ASME 128 (4) (2006) 969–976.

[40] J. Hogan, F. Hardiman, M.D. Daragh Naughton, Asset management: a review of contemporary & individualised strategies, Lect. Notes Eng. Comp. Sci. 2190 (1) (2011) 545–549.

[41] R. Huang, L. Xi, J. Lee, C.R. Liu, The framework, impact and commercial prospects of a new predictive maintenance system: intelligent maintenance system, Product. Plann. Control 16 (7) (2005) 652–664.

[42] B. Kamsu-Foguem, T. Coudert, C. Béler, L. Geneste, Knowledge formalization in experience feedback processes: an ontology-based approach, Comp. Indust. 59 (7) (2008) 694–710.

[43] B. Kamsu-Foguem, F. Rigal, F. Mauget, Mining association rules for the quality improvement of the production process, Expert Syst. Appl. 40 (4) (2013) 1034–1045.

[44] G. Köksal, I. Batmaz, M.C. Testik, A review of data mining applications for quality improvement in manufacturing industry, Expert Syst. Appl. 38 (10) (2011) 13448–13467.

[45] D. Kolb, Strategic Learning in a Knowledge Economy, The Process of Experiential Learning, Butterworth-Heinemann, 2000. chapter 15, pp. 313–331.

[46] K.U. Koskinen, Problem absorption as an organizational learning mechanism in project-based companies: process thinking perspective, Int. J. Project Manage. 30 (3) (2012) 308–316.

[47] L.F. Lai, A knowledge engineering approach to knowledge management, Inform. Sci. 177 (19) (2007) 4072–4094.

[48] D.T. Larose, Discovering Knowledge in Data: An Introduction to Data Mining, Wiley-Interscience, 2005.

[49] E. Lazcorreta, F. Botella, A. Fernandez-Caballero, Towards personalized recommendation by two-step modified apriori data mining algorithm, Expert Syst. Appl. 35 (3) (2008) 1422–1429.

[50] S.H. Liao, W.C. Fei, C.T. Liu, Relationships between knowledge inertia, organizational learning and organization innovation, Technovation 28 (4) (2008) 183–195.

[51] J. Lieber, Application of the revision theory to adaptation in case-based reasoning: the conservative adaptation, in: Proceedings of Case-Based Reasoning Research and Development', Lecture notes in Computer Science, vol. 4626, 2007, pp. 239–253.

[52] J. Liebowitz, Knowledge management and its link to artificial intelligence, Expert Syst. Appl. 20 (1) (2001) 1–6.

[53] B. Liu, W. Hsu, K. Wang, S. Chen, Visually aided exploration of interesting association rules, in: N. Zhong, L. Zhou (Eds.), Methodologies for Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, vol. 1574, Springer, Berlin Heidelberg, 1999, pp. 380–389.

[54] Y. Liu, W. Xu, H. Du, The method of test for state of railway tunnel lining based on association rules, in: 3rd International IEEE Conference on Communication Software and Networks (ICCSN), May 27–29, Xi'an, 2011, pp. 387–390.

[55] S. Létourneau, F. Famili, S. Matwin, Data mining to predict aircraft component replacement, IEEE Intell. Syst. Appl. 14 (6) (1999) 59–66.

[56] J. Lu, D. Ruan, Intelligent knowledge engineering systems, Knowl.-Based Syst. 20 (5) (2007) 437–438.

[57] A. Maquee, A.A. Shojaie, D. Mosaddar, Clustering and association rules in analyzing the efficiency of maintenance system of an urban bus network, Int. J. Syst. Assur. Eng. Manage. 3 (3) (2012) 175–183.

[58] C. Marinica, Association Rule Interactive Post-processing using Rule Schemas and Ontologies-ARIPSO, PhD thesis, Ecole polytechnique de l'Université de Nantes, 2010.

[59] D.L. McGuinness, A.T. Borgida, Explaining subsumption in description logics, Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI'95, vol. 1, Morgan Kaufmann Publishers Inc., 1995, pp. 816–821.

[60] R.J. Meseroll, C.J. Kirkos, R.A. Shannon, Data mining navy flight and maintenance data to affect repair, in: 42nd Annual AUTOTESTCON Conference, September 17–20, vols. 1 and 2, IEEE, Baltimore, MD, 2007, pp. 476–481.

[61] M. Minor, Introduction strategy and feedback from an experience management project, in: K.D. Althoff, A. Dengel, R. Bergmann, M. Nick, T. Roth-Berghofer (Eds.), Professional Knowledge Management, Lecture Notes in Computer Science, vol. 3782, Springer-Verlag, Berlin, 2005, pp. 284–292.

[62] M. Minsky, A framework for representing knowledge, Psychol. Comp. Vis. (1975) 211–277.

[63] A. Mirabadi, S. Sharifian, Application of association rules in iranian railways (rai) accident data analysis, Safety Sci. 48 (10) (2010) 1427–1435.

[64] J. Moubray, Reliability Centred Maintenance, Butterworth-Heinemann, 1991.

[65] M.L. Mugnier, On generalization/specialization for conceptual graphs, J. Exper. Theoret. Artif. Intell. 7 (3) (1995) 325–344.

[66] M.L. Mugnier, Knowledge representation and reasonings based on graph homomorphism, in: B. Ganter, G.W. Mineau (Eds.), Proceedings of Conceptual Structures: Logical, Linguistic, and Computational Issues, Lecture Notes in Artificial Intelligence, vol. 1867, Springer-Verlag, Berlin, 2000, pp. 172–192.

