A mix method of knowledge capitalization in maintenance.

Ivana Rasovska, Brigitte Chebel-Morello, Nourredine Zerhouni

A mix method of knowledge capitalization in maintenance

Ivana Rasovska, Brigitte Chebel-Morello, Noureddine Zerhouni
Laboratoire d'Automatique de Besançon
24 rue Alain Savary, 250 00 Besançon, France
[rasovska,bmorello,zerhouni]@ens2m.fr

Abstract

This paper deals with knowledge capitalization in maintenance especially in diagnosis and repair of industrial equipments. The goal is to propose a method of knowledge capitalization in order to develop a decision support system for maintenance operators. The knowledge capitalization cycle was adopted as the underlying principle. It consists of four principal steps: detect, preserve, capitalize and actualize the strategic knowledge. Different knowledge management tools and methods that can be used in the cycle are reviewed. We propose a mix method of knowledge capitalization in maintenance. This method integrates a representation and a reasoning model both completing each other and suitable to represent and manipulate the domain knowledge. The knowledge representation model using UML diagram proposes different domain models based on maintenance analysis to guide the domain expertise. The reasoning model uses the case-based reasoning which allows the manipulation of represented domain knowledge. Finally, the method is implemented on the pallet transfer system Sormel in the context of Proteus e-maintenance platform.

1. Introduction

This study makes part of our research work realized in the European project Proteus. The goal of this project was to develop distributed cooperative e-maintenance platform that proposes a set of decision help tools for different maintenance activities. Nowadays maintenance systems operate on very complex industrial equipments. Different automatic systems such as monitoring system or control command are installed on these equipments in order to realise conditional and predictive maintenance. They generate data and information bases in order to improve and extend proposed maintenance services. Further, enterprises ask for preservation and capitalization of the expert's and operator's know-how which is possible thanks to development of decision support systems. The power of having

good information where you need and when you need it facilitates problem solving and taking of strategic decisions.

The goal of this paper is to propose a mix method for knowledge capitalization in maintenance in order to develop a decision support system for diagnosis and repair of industrial equipments. The concept of such a decision support system is based on domain expertise supported by the cognitive modelling of process and knowledge used by maintenance operators during their activities. The decision help tool is targeted to these operators and developed by knowledge management techniques as well as artificial intelligence methods. The proposed mix method of knowledge capitalization is based on the knowledge capitalization cycle adopted as the underlying principle. This method consists of the knowledge model integrating the representation and the reasoning model. This integration permits, on the one side, to guide the domain expertise and to propose the aid for the knowledge modelling and representation in maintenance. On the other side, this represented knowledge can be manipulated by the selected problem solving method in the decision support system. The representation model uses UML class diagram to represent and model the domain knowledge which can be manipulated by the reasoning model based on the case-based reasoning. This method assures the ease of knowledge use and the dynamic aspect of knowledge acquisition and actualization from experts.

The paper is organised as follows. First, the knowledge capitalization cycle is introduced. Different knowledge management tools and methods that can be used in the cycle are reviewed. They are compared according to different criterions in order to choose the appropriate one. The third section concerns the proposed mix method of knowledge capitalization based on the representation and the reasoning model. The representation model is based on Unified Modelling Language and the reasoning model is the case-based reasoning. The implementation of the methodology is shown on the real case - the pallet transfer system Sormel. Finally, the architecture of the decision support system in the context of the e-maintenance platform Proteus is illustrated.

2. Knowledge capitalization cycle

Nowadays, knowledge is increasingly considered the most important asset of organisations and companies, especially within the service sector of knowledge-based industries. This knowledge, experiences and know-how of companies is stored and capitalized in order to be shared and so to become the intellectual capital (Rosario, 1996). This is the general purpose of knowledge management that can be defined also as a set of techniques, tools and activities focused on helping organizations capture and communicate their “resources, tacit and explicit

perspectives and capabilities, data, information, knowledge and maybe wisdom (competence)" (Jarboe, 2001). In a nutshell, the knowledge management is the overall task of managing the processes of knowledge creation, storage and sharing and related activities.

The "knowledge capitalization" results from the term of intellectual capital and means "to reuse, in a relevant way, the knowledge of a given domain previously stored and modelled, in order to perform new tasks" (Simon, 1996). It is also defined in (Matta*, et al,* 2001) as "the formalization of experience gained in a specific field". The principal purpose is to "locate and make visible the enterprise knowledge, be able to keep it, access it and actualize it, know how to diffuse it and better use it, put it in synergy and valorise it" (Grundstein, 1992). There are several models of knowledge management and capitalization (Holsapple & Joshi, 1999), (Lai & Chu, 2000) combining technical, human and organizational aspects. For our purpose, the knowledge capitalization cycle was chosen as the most appropriate because of its generalizing point of view. This cycle summarises the knowledge capitalization tasks in four major steps: detection, preservation, capitalization and actualization of strategic knowledge, each of them declined by several detailed tasks. These steps also reflect requirements for our objective, i.e. development of decision support system in maintenance.

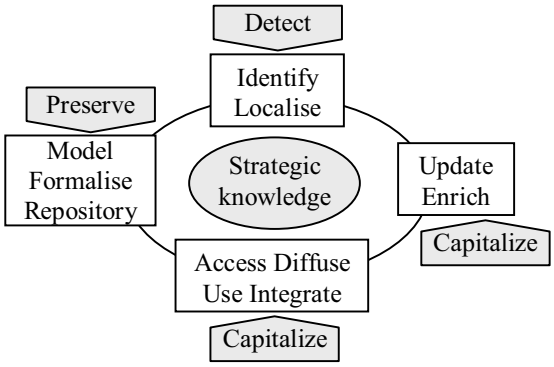

Figure 1. *Knowledge capitalization cycle (Grundstein, 1992)*

The knowledge capitalization cycle presented in (Barthès, 1996) and shown in fig. 1 was adopted as the underlying principle of our approach. The first step is to detect strategic knowledge - the "good" information that should be transmitted to the final user. The process of identification and localization of strategic knowledge depends on the knowledge management objectives and requirements (to whom the system is designed and for what purpose). The knowledge acquisition is issued from domain analysis, technical documents and interviews with experts. The second step is to preserve the acquired knowledge by its formalization, modelling and storing in

a repository model. The third step represents the knowledge capitalization, i.e. to make knowledge accessible, to integrate it and to diffuse it to final users. Finally, the knowledge actualization is based on experience feedback from final users. Previously stored knowledge is updated and eventually enriched by a new one. Different knowledge management methods and tools exist and can be used in this cycle. A review of these methods is introduced below.

2.1. Methodologies dedicated to the corporate memory

Several methodologies are directly designed to the building of enterprise "corporate memory" defined as "explicit, disembodied persistent representation of knowledge and information in an organization (Van Heijst et al., 1996). This corporate memory stores identified and localized strategic knowledge. Several examples of such methods are REX (Malvache & Prieur, 1993), MEREX (Corbel, 1997), Workshop FX (Foeut, 1997) or CYGMA (Bourne, 1997). REX method is due to the capitalization of experiences achieved during the enterprise activity realisation. The methods's application relies on three main steps: analysis and identification of needs and information resources, experience elements building and development and exploitation of knowledge management system. Three types of experience elements exist: documented knowledge elements, experience elements issued from interview with experts and know-how elements issued from particular activity. They are stored in the "experience memory" that contains a terminology net introducing a vocabulary and viewpoints used in the enterprise and a description model in form of semantic net. This method proposes a software tool for memorisation and diffusion of experience elements but does not provide their exploitation. Experience elements can be retrieved in answer to a natural language request during the knowledge management system exploitation.

