

Contribution à une méthodologie de capitalisation des connaissances basée sur le raisonnement à partir de cas : Application au diagnostic dans une plateforme d'e-maintenance.

Ivana Rasovska

▶ **To cite this version:**

Ivana Rasovska. Contribution à une méthodologie de capitalisation des connaissances basée sur le raisonnement à partir de cas : Application au diagnostic dans une plateforme d'e-maintenance.. Automatique / Robotique. Université de Franche-Comté, 2006. Français. NNT : . tel-00257893

HAL Id: tel-00257893
https://theses.hal.science/tel-00257893

Submitted on 20 Feb 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Année : **2006** N° ordre : 1156

THESE

présentée à

L'UFR des Sciences et Techniques
de l'Université de Franche-Comté

pour obtenir le

GRADE DE DOCTEUR DE L'UNIVERSITE DE FRANCHE-COMTE

en Automatique
(Ecole Doctorale Sciences Physiques pour l'Ingénieur et Microtechniques)

Contribution à une méthodologie de capitalisation des connaissances basée sur le raisonnement à partir de cas : Application au diagnostic dans une plateforme d'e-maintenance

par

Ivana Rasovska

Soutenue le 4 juillet 2006 devant la Commission d'examen :

Rapporteurs :	Daoud Ait-Kadi	Professeur, Université Laval
	Bernard Grabot	Professeur, ENIT
Examinateurs :	Pierre Borne	Professeur, Ecole Centrale de Lille
	Alain Mille	Professeur, Université de Lyon
Directeur de thèse :	Noureddine Zerhouni	Professeur, ENSMM
	Brigitte Chebel-Morello	Maître de conférences, Université de Franche-Comté

A mes chers parents avec tout mon amour et pour tout ce qu'ils m'ont appris.
A mon petit frère que j'adore.

Remerciements

Ce travail a été réalisé au Laboratoire d'Automatique de Besançon (LAB, UMR CNRS 6596) dirigé par Monsieur Alain Bourjault, professeur à l'ENSMM, que je remercie de m'avoir accueillie et de m'avoir offert d'excellentes conditions pour effectuer ce travail de recherche.

J'exprime toute ma reconnaissance à Monsieur Noureddine Zerhouni, professeur à l'ENSMM, de m'avoir accueillie au sein de l'équipe Maintenance et Sûreté de Fonctionnement et d'avoir dirigé ce travail, ainsi que pour son soutien, ses conseils et la confiance qu'il m'a accordé tout au long de ces années.

J'exprime toute ma gratitude à Madame Brigitte Chebel-Morello, maître de conférences à l'Université de Franche-Comté, pour son encadrement, sa compétence et sa disponibilité. Je la remercie non seulement de m'avoir orientée, conseillée et corrigée dans le travail, mais également pour sa sincère amitié, son soutien et pour ses qualités humaines.

Je remercie messieurs les membres du jury d'avoir accepté de participer au jury, ainsi que pour l'intérêt qu'ils ont porté à ce travail. J'adresse mes sincères remerciements aux :

- *professeur Daoud Ait-Kadi de l'Université Laval et professeur Bernard Grabot de l'ENIT, d'avoir rapporté ce travail de thèse ainsi que pour les corrections et les conseils qu'ils y ont apportés ;*
- *professeur Pierre Borne de l'Ecole Centrale de Lille et professeur Alain Mille de l'Université de Lyon, d'avoir examiné ce travail ;*
- *monsieur Denis Reboul et monsieur Pascal Bressy, d'avoir assisté en tant qu'invité à ma soutenance.*

Je tiens à remercier les membres de l'équipe de Cegelec Belfort, ainsi que les partenaires du projet Proteus pour leurs conseils et pour les nombreuses discussions que nous avons eues, et tout particulièrement Denis Reboul pour sa collaboration et son soutien tout au long du projet.

Je remercie très sincèrement l'ensemble du personnel du LAB pour son soutien, ses encouragements et pour l'excellente ambiance au laboratoire. Je tiens à remercier tout particulièrement Bruno pour son aide, pour de nombreuses heures de discussions et de démonstrations enrichissantes et pour son pessimisme constructif, ainsi que Sid-Ali pour son aide et ses encouragements lors de la préparation de soutenance. Merci à Gérald, Mimi et Johanna, ainsi que Sébastien et sa sœur pour la relecture du manuscrit.

