Improving maintenance strategies from experience feedback

Paula Potes-Ruiz, Bernard Kamsu-Foguem, Bernard Grabot

Improving Maintenance Strategies from Experience Feedback

P. Potes Ruiz, B. Kamsu Foguem, B. Grabot

Laboratoire Génie de Production / INP-ENIT - Université de Toulouse
47, Avenue d'Azereix, BP 1629, F-65016 Tarbes Cedex - France
{paula.potesruiz, bernard.kamsu-foguem, bernard.grabot}@enit.fr

Abstract: A huge amount of rough data is available in companies on past maintenance activities as a result of the implementation of CMMS (Computerized Maintenance Management System). In that context, we focus on an experience feedback system dedicated to maintenance, allowing the capitalization of past interventions by means of a formal knowledge representation language, and the extraction from these interventions of new knowledge for future reuse.

Keywords: Maintenance, Experience Feedback (EF), Computerized Maintenance Management System (CMMS), Conceptual Graphs (CGs), Knowledge Formalization, Association Rules Mining.

1. INTRODUCTION

According to the European Standards (EN 13306:2001), 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".

Many maintenance strategies have been developed in the last decades and applied to a large array of industries, such as Reliability Centered Maintenance (RCM) (Moubray, 1991) or Total Productive Maintenance (TPM) (Nakajima, 1988). Nevertheless, the idea of individualized maintenance strategies, dedicated to a given company, has recently emerged with the emphasis on "knowledge-based enterprise". The main objective of such methods is to use the immaterial resources of each organization in order to increase the economic benefit resulting from the construction of a maintenance strategy adapted to the requirements and resources of each organization (Hogan et al., 2011).

Even if knowledge is the base of human activity, only a part of it ("explicit knowledge") is easily accessible and reusable. Making explicit the "implicit" (or tacit) knowledge is the objective of Knowledge Engineering (Stewart, 1997), which has been object of an increased attention, especially from large companies, but has also shown the difficulty to identify, structure, store, and reuse knowledge (Minor, 2005). A consequence is for instance the recent interest of companies for the "Web 2.0" functionalities, especially blogs, wikis, and social networks, supposed to allow an easier collection of knowledge, if properly combined with semantic web technologies (Carbone et al., 2012).

Knowledge may be directly formalized by human experts, but it is often a long and complex task (Minor, 2005). On the other hand, it may also be extracted from information related to past experiences stored in the information system of the company: learning from these experiences has therefore become a very active field (Liao et al., 2008). In the domain of maintenance, the generalization of CMMS (Computerized Maintenance Management System), especially in large companies, makes available a lot of information provided by technicians after a maintenance intervention, often only used for traceability purpose. This information can be processed in order to allow extraction of useful knowledge for maintenance activities, then its reuse. In that purpose, it is necessary to determine how to formalize the information on past interventions, and how to extract meaningful knowledge based on their analysis, for the final benefit of "knowledge-based" maintenance strategies.

The rest of the paper is organized as follows: section 2 investigates the state of the art on industrial maintenance and on the experience feedback process. Section 3 describes a model of experience feedback in maintenance emphasizing the data processing phase, while section 4 presents an illustrative example.

2. STATE OF THE ART

2.1 Industrial maintenance

Nowadays, the performance of the companies depends to a large extent on their performance in maintenance. The increasing complexity of the industrial equipments makes it difficult for the users to operate and maintain them. Thus, the maintenance tasks are becoming more and more complex and diverse, involving not only activities on mechanical components, but also on electronic, hydraulic, electromechanical systems, and software (Alsyouf, 2009). As already emphasised, two main types of actions may be distinguished in maintenance: actions for retaining and actions for restoring a service. Thus, a classical taxonomy of maintenance distinguishes preventive maintenance from corrective maintenance (EN 13306:2001).

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 (Crespo Marquez and Gupta, 2006). Thus, since the individual knowledge and experience of the actors of maintenance cannot address all these fields, it is important to give them real time access to complementary knowledge. In this communication, we focus on the external source of knowledge that may result from a capitalization and processing of the past maintenance experiences, with the aim of creating a helpful experience-based knowledge for reuse.