2.2. Methodologies of knowledge engineering

Other methodologies of knowledge capitalization make part of knowledge engineering domain. Knowledge engineering is defined in (Charlet et al., 2000) as "study of concepts, methods and techniques to model and/or to acquire knowledge for decision support systems in order to realise tasks a priori hardly or not formalisable". Knowledge engineering proposes methods and tools (cognitive modelling, conceptual modelling, modelling and representation languages, etc.) for different domains such as knowledge acquisition, information research on the Web, knowledge management and capitalization, implementation of management indicators in information systems, etc. Among the others, methods like KOD (Vogel, 1988), KADS and CommonKADS (Wielinga et al., 1993), MKSM and its

extension MASK (Van Craeynest, et al., 2000) provide help to the knowledge representation. They can be dissociated into two different approaches of knowledge modelling (Motta et al., 1990), i.e. bottom-up (KOD) and top-down approach (KADS, MKSM). *The bottom-up approach* consists in collecting verbal data from domain experts and in gathering them to form a knowledge model. *The top-down approach* or “approach driven by models" is focused on the definition of the expertise model in order to filter acquired knowledge and to guide the acquisition process effectively. This comparison proved that top-down approach makes possible to separate knowledge from the domain of its use. The generic components can be so defined and reusable. On the contrary, even if the bottom-up approach provides a structured expertise model, this is described in the terminology suitable for the specific problem and is missing of abstraction. Thus our study was oriented to the top-down approach.

KADS (Knowledge Acquisition and Design Structuring) and its evolution CommonKADS (Wielinga et al., 1993) is one of the references among the top-down methods. Its structure in layer uses generic methods of problem solving describing the reasoning mechanisms on a good abstraction level. KADS relies on knowledge dissociation in three levels: domain elements, functional view of domain knowledge (inference) and control of reasoning phases in problem solving (task). This dissociation allows reusing of the domain elements model in different problem solving tasks. CommonKADS provides the knowledge analysis framework with an extensive method. This method describes business processes where knowledge-intensive tasks are carried out. Six models are proposed to the knowledge analyst: organization, task, agent, communication, expertise (operation performance) and design of knowledge based system. They give different viewpoints on domain knowledge and are used in the knowledge system development cycle. This cycle involves four main phases: objectives review, risk analysis, planning and monitoring. This method uses Unified Modelling Language (UML) for the model representation and the ontology for domain knowledge reusability.

MKMS (Method for Knowledge System Management) proposed by the CEA[1] was introduced in (Ermine, 1996). It aims at reducing knowledge system management complexity by using different methods at different grain levels. This system is modelled by three viewpoints: syntax (information), semantic (signification corresponding to task modelling) and pragmatic (context corresponding to activity modelling). Further, each of three viewpoints involves three other aspects, i.e. structure, function and evolution. Each viewpoint is represented by models with Object Modelling Technique and the set of instantiated and structured models is called the “knowledge book”. Five modelling phases are closed to CommonKADS notions: modelling of system, domain, activities, concept

[1] Commisariat à l’Energie Atomique

and tasks. MASK[2] (Method of knowledge analysis and structuration) is the extension of MKSM (Van Craeynest et al., 2000). It is based on cognitive principles of analysis and experience feedback. This concept allows representing of the expertise in three complementary points of view, i.e. the context (profession concerned approach), experience and know how (cognitive engineering approach) and computerizing (software engineering approach). This method takes into account the dynamic aspect of the knowledge acquisition.

These reference methods are not completely satisfactory and cannot take into account all characteristic types of problems. They are used in various works involving their adaptation. It gives a way to new perspectives by expertise models creation based on the integration of bottom-up and top-down approaches. Duribreux-Cocquebert et al. (2000) proposes a mix approach combining KADS and KOD methods by modelling from a corpus. The reusing of these methods can be partial, applied to the domain elements specifications or to the reasoning used in the method. Reynaud et al. (1997) declares: "… Applied to the domain elements, the reusing is based on the ontology definition describing explicitly the domain elements. Applied to the reasoning elements, it is based on abstract descriptions of the problem solving methods." Another method is the domain ontology. Information has to be clarified in form of concepts defined in the domain model representing explicit knowledge about the application area.

MOKA (A Methodology and tools Oriented to Knowledge-based engineering Applications) was developed within the MOKA Project. Its goal was to provide a methodology for developing and maintaining Knowledge Based Engineering (KBE) applications (Moka, 2000). A tool supporting this methodology was developed. The project identified the typical KBE life cycle involving six steps: identify, justify, capture, formalise, package and activate. MOKA focused on capture step by introducing the informal MOKA model to structure the raw engineering knowledge. It is a set of related ICARE forms: illustration, constraints, activities, rules and entities. This models are then translated into the MOKA formal model in the formalise step. The meta-models and views such as Product Model and Design Process Model were created with UML and some extensions defined in MOKA Modelling Language. They represent three viewpoints on domain knowledge: structure, function and behaviour.

2.3. Synthesis of reviewed methodologies

The synthesis of reviewed methodologies is presented in the table 1.

---

[2] Méthode d'Analyse et de Structuration des Connaissances

Methodology	Knowledge structure	Used models	Knowledge system development cycle	Knowledge exploitation
REX	Elements: document, experience, know-how	Experience memory: terminology and semantic net	Resources and need analysis Knowledge elements creation Implementation	Low because of low formalisation
CommonKADS	Task Inference Domain	Organisation Task Agent Knowledge Communication Design	Objectives review Risk analysis Planning Monitoring	High – possible direct exploitation
MASK	Information Context Sense With 3 aspects: structure, function, evolution	Reference system Domain Activity Concept Task	Framing Knowledge modelling Orientation scheme: strategy, policy, risk analysis	Medium – no direct exploitation
MOKA	Structure Function Behaviour	Informal (ICARE forms) : illustration, constraints, activity, rule, entities Formal: Product, Design Process	Identify Justify Capture Formalise Package Activate	High – methodology for knowledge based engineering applications

Table 1. *Synthesis of reviewed methodologies*

The methodologies are compared according to four different criterions: knowledge structure, used models, cycle of knowledge system development and finally the level of possible knowledge exploitation in a decision support system. The knowledge modelling and representation is in the core of all these methods. This consists in transformation of informal knowledge descriptions into the formal ones in order to handle this knowledge in the information system. Whilst the REX method distinguishes knowledge according to its resources (document, experience and experts know how), other methods apply three main points of view. Domain knowledge from CommonKADS represents the real world description and has the common characteristics as information viewpoint of MASK and structure viewpoint of MOKA. Inference knowledge of CommonKADS describes the functional viewpoint like a function in MOKA and a context in MASK. The task knowledge of CommonKADS defines the representation purpose and the adequate problem solving method corresponding to a sense in MASK and behaviour in MOKA.