Table des matières

Introduction générale 7

Chapitre 1 : Introduction à la maintenance industrielle 11

1. Introduction 12

2. Définitions de base de sûreté de fonctionnement et de maintenance 13

2.1. Types et niveaux de maintenance 14

2.2. Politiques et stratégies de maintenance 15

2.3. La fonction et le service de maintenance dans une entreprise 16

3. Analyse du système de maintenance 17

3.1. Le système d'information dédié au service de la maintenance 18

3.2. Différents systèmes informatiques en maintenance 24

3.3. Analyse du processus de maintenance 25

4. Différentes architectures des systèmes de maintenance 27

4.1. Définitions des différents systèmes 29

4.2. Maintenance 30

4.3. Télémaintenance 31

4.4. E-maintenance 33

4.5. S-maintenance 35

5. Conclusion 37

Chapitre 2 : Capitalisation des connaissances en maintenance 39

1. Introduction 40

2. Gestion des connaissances 41

2.1. Connaissances dans l'entreprise 42

2.2. Différents modèles de gestion 45

2.3. Tableau de comparaison 48

2.4. Cycle de capitalisation des connaissances 50

3. Capitalisation des connaissances et mémoire d'entreprise 51

3.1. Méthodes dédiées à la construction de la mémoire d'entreprise 52

3.2. Outils et méthodes de l'ingénierie des connaissances 55

3.3. Représentation des connaissances et ontologies 59

4. Formalismes de représentation des connaissances 60

5. Notre démarche de capitalisation des connaissances 62

5.1. Mémoire d'entreprise 63

5.2. Méthode de représentation 65

5.3. Méthode de raisonnement 71

6. Conclusion 72

Chapitre 3 : Raisonnement à partir de cas en diagnostic industriel 73

1. Introduction 74

2. Approche de raisonnement à partir de cas 75

2.1. Historique 76

2.2. Principes fondamentaux 77

3. Méthodes utilisées dans les phases du cycle de RàPC 81

3.1. Conception du système et phase d'élaboration 81

3.2. Remémoration 84

3.3. Adaptation 86

3.4. Révision 87

3.5. Mémorisation 88

4. Domaines d'applications 89

4.1. Typologie d'applications 90

4.2. Notre domaine de diagnostic 92

5. Etude comparative des différents systèmes de diagnostic existants 94

5.1. Systèmes développés à l'aide d'un système de développement 94

5.2. Systèmes développés dans un système de représentation des connaissances 97

5.3. JColibri 101

5.4. Tableaux récapitulatifs 103

6. Nos choix et notre démarche 105

7. Conclusion 107

Chapitre 4 : Développement et intégration d'un système d'aide au diagnostic et à la réparation dans une plateforme d'e-maintenance 109

1. Introduction 110

2. Cadre d'application 110

2.1. Architecture de la plateforme Proteus 112

2.2. Intégration d'un outil sur la plateforme 116

2.3. Plateforme de démonstration 119

3. Développement du système d'aide au diagnostic et à la réparation 122

3.1. Méthodologie proposée pour élaboration du système d'aide au diagnostic 123

3.2. Représentation et acquisition d'un cas 127

3.3. Remémoration des cas similaires 134

3.4. Adaptation 138

3.5. Révision et mémorisation 139

4. Conclusion 140

Conclusion générale 141

Annexe A : Politiques de maintenance 147

Annexe B : Modélisation UML 151

Annexe C : Web sémantique et Web service 155

Annexe D : Implémentation de l'outil sur la plateforme 157

Références bibliographiques 169

Introduction générale

Problématique

Les entreprises d'aujourd'hui doivent répondre aux demandes de plus en plus exigeantes en termes de qualité et quantité de produits et de services, de réactivité, de réduction des coûts. Afin de satisfaire ces demandes, l'entreprise doit disposer d'un système de production fiable, donc bien entretenu par un système de maintenance efficace et peu couteux. Un service de maintenance performant et bien organisé contribue à la bonne « santé » du système de production, il permet de prolonger la vie des équipements industriels et participe ainsi à la meilleure performance globale de toute l'entreprise. Ce besoin de maintenance concerne tout type d'entreprise, d'industrie ou de prestataire de services.

Avec les évolutions technologiques, informatiques et organisationnelles, les équipements industriels sont de plus en plus complexes et automatisés. Par conséquent, l'environnement évolue et induit une dynamique sur les processus industriels. De nombreuses dimensions comme la dynamique des processus, la mondialisation des marchés, les nouveaux modes organisationnels des entreprises autrefois figées et stables fragilisent les informations manipulées. Afin de les fiabiliser, celles-ci doivent être mises dans leur contexte d'évolution et devenir ainsi des connaissances. Les connaissances sont à la base des systèmes d'aide à la décision ou de déploiement de compétences stratégiques pour les métiers dans les entreprises, en l'occurrence ceux concernant la maintenance.