2.2 Experience Feedback (EF)

Experience management approaches have become a strategic need for enterprises (Delange and Vogin, 1994) and are often included in knowledge management systems. A common point is that experience management also deals with collecting, modelling, storing, evaluating, and maintaining experience (Bergmann, 2002). The main interest of experience management is that it is easier for actors to formalize their expertise from lived experiences than to try to describe a non-contextualized generic knowledge (Kolb, 2000). A close relationship exists between knowledge and experience, since an experience might be considered as a specialization of knowledge, or as a singular instance (or form) of previous knowledge (Sun, 2005).

Given the importance of managing the experiences properly, experience feedback can be defined as a structured approach for capitalization, processing and exploitation of information derived from the analysis of positive and/or negative events (Rakoto et al., 2002). We shall consider the three classic phases of the EF process (Fig. 1): information *capitalization* from past interventions in a database (called "EF database"), information *processing* in order to formalize the experiences and extract useful knowledge from their analysis, and finally *exploitation* of EF database for the development or improvement of maintenance strategies. We propose to adapt this approach to the field of industrial maintenance with appropriate tools in each phase.

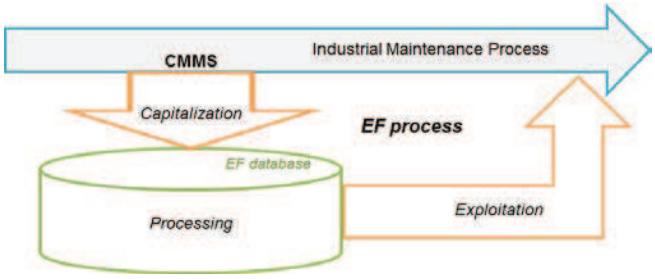

Fig. 1. EF approach in industrial maintenance.

3. EXPERIENCE FEEDBACK PROCESS IN MAINTENANCE

Problem solving methodologies based on past experiences play an essential role in improving maintenance strategies. The reuse of past experiences in order to provide a solution when a new problem occurs is a very active field. The best known technique in that purpose is certainly Case Based Reasoning (CBR) (Aamodt and Plaza, 1994) which adapts the solution of a past (but close) problem to a new one. Nevertheless, CBR does not always result in new generic knowledge. In order to obtain such knowledge, our target is to formalize the knowledge contained in past experiences as *rules*. These rules should be evaluated and validated by experts before their integration in the industrial maintenance process.

In order to reach this objective, we suggest to distinguish three different levels in the EF database: the *information database*, the *experiences database*, and the *rules database* (Fig. 2).

The aim of the processing phase is to formalize information from past interventions and to store it in the *information database*. We consider two types of information processing: *i)* basic information processing to formalize the experiences and store them in the *experience database* in order to have a good traceability for future reuse, and *ii)* a more synthetic way to process information, aiming at discovering new knowledge from an analysis of past interventions. This process is of course the critical step of the methodology, since it should result in generalized knowledge stored in the *rules database*, much more valuable than a list of experiences.

The major challenges are therefore how to formalize past experiences for a future improvement of maintenance strategies, and how to analyse and discover generic knowledge from information on past interventions in order to incorporate it in the industrial maintenance process. A descriptive scheme of our study is presented in Fig. 2.

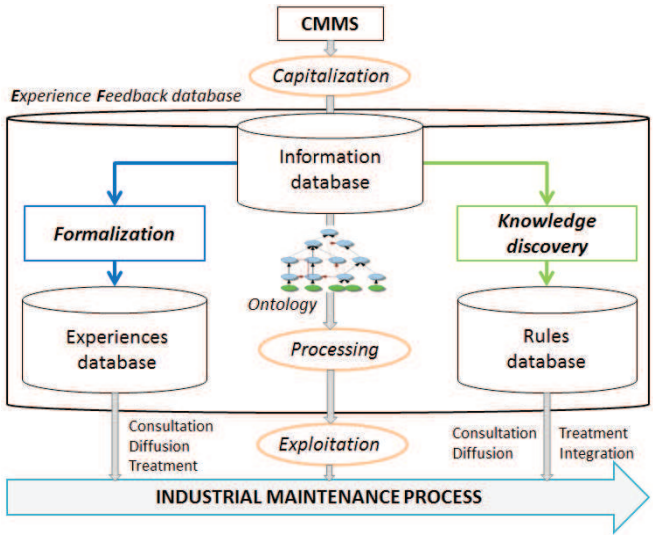

Fig. 2. General scheme of our EF process in maintenance.