Except REX which builds the experience memory based on terminological and semantic nets, other methods use different models to handle knowledge in the knowledge based system. REX has a low level of knowledge formalisation and do not allow a decision support system to reason automatically on domain knowledge. CommonKADS, MASK and MOKA propose help tools for knowledge representation and formalization. For this they employ their own system development cycle. The common point is the ontological aspect of the knowledge representation and the use of Unified Modelling Language and its eventual extensions (MOKA). Even thought the knowledge exploitation by a decision support system is not the primary goal of these methodologies, they have, in general, a good level of knowledge formalisation which facilitates its exploitation by problem solving mechanisms.

2.4. Conclusion

We do not reuse the existing methods for several reasons. The first one is that excepting MASK other methods do not consider a dynamic aspect of knowledge acquisition. Therefore they do not allow updating of the knowledge representation. The second one is due to our general purpose to develop a decision help system for diagnostic tasks in maintenance. This exceeds the simple knowledge representation and modelling and underlines process tasks and a chosen problem solving method. Further, engineering analysis methods and tools already exist in this domain and are used by experts. This necessitates their integration in the knowledge modelling process. The presented methods do not focus on knowledge acquisition from existing analysing methods and are not adapted to the knowledge exploitation. These tasks join tasks of artificial intelligence research domain where the purpose is to develop information systems supporting humans in their daily tasks and requiring knowledge implementation.

This is why we propose a mix method of knowledge capitalization based on the representation and reasoning model. In parallel, it corresponds to the dissociation of conceptual and operational model involved by the artificial intelligence and software engineering domains. The conceptual model is an intermediary model which facilitates dialogue between the expert and the knowledge engineer. This model is further made operational to create a decision support system.

3. Proposed mix methodology of knowledge capitalization

The underlying principle of the proposed methodology is the knowledge capitalization cycle presented by Grundstein. It contains four principal steps: detect,

preserve, capitalize and actualize (cf. fig. 2). Applied on the maintenance domain, each step is realised by different techniques from the knowledge engineering and the artificial intelligence. The detection phase is based on existing analytical methods and tools issued from the maintenance engineering. These tools detect the "right" knowledge for diagnosis task and requirements and optimize the knowledge acquisition and modelling process. The knowledge representation model based on UML diagrams supports the preservation step. It permits to represent and model domain knowledge and guarantees its repository. This representation model is compiled with the knowledge reasoning model which supports the capitalization and actualization step of capitalization cycle. The reasoning model is based on the artificial intelligence method – the case-based reasoning. This reasoning mechanism is associated to the representation model in order to ease the knowledge use in the knowledge based system. Both models complete each other and their integration permits to represent domain knowledge as well as "support knowledge" necessary for information system development. On the other side, we can exploit the dynamic aspect of the case-based reasoning cycle for continual dynamic knowledge acquisition and actualization from experts and technical documents. The object oriented representation was chosen as the description language.

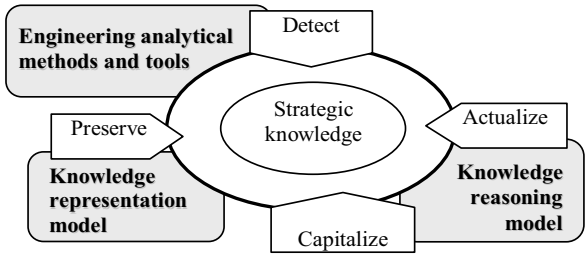

Figure 2. *The knowledge capitalization cycle in maintenance*

3.1. Knowledge representation model

To make a domain model means to define objects, to describe them and to structure the gathered descriptions. This implies to precise the description language and the organisation (acquisition) system. During the realisation of application, the informal description of objects should be transformed in the formal one (Bachimont *et al*., 2002). Object oriented conception method deals well with user needs, offers simple conception architecture and help to develop easily maintainable systems. Such methods create a set of models and diagrams in according to domain identification and its functionality specification. These diagrams provide multiple perspectives of the system under analysis or development for users as well as for software developers. Aamodt (2001) shows that the object oriented knowledge

representation is well adapted to handling composite entities and provides two modelling layers. The symbol level modelling deals with the developer's point of view and points out data structures and programming languages. The knowledge level modelling deals with the domain expert's point of view and represents the domain knowledge.

Bézivin (2000) stresses that meta models used by OMG (Object Management Group) correspond well to the creation of our knowledge representation model. The Unified Modelling Language (UML) was chosen as the modelling language to present our model. The UML specification and notation approved by OMG in the aim of reducing the confusion degree within the industry surrounding modelling languages (Rumbaugh et al., 1999). OMG (2003) defines the UML as a graphical language for visualizing, specifying, constructing, and documenting the artefacts of software systems, as well as for business modelling and other non-software systems, in general distributed object systems. It gives a complete representation of knowledge by its various diagrams. The class diagram in particular enables the representation of domain knowledge model. We use the UML diagrams to build the knowledge representation model proposed in this paper.

3.2. Knowledge reasoning model (case-based reasoning)

The most of engineering methods employs their own problem solving method or reasoning technique. We propose to use the case-based reasoning (CBR) using similar past cases to solve a new problem. This reasoning is close to the human one and so faithful for human users of knowledge based system. It provides a knowledge evolution which involves a difficulty for many methods but the case-based reasoning solves this problem in its evolution cycle. The case-based reasoning implements a case base made up of cases containing solutions that were used to solve old problems. This necessitates knowledge representation techniques for the case representation building and the acquisition of several cases in the case base. The cases must be well represented in order to easily retrieve case similar to the problem to be solved. Mille (1999) proposed the case-based reasoning cycle consisting of 5 phases as shown in fig. 3. In the elaboration phase the new problem must be described according to the formalised case representation. In the retrieval phase the most similar case or cases from case base to the new problem are retrieved. In the adaptation phase, the solution of retrieved case is reused, taking into account the differences between the retrieved and the new case, in order to solve this new problem. In the revision phase, the proposed solution is tested or evaluated in the real world. During the retain phase, if the new case is different from past cases, it is stored in the case base. The cases are stored and organized according to well defined criterions making possible to find them effectively. Moreover, the

acquisition of the new case makes evolve knowledge. The CBR feasibility for the decision help in industrial supervision was shown in the study of the decision making process (Mille *et al.*, 1999).

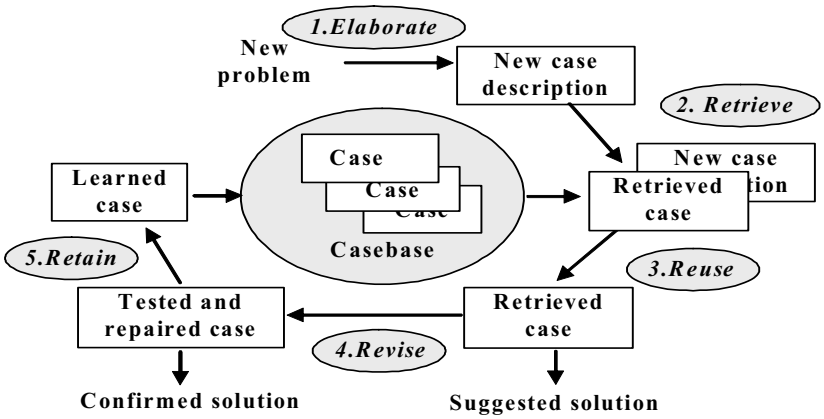

Figure 3. *The CBR cycle (Mille, 1999)*

The current research in this domain is focused on the detailed knowledge representation and in this view, Althoff (2001) considers the CBR as the technology of knowledge-based system implementation. Knowledge is stored in form of cases in a case base which can thus be regarded as a knowledge base. The case vocabulary joins the domain knowledge model and we can use it for the case base. Hence we use the ontology techniques for creation of the case representation. The advantages of such a match are studied in (Bergmann *et al*., 2003). We worked out the structural case-based reasoning system to create an interactive system of knowledge capitalization in e-maintenance.