Les entreprises d'aujourd'hui cherchent des stratégies d'amélioration de leur système de maintenance. Il existe actuellement deux principaux types de stratégies d'évolution des systèmes de maintenance mises en œuvre :

- Une stratégie d'internalisation de la maintenance qui vise à renforcer la fonction de maintenance dans l'entreprise au niveau des ressources internes.
- Une stratégie d'externalisation de la fonction de maintenance, où l'entreprise se recentre sur son corps de métier et soustraite des ressources externes.

La tendance actuelle est d'externaliser, en partie, la fonction de maintenance. Grâce aux nouvelles technologies de l'information et de la communication, à l'émergence de la technologie Web et du réseau Internet, la réalisation des services de maintenance et des contrôles peut être

effectuée automatiquement, à distance et à l'aide de différents systèmes informatiques implantés au sein des entreprises. D'où l'émergence du concept de services proposés au travers d'architectures de maintenance, qui peuvent aller des systèmes autonomes vers des systèmes intégrés où la coopération et la collaboration sont vitales pour tout fonctionnement. C'est l'idée d'e-maintenance qui rejoint ainsi le concept de l'entreprise étendue.

La plateforme d'e-maintenance implémente l'architecture actuelle la plus performante permettant d'échanger et partager l'information entre plusieurs systèmes de maintenance. Une des caractéristiques de cette plateforme est d'être générique et de fournir les méthodes et les interfaces applicatives nécessaires à l'intégration de différents composants proposés comme services génériques appelés des services Web. Notre problématique se situe dans ce contexte d'externalisation, et notre objectif est de fournir un outil d'aide à la décision en maintenance, sous forme de service, dans une plateforme d'e-maintenance.

Cadre de travail

Nos travaux de thèse se placent dans le cadre d'un projet Européen nommé PROTEUS, terminé en février 2005. Ce projet a eu comme objectif le développement et la mise au point d'une plateforme distribuée et coopérative d'e-maintenance fournissant un ensemble d'outils d'aide associés aux différentes activités de maintenance, à différents acteurs allant du réparateur d'équipement au gestionnaire de contrats de maintenance. L'architecture d'une telle plateforme a permis d'augmenter l'efficacité du processus de maintenance tout en apportant l'expertise via l'Internet directement vers un acteur du système de maintenance. La plateforme présente deux caractéristiques importantes :

- la facilité d'intégration de nombreux systèmes informatiques (d'information) en maintenance,
- la mise en relation de tout acteur concourant à la maintenance des équipements donnés dans un parc machines donné.

Dans ce contexte, l'objectif de cette thèse est de fournir, parmi un ensemble de services de cette plateforme d'e-maintenance basée sur la technologie Web, un service spécifique d'aide au diagnostic et à la réparation. Une partie des travaux actuels dans la communauté web s'orientent vers le web sémantique. Dans cette mouvance, nous proposons d'effectuer une démarche de capitalisation des connaissances pour l'élaboration d'un service de maintenance, destiné à une plateforme d'e-maintenance. Notre objectif est double :

- Premièrement, l'élaboration d'une ontologie liée à l'équipement à maintenir et commune pour la plateforme permettra de garantir l'interopérabilité entre les systèmes et les applications intégrés.

- Deuxièmement, la démarche de capitalisation des connaissances permettra de formaliser et de préserver les connaissances et les expériences des employés d'entreprise. En effet, la fluctuation du personnel et le départ des experts entraînent une perte de savoir et savoir faire, que les entreprises cherchent à pallier par une démarche de gestion des connaissances.

Afin d'élaborer un service d'aide à la maintenance industrielle sur une plateforme de maintenance, nous suivons une démarche qui consiste :

- à analyser le processus de maintenance afin d'identifier les services possibles d'aide et les différentes architectures sur lesquelles ces services sont proposés ;
- à étudier plus spécifiquement un des services d'aide, service concernant le diagnostic et la réparation d'un équipement ;
- à concevoir un outil d'aide à la décision s'appuyant sur l'expertise capitalisée en maintenance, à l'aide d'une mémoire d'entreprise qui permet de préserver le savoir et le savoir faire des acteurs humains. La démarche de capitalisation des connaissances employée s'articule autour d'un cycle de raisonnement à partir de cas (RàPC) guidé par les connaissances ;
- à implanter cet outil sur la plateforme d'e-maintenance mise au point dans le cadre d'un projet Européen Proteus.