3.1 Experience formalization

It is important to provide a formal knowledge representation and reasoning approach for allowing explications and sharing of knowledge (Chen, 2010). Since maintenance is a matter of communication between operators and experts of various fields, we draw a specific attention to the representation languages dedicated to ontologies, ensuring that information/knowledge exchanged by different actors is meaningful and that all the stakeholders interpret it in the

same way (Uschold and Grüninger, 1996). In our context, an ontology corresponds to a description of knowledge at the conceptual level, specifying the vocabulary of the maintenance domain and the semantics of its conceptual vocabulary (Fürst and Trichet, 2009). At the operational level, a knowledge representation formalism dedicated to ontologies is required to specify how the knowledge modelled in the ontology will be used for reasoning, i.e. what semantics, axioms and properties are required to use the ontology in an operational way (Fürst et al., 2003).

In the experience feedback process, the constraint to integrate knowledge so that it can be shared and reused leads to use a formal semantic to describe the application system. Several types of knowledge representation languages may be used to represent an ontology: Frame based systems (Minsky, 1975), Semantic Networks (Quillian, 1968), Description Logics (DLs) (Borgida, 1996) and Conceptual Graphs (CGs) (Sowa, 1984). Among these formalisms, we consider that CGs is the most promising in the context of experience feedback. This formalism, introduced by Sowa (1984), allows both representation and reasoning. It is currently the only logic-based model that has a corresponding interpretation in graph theory (Thomopoulos et al., 2010). Knowledge representation in CGs is entirely graphical and close to an expression in natural language; the reasoning is based on graph operations. CGs allow to express various types of knowledge (Baget and Mugnier, 2002) and to structure and contextualize knowledge through nesting of graphs. Thanks to these properties, CGs allow on one hand, the formalization of conceptual and procedural knowledge of a target domain, and on the other hand, provide reasoning tools that facilitate the visualization and the verification of the modelled knowledge by end-users (Dieng-Kuntz and Corby, 2005).

Therefore, the first step that we suggest for the experience formalization is to create a tree structure for the domain knowledge (ontology), dedicated to both modelling of equipments and EF system, following the three main components of an experience: *context*, *analysis*, and *solution*. The context part concerns the description of the situation in which the event has occurred; the second part is the analysis, i.e. the search of causes of the current intervention; finally, the solution part allows to choose the actions to perform for solving this problem (i.e. selected maintenance activities).

CGs have several interests in our maintenance context, especially in the formalization of past experiences and in the formalization and evaluation of extracted rules. Thus, our choice is not only based on the potential for expression and understanding by the user, but also on the possibility to perform visual reasoning using graph operations.

3.2 Knowledge discovery

Knowledge discovery is essential in many domains, since it provides a better understanding of the data, and gives a basis for making decisions. Our goal is here to find the way to extract generic knowledge from an analysis of past interventions, then to evaluate the results obtained in order to provide a validated rule database for a reuse.

Knowledge Discovery in Databases (KDD) usually includes several steps, shown in Fig. 3, among which Data Mining (DM) (Köksal et al., 2011).

In the field of DM, we are specifically interested in the domain of association rule mining, since procedural knowledge in the form of rules can be useful for at least two reasons, stated in (Marinica, 2010): *i)* the model of the extracted patterns is simple and comprehensible for a non-specialist user (the implications are the core of human thinking) and *ii)* during the process, a significant user implication is not required.

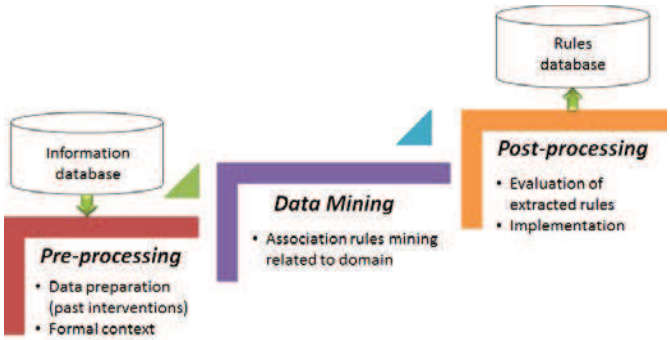

Fig. 3. Main steps of the knowledge discovery process.

Thus, we consider the three general phases in the knowledge discovery process for the rules extraction: *data pre-processing*, *data mining*, and *post-processing*.