3.3. Mix methodology of knowledge capitalization

The proposed mix methodology of knowledge capitalization is oriented toward the development of decision support system for diagnostic tasks in maintenance. It is based on the knowledge model integrating the representation and the reasoning model. This integration permits, on the one side, to guide the domain expertise and to propose the aid for the knowledge modelling and representation in maintenance. On the other side, this represented knowledge can be manipulated by the selected problem solving method in the decision support system. The conformity of represented knowledge and its use in the decision support system is so guaranteed.

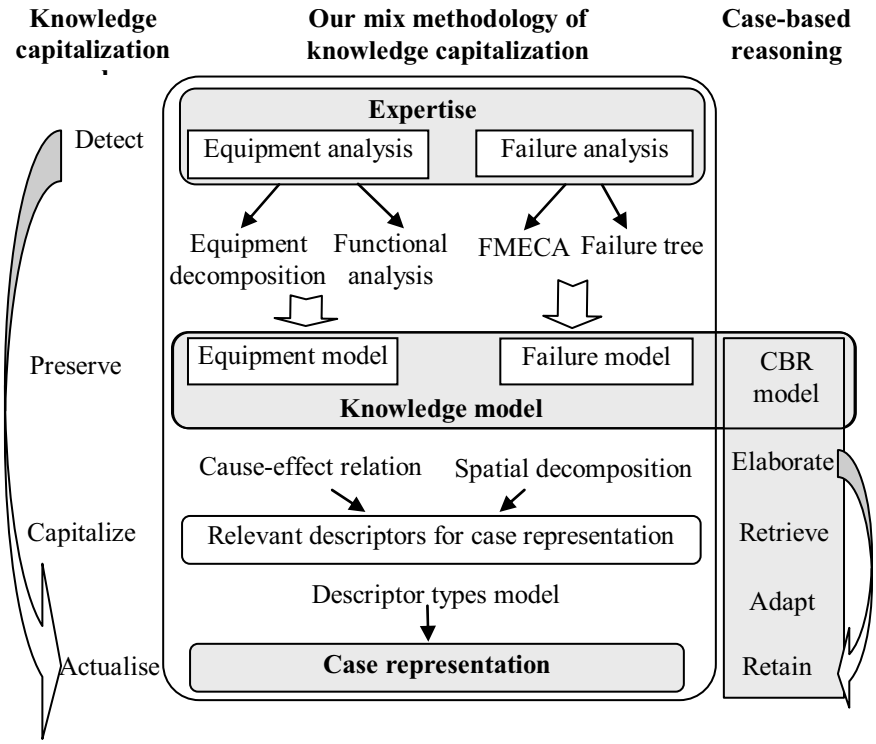

Figure 4. Principle of mix methodology

The methodology described in fig. 4 cures the lack of expertise in the diagnostic domain by using engineering safety tools like FMECA[3], functional analysis or failure history. These tools simplify the knowledge management by reducing its complexity. Two complementary types of analysis are identified. The first one concerns equipments and employs techniques of equipment decomposition and functional analysis. It determines a hierarchical equipment model. The second one concerns failures and employs the reliability concepts such as FMECA and failure history. The expertise issued from these analytical tools is enriched by information from intervention reports and then modelled and represented by knowledge engineering techniques. In more details, the knowledge models are used to represent the domain knowledge as well as the problem solving concepts which manipulates this domain knowledge. Two corresponding models, i.e. equipment and failure model, make part of the knowledge representation model created by UML diagrams. The third model participating in the general one proposes CBR concepts that handle

[3] Failure Modes and Effects Criticality Analysis

the domain concepts. This allows the system developer to take into account important aspects of CBR cycle steps in the case representation. These models can be associated to the terminology of the most known knowledge engineering method CommonKADS (Wielinga et al. 1993). This means that the equipment model represents the domain knowledge, the failure model represents the inference knowledge and the CBR model corresponds to the task knowledge.

The knowledge items in the UML class diagram are connected by relations as "composed-of, is, has-for-instance" and other associations. The relations between concepts are used to create supplementary models such as the cause-effect relation model (based on associations) and the spatial decomposition model (based on "composed-of" relations). They are necessary to identify and describe possible relevant descriptors for the case representation. In order to take into account common equipments characteristics and to take into account similarity and adapting knowledge, the descriptors are generalized and formalized in the descriptor types' model (based on generalization and inheritance). These general models are proposed to create the case representation in different applications while other CBR applications propose the case model already fixed.

UML class diagram

The knowledge model consisting of domain and reasoning model is illustrated on fig.5 in the form of UML class diagram. The domain part is on the left side of the figure and includes equipment and failure model. The first one describes equipment in two different aspects: its type and its hierarchical decomposition. The equipment could be of different types such as mechanical, hydraulic, electrical etc. which is important for the equipment repair. The equipment decomposition is in the form of tree structure which consists of class zone, field and subfield. This is important for localization of failure equipment. The failure model is issued from equipment's failure analysis made by reliability tools such as FMECA, failure tree in order to assure equipment maintenance. This model describes functions assured by equipment in the whole system. These functions are devised on main, second and constraint. The equipment operates in two modes: functional and failure mode. If one of equipment function is failed its mode is failure. Different events that arrive on equipment are distinguished such as repair action, failure or alarm. Moreover, the monitoring principle is presented based on sensor and its variable collection consisting composed of different variables. If a variable gets over some limit it raises an alarm.