Organisation du mémoire

Au premier chapitre, nous définissons ce que sont une fonction et un service de maintenance pour pouvoir proposer ce service ou cet ensemble de services au travers d'une plateforme d'e-maintenance. Cette définition passe par l'analyse du processus général de maintenance, qui nous permet de déterminer quatre niveaux d'applications imbriqués les uns dans les autres. A chaque niveau est associé un ensemble d'outils d'aide à la décision, qui peut être agencé en modules intelligents concourant à la maintenance d'équipements et à la gestion des stratégies de maintenance. Dans un deuxième temps, nous proposons une classification des différentes architectures de maintenance existantes pour en déduire une architecture comme système support aux services de maintenance. Nous mettons en évidence un type d'architecture de maintenance sémantique : la s-maintenance qui prend appui sur la technologie émergeante du Web sémantique.

Au deuxième chapitre, nous présentons un état de l'art des différentes approches de gestion des connaissances en entreprise ainsi que les différentes méthodes et outils de représentation et de modélisation des connaissances. Cela nous permet de suivre une démarche de capitalisation des connaissances et d'élaboration d'une mémoire d'entreprise pour les systèmes d'aide à la décision en maintenance. La mémoire d'entreprise permet de préserver le savoir et le savoir faire des employés de l'entreprise et ensuite de l'exploiter par un système de capitalisation des

connaissances. Nous employons les méthodes de représentation et de modélisation des connaissances pour structurer et modéliser les concepts identifiés dans l'analyse du système de maintenance au premier chapitre. Ces concepts sont les connaissances essentielles pour la mémoire d'entreprise que nous exploitons par une démarche de capitalisation. L'expertise capitalisée est donc à la base de la conception de notre outil d'aide à la décision. Sa conception passe par la modélisation des connaissances qui se décline par un modèle de représentation associé à un modèle de résolution de problèmes. Ce système emploie la méthodologie du raisonnement à partir de cas.

Au troisième chapitre, nous introduisons des principes fondamentaux du raisonnement à partir de cas et un état de l'art des méthodes utilisées dans son cycle. Nous analysons quelques systèmes de raisonnement à partir de cas en diagnostic technique. Nous faisons leur comparaison à partir de différentes phases de raisonnement. Cela nous permet de créer notre propre système d'aide au diagnostic et à la réparation où nous associons la démarche de raisonnement à partir de cas à une démarche de capitalisation.

Au quatrième chapitre, nous présentons l'architecture de la plateforme d'e-maintenance développée dans Proteus. Nous décrivons le principe d'intégration d'un outil spécifique présenté comme un service Web. Nous développons un système d'aide au diagnostic et à la réparation basé sur le raisonnement à partir de cas. Un soin particulier est apporté à l'élaboration de cas et de la base de cas par des techniques empruntées à la sûreté de fonctionnement. Nous faisons un choix des mesures de similarité et d'algorithme de recherche des cas similaires orienté vers l'adaptabilité des cas similaires. Nous montrons la faisabilité de notre démarche sur une plateforme de démonstration assurant la maintenance d'un système de transfert de palettes Sormel.

Chapitre 1

Introduction à la maintenance industrielle

1. Introduction 12

2. Définitions de base de sûreté de fonctionnement et de maintenance 13

2.1. Types et niveaux de maintenance 14

2.2. Politiques et stratégies de maintenance 15

2.3. La fonction et le service de maintenance dans une entreprise 16

3. Analyse du système de maintenance 17

3.1. Le système d'information dédié au service de la maintenance 18

3.2. Différents systèmes informatiques en maintenance 24

3.3. Analyse du processus de maintenance 25

4. Différentes architectures des systèmes de maintenance 27

4.1. Définitions des différents systèmes 29

4.2. Maintenance 30

4.3. Télémaintenance 31

4.4. E-maintenance 33

4.5. S-maintenance 35

5. Conclusion 37

1. Introduction

Les enjeux socio-économiques croissants sont liés aux impératifs de sécurité des hommes et des matériels, aux exigences de protection de l'environnement et de réduction des nuisances, et, naturellement, aux gains de productivité sur des systèmes de production. Cela place les problèmes de sûreté de fonctionnement au centre des préoccupations liées à l'optimisation des processus industriels [Journal, 2001]. Cette notion de sûreté de fonctionnement est sensible à la demande du consommateur, en l'occurrence avoir un produit de haute qualité, efficace et sûr de fonctionnement dans le service pour lequel il a été fabriqué [Küssel et al., 2000].