3.2.1 Data pre-processing

This step aims at improving data quality by techniques such as data cleaning, data transformation, data reduction or discretisation. This is a very important step in the process, with the objective to organize the data in a way suitable and appropriate for the mining step. In that purpose, we build a "formal context" of past interventions taking into account the ontology, defined as a triplet $D = (O, I, R)$, such that D is the database, O is a set of transactions (interventions), I is a set of items (concepts defined in the ontology), and R is a binary relation between O and I.

3.2.2 Data Mining (DM)

DM is the most essential step in the knowledge discovery process, consisting in applying data analysis and discovery algorithms (Köksal et al., 2011) for generating knowledge. In our context, the goal is to derive association rules (Agrawal et al., 1993) linking the concepts of a modelled domain. An association rule is defined as an implication between two itemsets (antecedent and consequence), and represents the regularities of a database as implications of the form "if X then Y", denoted as $X \rightarrow Y$, where $X, Y \in I$ and $X \cap Y = \varnothing$.

Extraction can be performed by determining the rules which *support* (1) (i.e. frequency of occurrence, defined as the number of occurrences of the rule on the total number of cases) and *confidence* (2) (i.e. strength of a rule, defined as the percentage of rule achievement when the antecedent part appears) are at least equal to user-predefined thresholds *minsup* and *minconf* (Gasmi et al., 2006).

$$Support\ (X \rightarrow Y) = Support\ (X \cup Y) \quad (1)$$

$$Confidence\ (X \rightarrow Y) = \frac{Support\ (X \cup Y)}{Support\ (X)} \quad (2)$$

After rules mining, redundant rules may appear, so that trivial or false ones; therefore, the user has to evaluate and validate the extracted rules in the post-processing phase taking into account his goals and the domain knowledge.

3.2.3 Post-processing

Post-processing is the assessment of the utility and reliability of the mined rules, then the interpretation of the discovered information (Giudici, 2003). Following the *objective evaluation* performed by rule mining algorithms, we suggest to perform a *subjective evaluation* in order to improve the quality of rules, thanks to a post-processing method where user intervention and semantic criteria are needed.

Our aim is to use the semantics related to the ontology (domain knowledge) and the user expectation (user knowledge), expressed by a "model" of rule that the user expects, to evaluate the extracted rules. In that purpose, we suggest a query/answering mechanism using the "projection" operation, based on operations of graphs defined in the CGs. The fact that the same language (CGs) is used as an interface and as an operational tool makes transparent the logical structure of information, facilitating the understanding and interpretation of the results by the user (Mugnier, 2000).

Projection is generally defined by a homomorphism. Given two graphs G and H, H ≤ 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 ≤ G implies the existence of a projection from G to H (Mugnier, 2000).

We use the query/answering mechanism to analyse the discovered rules, by searching homomorphisms between the query graph (user expectation) and the "knowledge base" (Baget et al., 2010), in our context, the extracted rules database. Then, it is possible to classify the results obtained after the projection operation in different groups of rules potentially useful for the user. Four sets of rules can be defined (Liu et al., 1999): conforming rules (if both the antecedent and consequence are consistent with the user expectation), unexpected consequence rules (showing discovered rules which consequences are different from those expected), unexpected antecedent rules (showing other antecedents that can lead to the required result), and both-side unexpected rules (which are not known by the user or are not mentioned in its expectations).

In Fig. 4, are illustrated the data mining and post-processing phases of the KDD process.

4. APPLICATION EXAMPLE

An illustrative (simplified) example will provide an overview of the expected system.

Several platforms and implementation tools for CGs have been proposed (Baget et al., 2008), allowing to define an ontology and to build the graphs. We have chosen the CoGui platform for this implementation: the CoGui editor[1] is a free graph-based visual tool, developed in Java, which allows building intuitive visual structures with reasoning capabilities. Essentially, this tool allows to build an ontology and a set of CGs representing assertions, usually called "facts" but in our context denoted as "experiences" (in the *experiences database*), and "rules" (in the *rules database*).

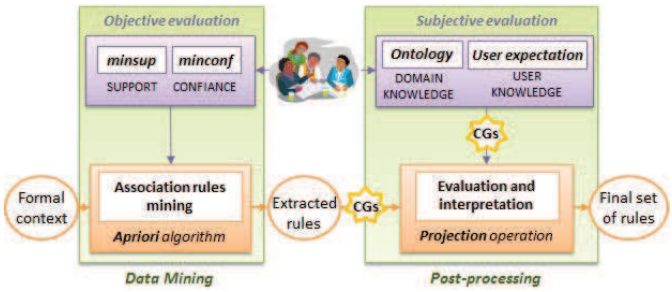

Fig. 4. Data mining and post-processing in KDD process.