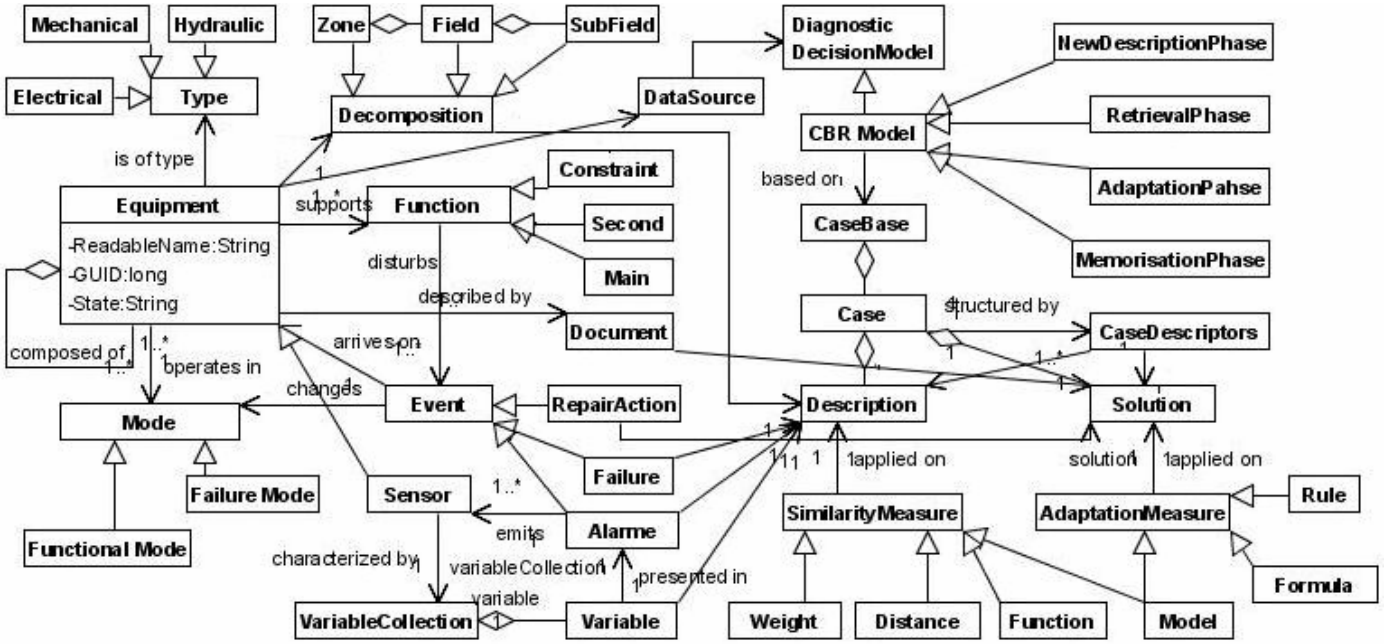

Figure 5. *General maintenance knowledge model of diagnosis and repair help system*

The decision help system is describes by reasoning model illustrated on the ride side of figure 5. The diagnostic decision model is based on the class data source linked with the concerned equipment. CBR model consists of four main steps: new description, retrieval, adaptation and memorisation. CBR model stores the cases on the case base. The case is structured by case descriptors and composed of two parts: description and solution. The description is associated with equipment decomposition, failure, alarm and its variable that characterizes the equipment status. The description is linked to the similarity measure such as distance, weight or model to find similar cases in the case base. The diagnosis consists in describing the symptom by variables which allows identification of failure origin (equipment decomposition). This determines the repair action and suitable technical Document for the operation provided in solution of the case. The class of adaptation measure is introduced to adapt the solutions of old cases to the new ones. Each class of this diagram can be instantiated in order to create objects and consequently cases as. A case is represented by an instantiation of the class diagram.

4. Demonstration platform

The objective of our work is to build an intelligent application for diagnosis and repair in the context of maintenance services. This decision tool is developed within the framework of the distributed e-maintenance platform shown in fig. 6. The design of generic software architecture for web-based e-maintenance centers was the goal of the European project Proteus. 16 partners from France, Germany and Belgium participated in this project. The platform brings a major asset to maintenance interventions and maintenance services in general by enabling expertise via Internet directly to the user site. The web portal in the core allows different users – maintenance actors – to access to decision support for their activities. The platform integrates a number of systems and knowledge bases like CMMS (Computerized Maintenance Management System), SCADA (Supervisory Control and Data Acquisition), maintenance data bases and e-documentation and finally applications and tools for decision support.

The platform is used as a support for information diffusion. Web services were developed in order to link together knowledge acquisition, information systems and decision help tools. The description of these interfaces is not studied in this paper, because this would bring nothing to the methodology of knowledge capitalization. The access to the diagnostic service is opened to all maintenance actors. On the other hand, modification of already stored knowledge for its up to date handing-over is authorized only to the designated experts. Our objective is to develop a decision help system for equipment diagnosis and repair. The concept of a decision support system is based on domain expertise which needs the cognitive process and knowledge modelling. Knowledge management techniques as well as artificial

intelligence method were studied and the mix knowledge capitalization methodology is proposed in this paper in order to develop such a decision help system.

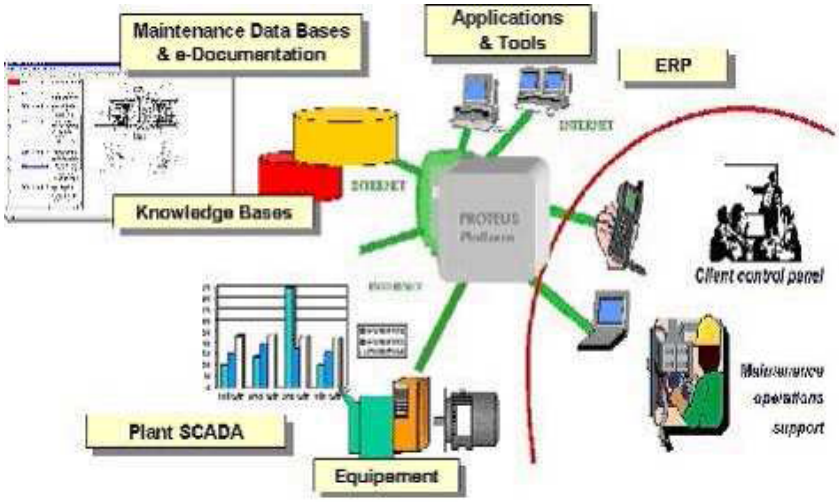

Figure 6. *Architecture of Proteus platform (http://www.proteus-iteaproject.com)*

This methodology is demonstrated on the industrial platform SORMEL shown in fig. 7. It represents a flexible production system composed of five robotized working stations. They are served by the pallet transfer system organized in double ring (internal and external). The pallets are conveyed on the interior ring which allows the transit between the various stations. When the pallet should be handled by a robot in a working station (information read on the pallet label), the pallet is deviated on the external ring where the concerned working station is. The working station is situated on the external ring and contains pneumatic, electric and inductive sensors. The working station is described in more details in fig. 8. Each station is equipped with pneumatic actuators (pushers, pullers and indexers) and electric actuators (stopper) as well as a certain number of inductive sensors (proximity sensors). An inductive read/write module allows to identify and locate each pallet and to provide information relative to the required operation in a concrete station. The displacement of the pallets is ensured by friction on belts which are involved by electric motors. Each pallet has a magnetic label that is used like embarked memory. This memory can be read in each working station thanks to magnetic read/write modules (Balogh) and allows memorizing of the product assembly sequence. These labels thus enable to determine the pallet path through the system.