La sûreté de fonctionnement est définie par [Laprie et al., 1995] comme « la propriété d'un système qui permet de placer une confiance justifiée dans le service qu'il délivre ». Selon [Francastel, 2001] la sûreté de fonctionnement est « un ensemble d'aptitudes nécessaires à un bien à être disponible aux différents moments de son cycle de vie, en offrant les performances requises, soit : fiabilité (ou probabilité pour que le bien accomplisse sa fonction), maintenabilité (ou aptitude à être maintenu), sécurité pour les biens eux-mêmes, le personnel et leur environnement) ».

La maintenance apporte une contribution essentielle à la sûreté de fonctionnement d'un bien. Au cours de ces dernières décennies, la fonction maintenance est devenue « la réponse » au besoin de maîtriser techniquement et économiquement des systèmes de production dans un environnement de plus en plus automatisé. Cette fonction s'est étendue à toutes les industries et s'est répandue même dans les services. Cette expansion s'explique par la croissance des coûts des pannes et leurs conséquences dans les systèmes de production de plus en plus complexes et sophistiqués. La maintenance participe aux résultats finaux de l'entreprise et devient une des fonctions essentielles dans un système de production. Elle est passée du centre de coûts au centre de profit et participe à la compétitivité de l'entreprise dans un milieu concurrent.

L'objectif de ce chapitre est de définir ce que sont une fonction et un service de maintenance pour pouvoir proposer ce service ou cet ensemble de services à travers une plateforme distribuée communiquant sur le web. Notre contribution débute par une analyse du processus de maintenance, qui doit nous permettre de déterminer un agencement de briques intelligentes encastrées les unes dans les autres et concourant à la maintenance des équipements et à la gestion des stratégies de maintenance.

Les concepts et définitions de base liés à la sûreté de fonctionnement ainsi qu'à la maintenance industrielle sont présentés au paragraphe 2 suivi par l'introduction de la fonction de maintenance au sein d'une entreprise dotée d'un service propre, d'un budget et d'une autonomie de gestion. Dans le paragraphe 3, nous introduisons une analyse détaillée du système de maintenance. Le processus de maintenance, les données ainsi que les acteurs de ce système y sont

définis et un état de l'art des systèmes informatiques dédiés à la maintenance est présenté. Le processus est ensuite structuré selon ces différents systèmes informatiques. Dans le paragraphe 4, nous définissons et classifions différentes architectures associées aux systèmes informatiques en maintenance et nous proposons une nouvelle architecture de s-maintenance basée sur la sémantique commune des systèmes.

2. Définitions de base de sûreté de fonctionnement et de maintenance

Dans cette partie nous allons introduire les concepts et les définitions de base de la norme [Afnor, 2001] concernant la sûreté de fonctionnement et plus précisément la maintenance.

La sûreté de fonctionnement est « un ensemble des propriétés qui décrivent la disponibilité et les facteurs qui la conditionnent : fiabilité, maintenabilité, et logistique de maintenance ».

La disponibilité est « une aptitude d'un bien à être en état d'accomplir une fonction requise dans des conditions données, à un instant donné ou durant un intervalle de temps donné, en supposant que la fourniture des moyens extérieurs nécessaires est assurée ». Cette aptitude dépend de la combinaison de la fiabilité, de la maintenabilité et de la supportabilité de maintenance.

La fiabilité est « une aptitude d'un bien à accomplir une fonction requise, dans des conditions données, durant un intervalle de temps donné ».

La maintenabilité est « dans des conditions données d'utilisation, une aptitude d'un bien à être maintenu ou rétabli dans un état où il peut accomplir une fonction requise, lorsque la maintenance est accomplie dans des conditions données, en utilisant des procédures et des moyens prescrits ».

La supportabilité est « une aptitude d'une organisation de maintenance à mettre en place les moyens de maintenance appropriés à l'endroit voulu en vue d'exécuter l'activité de maintenance demandée à un instant donné ou durant un intervalle de temps donné ».

La logistique de maintenance représente « des ressources, services et moyens de gestion nécessaires à l'exécution de la maintenance ».

La fonction requise est définie comme « une fonction, ou un ensemble de fonctions d'un bien considérées comme nécessaires pour fournir un service donné ».

La maintenance est « un ensemble de toutes les actions techniques, administratives et de gestion durant le cycle de vie d'un bien, destinées à le maintenir ou à le rétablir dans un état dans lequel il peut accomplir la fonction requise ». Maintenir un bien est considéré comme étant une action à long terme contrairement à l'action rétablir, considérée à court terme.