We use here a real set of maintenance work reports on overhead cranes used to assemble different sections of aircrafts. The graphical representation in CGs contains concept nodes (indicated by boxes) and relation nodes (indicated by circles). A concept node is defined by a label and a marker that identify the considered instance (the "*" denotes an undefined instance) (Fürst and Trichet, 2009). Experiences and rules are represented by CGs.

4.1 Experience formalization with CGs

In Fig. 5, an event on a bridge crane is the basis of the experience. The CG for the experience is built according to the defined ontology. It can be interpreted as follows: in Experience 1, Context 1 requires Analysis 1, which generates Solution 1. More specifically, the context is described by the Work Order No 698188 for equipment POMC02002. We distinguish here the object of the maintenance act, the default (on 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 solution description concerns the type of intervention carried out and the actions performed (in this case, a technical assistance consisting in a realignment of the instrument).

4.2 Knowledge discovery

In our formal context, O = {*Intervention 1*, *Intervention 2*, *Intervention 3*,…}, I = {*type of equipment*, *faulty zone*, *cause of intervention*,...} and R is the binary relations between facts. We have chosen the SPMF[2] software (Sequential Pattern Mining Framework), which is an open-source Data Mining written in java, for association rules mining from a formal context.

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

[2] http://www.philippe-fournier-viger.com/spmf/

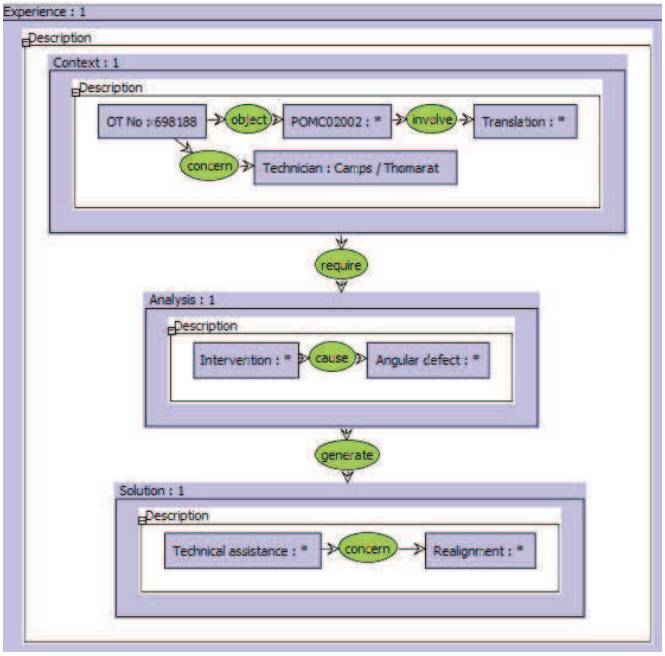

Fig. 5. Conceptual graph for the example.

The Apriori algorithm defined in (Agrawal and Srikant, 1994) is used for mining association rules on the base of a *minsup*, a *minconf*, and a *binary context*. In this example, minsup = 20% and minconf = 90%, leading to the extraction of 16 rules (Table1).

Table 1. Extracted association rules

Rule ID	Rule	Sup (%)	Conf (%)
R_1	Reset-acknowledgement→Urgent corrective	0.29	0.91
R_2	Realignment→Angular defect	0.33	0.97
R_3	Realignment→Translation	0.34	0.99
R_4	Angular defect→Translation	0.42	0.99
R_5	Technical assistance,Realignment→Angular defect	0.20	0.99
R_6	Technical assistance,Angular defect→Realignment	0.20	1
R_7	Technical assistance,Realignment→Translation	0.20	1
R_8	Technical assistance,Angular defect→Translation	0.22	0.97
R_9	Realignment,Translation→Angular defect	0.33	1
R_{10}	Realignment,Angular defect→Translation	0.33	0.97
R_{11}	Realignment→Translation,Angular defect	0.33	0.99
R_{12}	Tech.assistance,Realignment,Translation→Angular defect	0.20	1
R_{13}	Tech.assistance,Realignment,Angular defect→Translation	0.20	0.90
R_{14}	Tech.assistance,Angular defect,Translation→Realignment	0.20	0.90
R_{15}	Tech.assistance,Realignment→Translation,Angular defect	0.20	0.91
R_{16}	Tech.assistance,Angular defect→Translation,Realignment	0.20	0.93