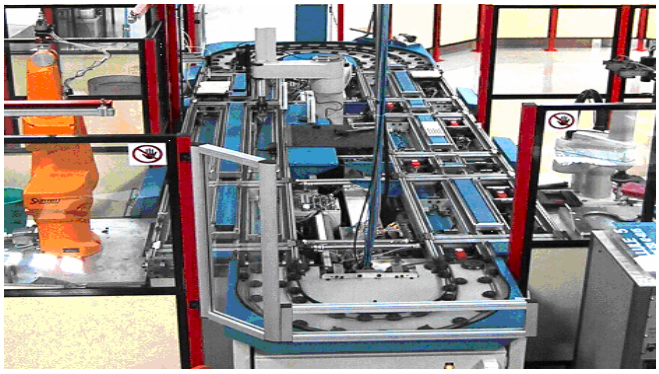

Figure 7. *Transfer system SORMEL*

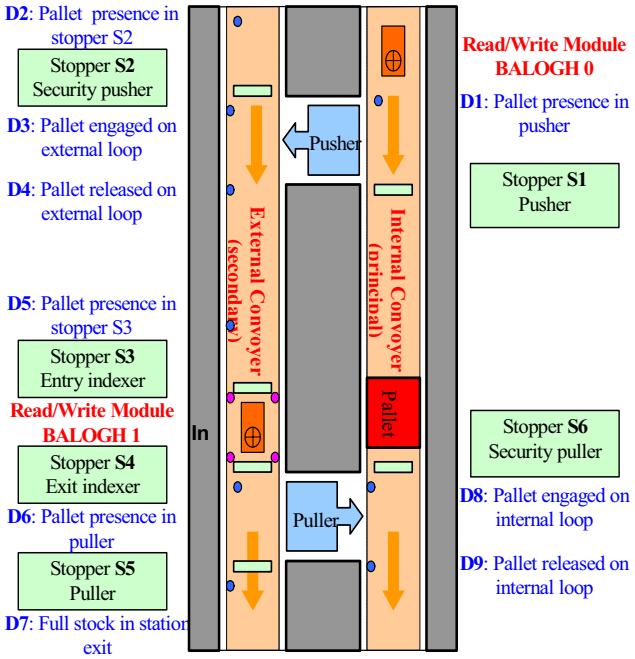

Figure 8. *Station of the transfer system SORMEL*

4.1. Implementation of proposed methodology on demonstration platform

The domain knowledge model of the pallet transfer system SORMEL is created with the aid of engineering analytical methods and tools. Its short view concerning the relevant concepts for our decision support system is illustrated in fig. 9. It

contains the equipment model characterized by the equipment functional and component decomposition and the failure model characterized by the functional mode and equipment condition. This model leads to identification of eventual case descriptors corresponding to the system equipments.

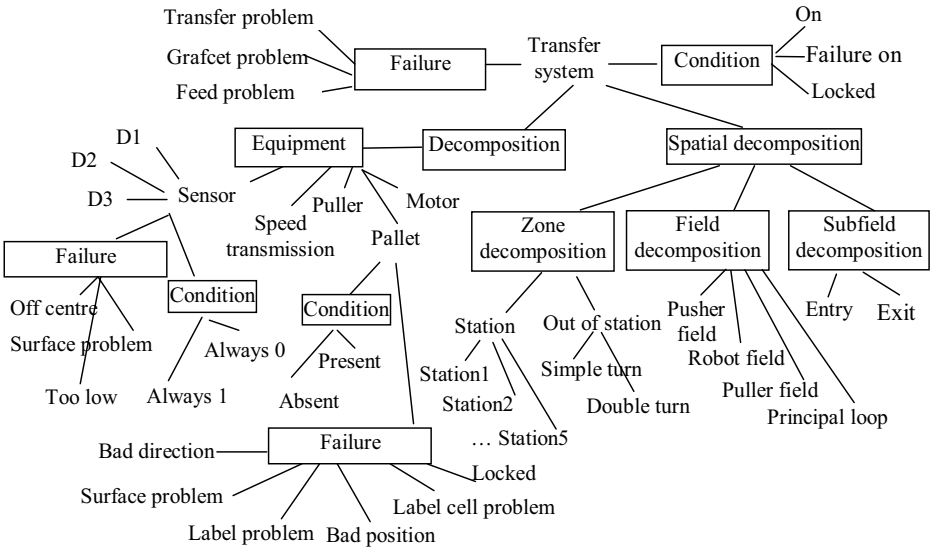

Figure 9. *The transfer system SORMEL knowledge model*

The case representation and case acquisition are essential components in CBR applications development. The case acquisition phase proves to be a significant task of knowledge engineering. A case is a description of problem solving episode. In general, it is the association of some problem and its solution. There is a number of different theories on the case representation but the most often used one is structured in a list of descriptors that take the form of complex objects. The case representation requires to list the various system components and to characterize them. In our case, it is based on the knowledge representation model and on the knowledge reasoning model of CBR system presented in fig. 5. The CBR system manipulates the knowledge representation model and uses it in order to create cases in the case base.

The case development consists in facilitating the problem description in order to facilitate the search of a case whose solution will be most easily adaptable. The general method lies on completion or filtration of problem description which is based on domain knowledge. So that the eventual incomplete description is deduced and the weighting of descriptors is done according to identified dependencies between the new problem's descriptors and the searched solution's ones. The case descriptors are issued from components of different nature such as sensors,

controllers and control (command) units. To characterize the case descriptors (attributes) the equipment functional mode and conditions are put in the context (cause-effect relation model). The conditions are associated as descriptors values. The failure detection rules are applied: IF ((pallet: present) AND (sensor: always 0)) THEN (sensor problem OR pallet: bad direction) to detect the failure component. The failure component with its functional mode represents the context evaluation with the repair action as the final solution. An example of such a characterization is introduced in fig. 10 for sensor.

Component	**Symbol**	**Context**	**Component condition**	**Component functional mode : *action***
Sensor	D1,D2 … D9	Pallet : Present	Always 0	Sensor problem – too low : put up sensor
				Sensor problem – off centre: *push sensor*
				Sensor problem – defected : *change sensor*
		Pallet: Absent	Always 1	Sensor problem - surface problem (metalic element, iron powder): *clean sensor*

Figure 10. The equipment characterization (cause-effect model)

Zone	Field	Subfield	Context localisation		Pertinent descriptors
			Precedent field	Next field	
Station	Secondary loop	Entry	Exit secondary loop	Exit pusher	D2, S2, ExtBelt
		Exterior conveyor	Exit pusher	Entry post	D3, D4, ExtBelt
		Exit	Entry puller	Entry second loop	S5, D7, ExtBelt
	Pusher loop	Entry	Entry principal loop	Pusher	D1, S1, BLGH0, Pusher
		Pusher	Entry pusher	Exit pusher	Pusher
		Exit	Entry secondary loop, Pusher	Entry external conveyor	Pusher, D3, D2, S2
	Post field	Entry	External conveyor	Indexer	D5, S3, ExtBelt
		Post	Indexer	Entry puller	Balogh1, S4, Indexer, Robot

Figure 11. *The knowledge hierarchy for the case base (spatial decomposition)*

In order to create the case as a “diagnostic situation” the spatial decomposition model is used (fig.11). This decomposition allows identification of relevant and pertinent descriptors for every diagnostic situation. At the same time, this model represents the information hierarchy established for the future case base. This will facilitate the retrieval of similar case.

4.2. Descriptor type model

Finally, a descriptor type model is established issued from the knowledge representation model. It serves for retrieve and reuse tasks in the case-based reasoning cycle. In the model, descriptors are classified according to their functionality. The square represents the generalized concepts and the oval represents the instances of the transfer system SORMEL. As shows the contoured part of fig. 12, the general class *Magnetic sensor* is composed of two subclasses *Presence sensor* and *Balogh.* Then the instances corresponding to the real system components make part of both classes.