2.1. Types et niveaux de maintenance

La norme [Afnor, 2001] distingue différents types de maintenance appelés parfois stratégies de maintenance présentées à la figure 1.1.

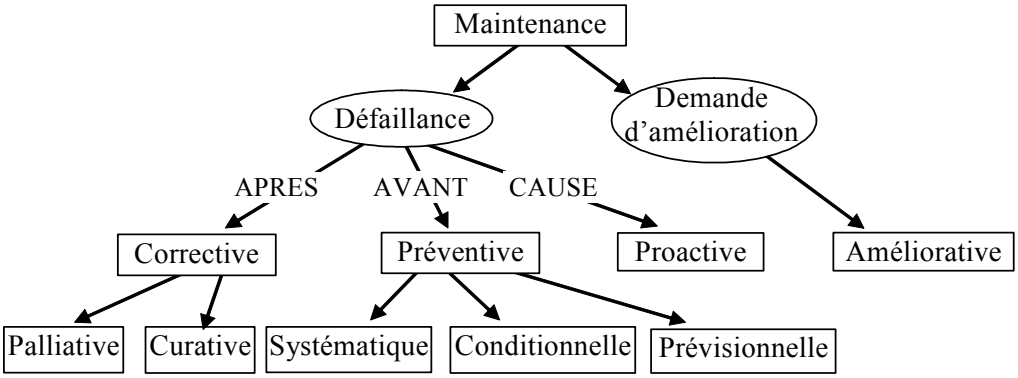


Figure 1.1. Types de maintenance

La maintenance corrective exécutée après détection d'une panne est destinée à remettre un bien dans un état dans lequel il peut accomplir une fonction requise. Une intervention de la maintenance corrective peut être de caractère :

- provisoire type dépannage – *maintenance palliative*,
- définitif type réparations – *maintenance curative*.

La maintenance préventive exécutée à des intervalles prédéterminés ou selon des critères prescrits, est destinée à réduire la probabilité de défaillance ou de la dégradation du fonctionnement du bien. Cette maintenance se dissocie en :

- *La maintenance systématique* exécutée à des intervalles de temps préétablis ou selon un nombre défini d'unités d'usage indépendamment de l'état du bien.
- *La maintenance conditionnelle* basée sur une surveillance du fonctionnement du bien et/ou des paramètres significatifs de ce fonctionnement intégrant les actions qui en découlent.
- *La maintenance prévisionnelle* exécutée en suivant les prévisions extrapolées de l'analyse et de l'évaluation de paramètres significatifs.

La maintenance proactive est un terme émergent, selon [Monchy, 2000] elle « repose sur l'exploitation du retour d'expérience et sur l'analyse approfondie des phénomènes pathologiques à

l'origine des défaillances. » Elle est basée sur la maintenance conditionnelle et prévisionnelle et s'enrichit du diagnostic des causes de pannes et défaillances.

La maintenance améliorative est exécutée afin d'améliorer les fonctions ou la fiabilité d'un équipement.

Nous présentons une typologie des différents travaux de maintenance communément utilisée en entreprise. Cette typologie classe la maintenance suivant le niveau de qualification requis pour être exécutée par un employé. La tendance actuelle est de caractériser l'organisation de maintenance par les trois niveaux regroupant les cinq anciens niveaux de la maintenance [Monchy, 2000] :

- ***Niveau I*** représente la maintenance de première ligne transférée progressivement aux opérateurs de production, assistés, si nécessaire, par les techniciens de maintenance de l'antenne sectorisée.
- ***Niveau II*** représente le domaine d'action privilégié des équipes polyvalentes de techniciens de maintenance. Diagnostic, interventions, mise en œuvre d'améliorations, etc.
- ***Niveau III*** représente les travaux spécialisés souvent sous-traités pour que la maintenance puisse recentrer ses moyens sur son savoir-faire.

2.2. Politiques et stratégies de maintenance

Dans la gestion de maintenance nous distinguons les notions de politique et de stratégie de maintenance :

- ***La politique de maintenance*** définit les méthodes de management utilisées en vue d'atteindre les objectifs de maintenance.
- ***La stratégie de maintenance*** définit un type ou une combinaison de types de maintenance (corrective, préventive, améliorative) à appliquer à un équipement donné dans le but d'optimiser la production sur cet équipement.