[67] J. Nahar, T. Imam, K.S. Tickle, Y.P.P. Chen, Association rule mining to detect factors which contribute to heart disease in males and females, Expert Syst. Appl. 40 (4) (2013) 1086–1093.

[68] T. Nakagawa, Periodic and sequential preventive maintenance policies, J. Appl. Probab. 23 (2) (1986) 536–542.

[69] S. Nakajima, Introduction to TPM: Total Productive Maintenance, Productivity Press, Cambridge, Mass, 1988 (translated into English from the original text published by the Japan Institute for Plant Maintenance, 1984).

[70] J. Nobécourt, B. Biébow, Mdws: a modelling language to build a formal ontology in either description logics or conceptual graphs, 12th International Conference on Knowledge Engineering and Knowledge Management. Methods, Models, and Tools, EKAW 2000, Juan-les-Pins, France, October 2–6, vol. 1937, Springer, 2000, pp. 57–64.

[71] G. Pieri, M. Klein, M. Milanese, Maic: a data and knowledge-based system for supporting the maintenance of chemical plant, Int. J. Product. Econ. 79 (2) (2002) 143–159.

[72] M. Quillian, Semantic memory, in: M. Minsky (Ed.), Semantic Information Processing, MIT Press, 1968, pp. 227–270.

[73] H. Rakoto, P. Clermont, L. Geneste, Elaboration and exploitation of lessons learned, Intell. Inform. Process. 93 (2002) 297–300.

[74] I. Rasovska, B. Chebel-Morello, N. Zerhouni, A mix method of knowledge capitalization in maintenance, J. Intell. Manufact. 19 (3) (2008) 347–359.

[75] T. Rosqvist, K. Laakso, M. Reunanen, Value-driven maintenance planning for a production plant, Reliab. Eng. Syst. Safety 94 (1, SI) (2009) 97–110.

[76] W. Sammouri, E. Come, L. Oukhellou, P. Aknin, C.-E. Fonlladosa, K. Prendergast, Temporal association rule mining for the preventive diagnosis of onboard subsystems within floating train data framework, in: 15th International IEEE Conference on Intelligent Transportation Systems (ITSC), September 16–19, IEEE, Anchorage, AK, 2012, pp. 1351–1356.

[77] G. Schreiber, Chapter 25 knowledge engineering, in: V.L. Frank van Harmelen, B. Porter (Eds.), Handbook of Knowledge Representation, Foundations of Artificial Intelligence, vol. 3, Elsevier, 2008, pp. 929–946.

[78] L. Shen, F. Tay, L. Qu, Y. Shen, Fault diagnosis using rough sets theory, Comput. Indust. 43 (1) (2000) 61–72.

[79] A. Silberschatz, A. Tuzhilin, User-assisted knowledge discovery: how much should the user be involved, in: SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, Montreal, 1996.

[80] J. Sowa, Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley Publishing Company, Reading, MA, 1984.

[81] J.-F. Sowa, Knowledge Representation: Logical, Philosophical, and Computational Foundations, vol. 13, MIT Press, 2000.

[82] J.-F. Sowa, J.-A. Zachman, Extending and formalizing the framework for information-systems architecture, IBM Syst. J. 31 (3) (1992) 590–616.

[83] T. Stewart, Intellectual Capital: The New Wealth of Organizations, Doubleday/Currency, New York, 1997.

[84] R. Studer, V.-R. Benjamins, D. Fensel, Knowledge engineering: principles and methods, Data Knowl. Eng. 25 (1–2) (1998) 161–197.

[85] Z. Sun, A waterfall model for knowledge management and experience management, in: M. Ishikawa, S. Hashimoto, M. Paprzycki, E. Barakova, K. Yoshida, M. Koppen, D. Corne, A. Abraham (Eds.), Proceedings of the 4th

International Conference on Hybrid Intelligent Systems (HIS 04), IEEE Computer Soc., pp. 472–475.

[86] L. Swanson, An information–processing model of maintenance management, Int. J. Product. Econ. 83 (1) (2003) 45–64.

[87] R. Thomopoulos, J.-R. Bourguet, B. Cuq, A. Ndiaye, Answering queries that may have results in the future: a case study in food science, Knowl.-Based Syst. 23 (5) (2010) 491–495.

[88] M. Uschold, M. Grüninger, Ontologies: principles, methods and applications, Knowl. Eng. Rev. 11 (2) (1996) 93–136.

[89] K. Wang, Applying data mining to manufacturing: the nature and implications, J. Intell. Manufact. 18 (4) (2007) 487–495.

[90] T. Young, M. Fehskens, P. Pujara, M. Burger, G. Edwards, Utilizing data mining to influence maintenance actions, in: AUTOTESTCON Systems Readiness Technology Conference, September 13–16, IEEE, Orlando, FL, 2010, pp. 267–271.

[91] K.-M. Yu, J. Zhou, Parallel tid-based frequent pattern mining algorithm on a pc cluster and grid computing system, Expert Syst. Appl. 37 (3) (2010) 2486–2494.

[92] R.Q. Zhang, J.L. Yang, Association rules based research on man-made mistakes in aviation maintenance: a case study, in: Y. Chen, A. Abraham (Eds.), 6th International Conference on Intelligent Systems Design and Applications (ISDA'06'), vol. 1, IEEE Computer Soc., Jinan, 2006, pp. 545–550.