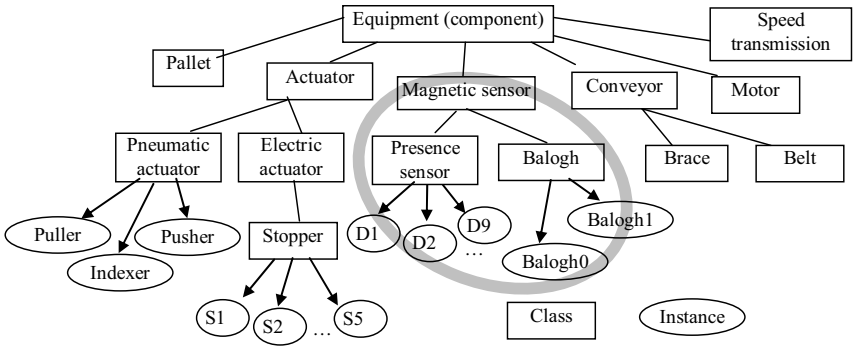

Figure 12. *The descriptor type model*

The retrieve phase of the case-based system is based on the similarity notion. The similarity between two cases is calculated on the case descriptors. Similarity measures are adapted to the object oriented case representation. The similarities between the same descriptors are calculated by simple comparison of their values. For two different descriptors the similarity is calculated while going up to the first common concept in the descriptor type model. The similarity grows as we descend in the hierarchy and is given by comparing the attributes common on this level.

In the reuse phase, the descriptors hierarchy is used in order to generalize the cases in the case base. For example, one descriptor relates to the pusher. The new

descriptor corresponds to the puller and in the descriptors hierarchy one can see that the two descriptors belong to the class actuator. The general class actuator leads to the solution for the new problem. To every general descriptor class we associate repair operations, necessary human and material resources, appropriate technical and other documentation and the time duration of the intervention as the solution attributes. Further, in reuse phase of the CBR cycle this hierarchy is used to replace given component of a new case by another one from the same family (the same generic class) already existing in the case base. We reuse solution attributes of the general class to the new component. The adaptation strategy is introduced based on adaptation operators. The adaptation operator is applied to a characteristic attribute of a case solution. This hierarchy with general classes represented in the case base by general cases limits the size of this case base and so the time and effectiveness of the case retrieval. Moreover the transfer system consists of 5 identical stations; it is thus possible to build generic classes to adapt the solutions for each particular station.

Finally, the case representation is retained as shows fig. 13. A case consists of four main parts: context, attribute-value list, its evaluation and final solution (repair action). The case is elaborated from the symptom description characterizing the problem nature. During the acquisition of a new problem description one specifies context (symptom and its localisation), components – descriptors of this context and their values. The problem solution summarizes components identified in the context with their failure modes. This leads to the identification of the failed one and to the repair action associated to the proposal of the operator skills for this intervention, required spare parts, required tools and suitable technical documentation.

Context		Attributes–values				Evaluation		Solution
Symptom	Localisation					Failure component : Functional mode	General class	
Transfer problem	Station,Actuator.Entry	D6:1	B1:1	S5:0	P:0	Pusher : Cylinder out of order	Actuator	Change cylinder

Figure 13. *The case representation*

4.3. Architecture of decision support system

The architecture of the decision support system for diagnosis and repair within the e-maintenance platform is illustrated in fig. 14. The web portal of the CBR tool

for users is connected by Proteus with the CBR algorithm module and the web services (developed under Java and Python). This module is connected with the case base and the description procedures developed in Protégé. The description procedures formalise dynamically suitable questions in order to work out a new problem description. These questions are answered by operator, or by other modules integrated in the platform. The CBR module uses the Belfort site ontology and the generic maintenance ontology.

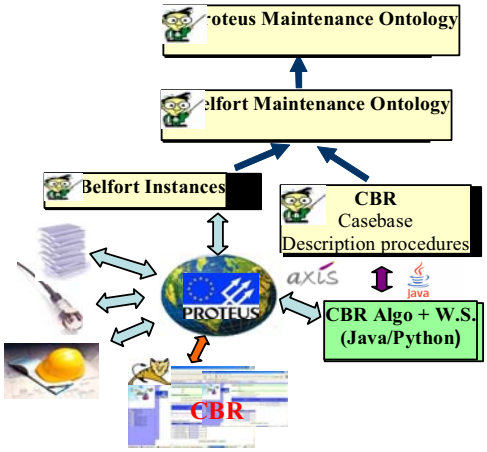

Figure 14. *Architecture of decision support system*

Once the maintenance intervention request with the basic failure information has been received, maintenance operator asks for help from the diagnosis and repair decision support system. The system works out a new problem description by asking the operator and by automatic collection of information and values coming from sensors from integrated systems in the platform, namely monitoring system SCADA. A new problem is matched against cases in the case base and the most similar cases are retrieved. The solution suggested by these cases is adapted for the new situation and tested for success.

Conclusion

The goal of this paper was to propose a methodology for capitalisation knowledge in the maintenance domain. This methodology suggests development of

a decision support system for industrial equipments diagnosis and repair. The concept of such a system is based on domain expertise that requires cognitive modelling of the process and knowledge used by operators during their activities. Thus the review of different knowledge management methods and tools was introduced. Their synthesis revealed that the existing methods are not adapted to our purpose. The represented knowledge is not adapted for reasoning and manipulation in the knowledge based system. The knowledge acquisition is not dynamic and does not take into account the existing engineering analytical tools.

This is why we introduced a mix method adopting the knowledge capitalisation cycle as the underlying principal of the proposed method. It is based on the knowledge model integrating the representation and the reasoning model. This integration permits to guide the domain expertise and to propose the aid for the knowledge modelling and representation in maintenance. This represented knowledge can be manipulated by the selected problem solving method in the decision support system. We proposed to compile the knowledge engineering techniques as representation and modeling method and the case-based reasoning as the problem solving method. Both methods complete each other and their integration permits to represent domain knowledge as well as knowledge necessary for information system development. On the other side, we can exploit the dynamic aspect of case-based reasoning cycle for continual dynamic knowledge acquisition and actualization from experts and technical documents. At the same time, the knowledge acquisition is based on existing engineering analytical methods.

For the implementation of decision support systems, it is necessary to cover the design process and to match requirements and needs for proposed systems. The equipment maintenance processes a huge volume data that is not always complete and requires up to date handing-over. The incremental decision help tool is to be envisaged. System supporting the process operation requires a case-based reasoning method that can represent system dynamics and fault propagation phenomena. Nevertheless, there are several problems in using CBR systems. Until today CBR systems are often isolated and they cannot cooperate with other systems. On the contrary, our decision support system is integrated in the e-maintenance platform. It communicates with other maintenance systems and applications of this platform through developed web services. On the other side, there is no standardisation in the creation of case and case vocabulary. The knowledge representation techniques aim at the systematic knowledge assets creation and storage based on the knowledge items characterization. So the CBR system is completed by using the knowledge representation techniques which are based on the formal description and standardisations but have no reasoning mechanism allowing the use of existing knowledge.