Parmi différentes politiques de maintenance présentées dans la littérature [Cuignet, 2002], [Fréderic, 2003], [Boucly, 1990], [Monchy, 2000] nous citons le concept de la maintenance productive totale (TPM), la maintenance qualité totale, centrée sur la fiabilité (MBF), basée sur le risque (MBR), la maintenance en conception, le benchmarking ou encore le maintien en conditions opérationnelles (MCO). Différentes politiques de maintenance décrites en annexe A ont été appliquées et utilisées dans différents domaines industriels ainsi que dans différents pays et consécutivement dans différentes cultures.

Les deux premières (TPM et MBF) ont longtemps été opposées et orientées différemment, la TPM vers l'industrie manufacturière et la seconde MBF vers des industries de procès [Despujols, 2004]. Récemment ces deux approches ont été considérées complémentaires, la TPM

correspondant plutôt à l'organisation en visant l'objectif de la disponibilité et la MBF correspondant au choix de types de maintenance en visant l'objectif de la sûreté et sécurité ainsi que la maîtrise des coûts.

La maintenance basée sur le risque est issue de la maintenance basée sur la fiabilité et a pour objectif la maîtrise des risques pour la sécurité. La maintenance en conception est un concept plus large s'intéressant davantage aux processus de fabrication qu'elle cherche à améliorer de façon continue.

Nous nous situons dans la dernière politique de maintenance, c'est-à-dire dans la politique de maintien en condition opérationnelle qui a pour ambition d'être la plus complexe ainsi que d'utiliser les nouvelles technologies d'Internet, de transfert de données, de retour d'expériences etc. Cette politique est basée sur la stratégie de maintenance prévisionnelle (autrement dit pronostic) et les nouvelles formes de la maintenance plus coopératives et plus proactives sont indispensables. La maintenance coopérative dans ce cas « désigne une approche de maintenance collective organisée autour d'un ensemble d'acteurs partageant un but commun » [Muller, 2005]. Selon la même source [Muller, 2005] la maintenance proactive s'attache à surveiller les causes potentielles de défaillance alors qu'une approche réactive ne s'attaque qu'aux symptômes de cette défaillance.

2.3. La fonction et le service de maintenance dans une entreprise

Après avoir présenté quelques définitions de base, nous allons préciser le rôle de la maintenance dans une entreprise. Dans l'histoire de la fonction de maintenance nous pouvons identifier son évolution d'un simple service d'entretien comme une sous fonction de la production vers un système complexe de maintenance [Monchy, 2000]. L'entretien correspond à des actions de maintenance systématique. Le dépannage et la réparation étaient faits après défaillance indépendamment des coûts engendrés. Avec le temps, la question de sécurité s'est posée, et avec cela la maîtrise des systèmes automatisés, ainsi que la prévention des défaillances ; nous parlons de maintenance du parc matériel.

Dans les années 80, une phase de structuration et de normalisation des services de maintenance se met en place. Puis l'évolution des marchés, leur mondialisation et l'accent sur le profit et la compétitivité de l'entreprise provoquent la mise en place de nouveaux concepts concernant l'organisation de la production aussi bien que l'organisation de la maintenance. En même temps, l'aspect qualité commence à jouer un rôle important tout comme la fonction de sûreté de fonctionnement et, plus particulièrement, la fonction de maintenance dans l'entreprise. Pour bien analyser la maintenance industrielle, nous allons définir et préciser la différence entre deux termes essentiels concernant la maintenance dans une entreprise [Boucly, 1990]. Il s'agit de la fonction maintenance et du service maintenance.

La fonction de maintenance des équipements en service devient dernièrement une des fonctions essentielles de l'entreprise. Elle contient le bureau de méthodes, l'ordonnancement (approvisionnement) et la réalisation de la maintenance.

Le service de maintenance représente une structure ayant un certain niveau d'autonomie et assurant les tâches de la fonction de maintenance. Le service de maintenance effectue les interventions de maintenance avec son personnel et son budget propre. Cette autonomie de gestion peut éventuellement permettre d'externaliser la fonction de maintenance en la sous-traitant à une entreprise de maintenance et donc passer la maintenance d'un objectif secondaire à l'objectif primaire.

Les entreprises cherchent à améliorer le système actuel de maintenance par l'entremise de deux niveaux stratégiques [Francastel, 2001] :

- Le premier niveau vise à renforcer le service de maintenance dans l'entreprise elle-même, c'est-à-dire au niveau des ressources internes – nous parlons ***d'internalisation de la fonction de maintenance.***
- Le deuxième niveau fait appel à des ressources externes de l'entreprise par le biais de sous-traitants (réduction de service de maintenance) - nous parlons ***d'externalisation de la fonction de maintenance.***

D'autres stratégies combinant ces deux dernières sont possibles et permettent ainsi la création de réseaux d'entreprises clients et fournisseurs plus ou moins indépendants les uns des autres.