To analyse the extracted rules, we first use the CGs to represent *facts* that match the extracted rules and *queries* that correspond to the user expectations. The evaluation and interpretation of the results are done with the projection operation of CGs. In our query/answering mechanism, let us consider a temporary database composed of the set of extracted rules. A query (Q) is expressed through a CG in order to translate the user expectation. We show in Fig. 6 a user expectation (new query) that corresponds to query Q: the software looks for rules with the form [Solution]→(implies)→[Context].

The results of these projections have been classified in: *i)* conforming rules (R_3 and R_7 are the only rules consistent with the user query); *ii)* unexpected consequence rules (R_{11}, R_{15}, R_2, and R_5); *iii)* unexpected antecedent rules (R_8, R_{10}, R_{13}, and R_4); and *iv)* both-side unexpected rules (the rest of the evaluated rules: R_9, R_{12}, R_{16}, R_1, R_6, R_{14}).

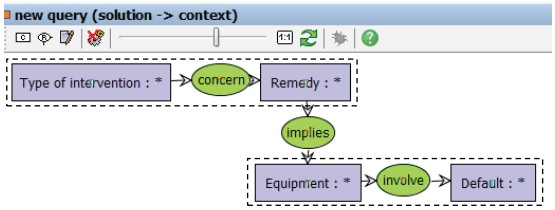

Fig. 6. User expectations (query Q).

5. CONCLUSION

This paper describes a framework for the development of an EF process in maintenance. We have suggested to use the potential of CMMS as knowledge sources, by analysing and transforming the huge volume of available information into useful knowledge associated to past maintenance experiences. In this approach, the role that plays the user in each step and the quality of information exported from CMMS are decisive and both affect the quality of the extracted results, evaluated using a formalism (CGs) that facilitates knowledge interpretation and use by the user.

The perspectives of this work are now in the exploitation phase of EF process for the final benefit of "knowledge-based" maintenance strategies relayed on experiences, in order to optimise the knowledge acquisition and performance of the industrial process.

REFERENCES

Aamodt, A. and Plaza, E. (1994). Case-based reasoning: foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications*, 7(1), 39-52.

Agrawal, R. and Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. In *Proc. of the 20th International Conference on Very Large Data Bases*, September 12-15, 487-499.

Agrawal, R., Imielinski, T. and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proc. of the ACM SIGMOD Intl. Conf. on Management of Data*, May 26-28, Washington, USA, 207-216.

Alsyouf, I. (2009). Maintenance practices in Swedish industries: Survey results. *International Journal of Production Economics*, 121(1), 212-223.

Baget, J.F., Corby, O., Dieng-Kuntz, R., Faron-Zucker, C., Gandon, F., Giboin, A., Gutierrez, A., Leclère, M., Mugnier, M.L. and Thomopoulos, R. (2008). Griwes: Generic Model and Preliminary Specifications for a Graph-Based Knowledge Representation Toolkit. In *Proc. of the 16th International Conference on Conceptual Structures (ICCS)*, July 7-11, Toulouse, France.

Baget, J.F. and Mugnier, M.L. (2002). Extensions of Simple Conceptual Graphs: the Complexity of Rules and Constraints. *Journal of Artificial Intelligence Research*,

16, 425-465.

Baget, J.F., Chein, M., Croitoru, M., Gutierrez, A., Leclère, M. and Mugnier, M.L. (2010). Logical, graph based knowledge representation with CoGui. *Atelier GAOC: Graphes et Appariement d'Objets Complexes en conjonction avec EGC'10*, January 26-29, Hammamet, Tunisia, 15-25.

Bergmann, R. (2002). *Experience Management: Foundations, Development Methodology, and Internet-Based Applications*, vol. 2432 of LNAI (Lecture Notes in Artificial Intelligence). Springer-Verlag, Berlin.

Borgida, A. (1996). On the relative expressiveness of description logics and predicate logics. *Artificial Intelligence*, 82(1/2), 353-367.