Our approach was demonstrated on the pallet transfer system Sormel as the industrial application. The decision support system for maintenance intervention management is designed as an interactive system. It deals with the expert knowledge in form of cases in the case base. These cases were created by using the proposed methodology. They connect in certain manner the domain knowledge concepts according to different models developed during the methodology implementation. Thus the case representation joins the domain representation model creation and the cases as knowledge items are reused and handled by the case-based reasoning mechanism. Thanks to the methodology models, the case representation takes into account different reasoning tasks of CBR such as retrieve and reuse. Thus the retrieval and adapting process are optimized. Actually, the case base contains about 40 cases. This allowed the case retrieval testing. Tests permitted to find the general cases and to replace the non generic ones. So the case base size is reduced and the case retrieval is faster. We generated randomly 15 cases corresponding to one tierce of case base and we obtained precision of 95 %. These tests will follow up.

References

Aamodt. A. (2001) Modeling the knowledge contents of CBR systems. Proc. Of the Workshop Program at the Fourth International Conference on Case-Based Reasoning, Vancouver.

Althoff. K.D. (2001) Case-Based Reasoning. In: S.K. Chang (Ed.). *Handbook on Software Engineering and Knowledge Management*, pp. 549-588.

Bachimont B., Isaac A., Troncy R. (2002) Semantic Commitment for Designing Ontologies: A Proposal. Proc. of the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW'2002, Springer Verlag, LNAI 2473, Spain, pp. 114-121.

Barthès. J.P. (1996) ISMICK and Knowledge Management. In Knowledge Management: Organization, Competence and Methodology, Proc. Of ISMICK'96, Rotterdam, Netherlands, pp. 9-13.

Bergmann. R. Schaaf. M. (2003) Structural case-based reasoning and ontology-based knowledge management: A perfect match? Journal of Universal Computer Science, 9(7), 608-626.

Bézivin. J. (2000) From the object programming to the ontology modeling « in french ». *Ingénierie de connaissances*, pp. 209-222.

Bourne. C. (1997) Categorization and formalization of industrial knowledge « in french ». In Fouet(Coord.), *Connaissances et Savoir-faire en entreprise*, Hermès, pp. 179-197.

Charlet. J. Bachimont. B. Bouaud. J. Zweigenbaum. P. (1996) Ontology and reutilisability: experiment and discussion « in french ». *Acquisition et ingénierie des connaissances : tendances actuelles*, Cépaduès, Toulouse.

Charlet. J. Zacklad. M. Kassel. G. Bourigault. D. (2000) Knowledge engineering: Recent evolutions and new challenges « in french ». Editions Eyrolles, Paris, France.

Corbel J.C. (1997) Experience feedback methodology : approach MEREX from Renault « in french *». Connaissances et savoir-faire en entreprise*, Hermès, pp. 93-110.

Duribreux-Cocquebert. M. Houriez. B. (2000) Industrial application of a mixed approach of knowledge modeling « in french ». *Ingénierie de connaissances*, pp. 25-42.

Ermine. J.L. (1996) Knowledge systems « in french ». Hermès, Paris.

Fouet J.M. (1997) Knowledge and know how in enterprise, integration and capitalization « in french », Hermès.

Grundstein. M. (1992) Knowledge Engineering Within the Company: An Approach to Constructing and Capitalizing the Knowledge Assets of the Company. In IAKE'92 Proceedings, Third Annual Symposium of the International Association of Knowledge Engineers, Washington DC.

Grundstein. M. (1994) To develop a system containing knowledge: an effort of co-operation to build jointly an unknown object « in french ». CP2I, Paris.

Holsapple. C.W. Joshi. K.D. (1999) Description and Analysis of Existing Knowledge Management Frameworks. In Proc. of 32nd Hawaii International Conference on System Sciences, IEEE.

Jarboe. K.P. (2001) Knowledge Management As an Economic Development Strategy, Reviews of Economic Development Literature and Practice n° 7.

Lai. H. Chu. T. (2000) Knowledge Management: A Review of Theoretical Frameworks and Industrial Cases. In Proc.of the 33rd Hawaii International Conference on System Sciences, IEEE.

Malvache P. et Prieur P. (1993) Mastering corporate experience with the Rex method. In proc. of ISMICK'93, Compiègne.

Matta. N. Ermine. J.L. Aubertin. G. Trivin J.Y. (2001) Knowledge capitalization with a knowledge engineering approach : the MASK method. In proceeding of IJCAI'2001 workshop on Knowledge Management and Organizational Memory.

Mille. A. (1999) Tutorial CBR: State of the art of case-based reasoning « in french ». Plateforme AFIA'99, Palaiseau.

Mille. A. Fuchs, B. Chiron B. (1999) Reasoning based on the experiment: a new paradigm in industrial supervision « in french ». *Revue d'intelligence artificielle*, 13:97-128.

MOKA (2000) Final synthesis, *Deliverable 4.3*, The MOKA consortium.

Motta. E. Rajan. T. Eisenstadt. M. (1990) Knowledge Acquisition as a Process of Model Refinement. *Knowledge Acquisition,* 2, pp. 21-49.

OMG (2003). Unified Modeling Language Specification. Version 1.5.

Rasovska. I. Chebel-Morello. B. Zerhouni. N. (2004) A conceptual model of maintenance process in unified modelling language. In Proceedings of the 11th IFAC Symposium on Information Control Problems in Manufacturing, Salvador-Bahia, Brasil.

Reynaud. Ch. Tort. F. (1997) Using Explicit Ontologies to Create Problem Solving Methods. International Journal of Human-Computer Studies - Special issue on Explicit Ontologies in KBS Development, Vol. 46, pp. 339-364.

Rosario. J.G. (1996) Much ado about knowledge capital, Business World, Philippines.

Rumbaugh. J. Jacobson. L. Booch. G. (1999) The unified modeling language reference manual. Reading, Addison Wesley.

Schreiber. A. Terpstra. A. Magn. P. Van Veelzen. M. (1994) Analysing and Implementing VT Using CommonKads. In Proceedings of the 8th Banf Knowledge Acquisition for Knowledge-Based Systems Workshop.

Simon. G. (1996) Knowledge Acquisition and Modeling for Corporate Memory: Lessons learnt from Experience. In Proc. Of *KAW'96,* Banff, Canada, pp. 41.1-18. Also in http://ksi.cpsc.ucalgary.ca/KAW/KAW96/KAW96ProcS.html

Van Craeynest. J.M. Ermine. J.L. Chagnot. Ch. (2000) Knowledge capitalization applied on the industrial transfer « in french ». *Ingénierie de connaissances*, p.465-480.

Van Heijst. G. Van der Spek. R. Kruizinga. E. (1996) Organizing Corporate Memories. In Proc. of *KAW'96*, Banff, Canada, pp. 42.1-17. Also in http://ksi.cpsc.ucalgary.ca/KAW/KAW96/KAW96ProcS.html

Vogel. C. (1988) Cognitive sciences « in french ». Collection „Sciences cognitives", Masson, Paris.

Wielinga B., Van De Velde W., Schreiber G. and Akkermans H. (1993) Towards a unification of knowledge modelling approaches. *Second Generation Experts Systems*, David J.M., Krivine J.P. et Simmons R. (Eds.), pp. 299-335, Springer Verlag, New-York, USA.