Carbone, F., Contreras, J., Herandez, J.Z. and Gomez-Perez, J.M. (2012). Open Innovation in an Enterprise 3.0 framework: Three case studies. *Expert System with Applications*, 39(10), 8929-8939.

Chen, Y.J. (2010). Development of a method for ontology-based empirical knowledge representation and reasoning. *Decision Support Systems*, 50(1), 1-20.

Crespo Marquez, A. and Gupta, J.N.D. (2006). Contemporary maintenance management: process, framework and supporting pillars. *Omega - International Journal of Management Science*, 34(3), 313-326.

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

Dieng-Kuntz, R. and Corby, O. (2005). Conceptual graphs for semantic web applications. In *Proc. of the 13th International Conference on Conceptual Structures (ICCS'05)*, July 17-23, Kassel, Germany, 19-50.

EN 13306:2001, 2001. *Maintenance Terminology. European Standard*. CEN (European Committee for Standardization), Brussels.

Fürst, F., Leclère, M. and Trichet, F. (2003). Ontological engineering and mathematical knowledge management: A formalization of projective geometry. *Annals of Mathematics and Artificial Intelligence*, 38(1), 65-89.

Fürst, F. and Trichet, F. (2009). Axiom-based ontology matching. *Expert Systems,* 26(2), 218-246.

Gasmi, G., Ben Yahia, S., Mephu Nguifo, E. and Slimani, Y. (2006). IGB: une nouvelle base générique informative des règles d'association. *Revue I3 (Information-Interaction-Intelligence)*, 6(1), 31-67.

Giudici, P. (2003). *Applied data mining: Statistical methods for business and industry*. New York: J. Wiley.

Hogan, J., Hardiman, F. and Daragh Naughton, M. (2011). Asset Management: A Review of Contemporary & Individualised Strategies. In *Proc. of the World Congress on Engineering*, Vol I, London, U.K.

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

Kolb, D. (2000). Chapter 15: The Process of Experiential Learning. *Strategic Learning in a Knowledge Economy*, 313-331.

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

Liu, B., Hsu, W., Wang, K. and Chen, S. (1999). Visually aided exploration of interesting association rules. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, April 26-28, 380-389.

Marinica, C. (2010). *Association Rule Interactive Post-processing using Rule Schemas and Ontologies - ARIPSO*. Thèse de doctorat, Ecole polytechnique de l'Université de Nantes, France.

Minor, M. (2005). Introduction strategy and feedback from an experience management project. In K.D. Althoff, A. Dengel, R. Bergmann, M. Nick, and T. Roth-Berghofer (ed.), *Professional Knowledge Management*. WM 2005 postconference proceedings., LNAI 3782, 284-292. Springer-Verlag.

Minsky, M. (1975). A framework for representing knowledge. In P. Winston (ed.), *The Psychology of Computer Vision*, 211-277, Mc Graw-Hill, New-York.

Moubray, J. (1991). *Reliability Centred Maintenance*. Butterworth Heinemann, UK.

Mugnier, M.L. (2000). Knowledge Representation and Reasonings Based on Graph Homomorphism. In *Proc.of the 8th International Conference on Conceptual Structures ICCS*, Lecture Notes in Computer Science, 1867/2000, 172-192.

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

Quillian, M.R. (1968). Semantic memory. In *semantic Information Processing*, 227-270, MIT Press.

Rakoto, H., Clermont, P. and Geneste, L. (2002). Elaboration and exploitation of lessons learned. *Intelligent Information Processing*, 93, 297-300.

Sowa, J.F. (1984). Conceptual structures: information processing in mind and machine. The Systems Programming Series, Addison-Wesley, Reading, Massachusetts, USA.

Stewart, T.A. (1997). *Intellectual Capital: the New Wealth of Organizations*. Doubleday Currency, New York.

Sun, Z. (2005). A waterfall model for knowledge management and experience management. In M. Ishikawa, S. Hashimoto, M. Paprzycki, E. Barakova, K. Yoshida, M. Koppen, D. Corne and A. Abraham (ed.), Proceedings of the 4th International Conference on Hybrid Intelligent Systems (HIS'04), IEEE Computer Soc., 472-475.

Thomopoulos, R., Bourguet, J.R., Cuq, B. and Nyiaye, A. (2010). Answering queries that may have results in the future: A case study in food science. *Knowledge-Based Systems*, 23(5), 491-495.

Uschold, M. and Grüninger, M. (1996). Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, 11(2), 93-155.