Notre but étant de proposer un service automatisé de maintenance via une plate-forme distribuée d'e-maintenance, nous présentons dans la partie suivante une analyse du système de maintenance contenant le processus de maintenance, le flux d'information et leurs types ainsi que les acteurs et les systèmes informatiques déjà existants dans le domaine et destiné aux différentes tâches de maintenance.

3. Analyse du système de maintenance

Quant à la définition d'***un système***, selon [Spadoni, 2004] c'est « un ensemble de méthodes et de procédés destinés à assurer une fonction définie ou à produire un résultat. Il est donc important d'en préciser son objectif et ses concepts. » Les concepts de base d'un système d'entreprise sont les entités de l'entreprise, les processus opérationnels et les rôles des acteurs [Eriksson & Penker, 2000]. Les processus opérationnels décrivent le flux de contrôle du système, autrement dit le comportement opérationnel du système [Vernadat, 1996]. Un système est décrit comme un

ensemble d'éléments en interaction entre eux et avec l'environnement, intégré pour rendre à son environnement les services correspondants à sa finalité.

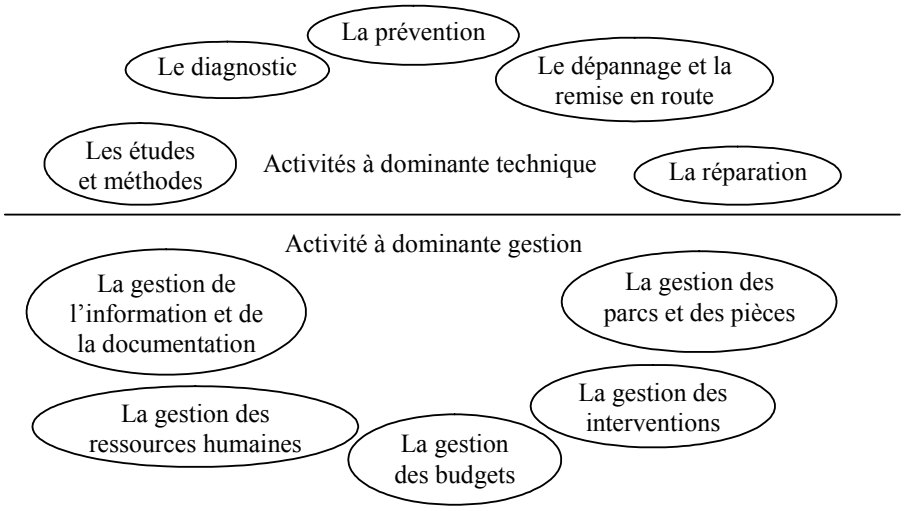


Figure 1.2. La fonction maintenance et son contenu [Retour et al., 1993]

Pour analyser le système de maintenance, nous allons étudier le contenu de la fonction maintenance (voir la figure 1.2.) présentée dans [Retour et al., 1993] comme un ensemble d'activités regroupées en deux sous-ensembles :

- les activités relatives à ses aspects techniques,
- les activités relatives à la gestion et à l'organisation de la maintenance.

Le champ à dominante technique représente les tâches industrielles primaires d'entretien et est souvent englobé dans la supervision notamment quand il s'agit de traiter des systèmes complexes tels que des centrales nucléaires [Piechowiak, 2003]. Le champ des activités à dominante gestionnelle représente les tâches de gestion du parc d'équipements, les ressources humaines et matérielles, la documentation, etc. et d'organisation des activités correspondantes.

Dans l'analyse du système de maintenance nous nous intéressons à l'échange d'informations. Notre objectif est de décrire le système d'information qui en découle et qui peut ensuite servir de support pour les systèmes informatiques. Ces systèmes sont proposés dans l'objectif d'automatiser la réalisation des différentes tâches identifiées dans le processus de maintenance.

3.1. Le système d'information dédié au service de la maintenance

D'abord nous rappelons la définition **d'un système d'information.** Selon [Wikipedia, 2005] c'est « un système, automatisé ou manuel, qui comprend des hommes, des machines et des méthodes organisés pour assembler, transmettre et répandre des données qui représentent de l'information pour les utilisateurs. Selon leur finalité principale, on distingue des systèmes

d'information supports d'opérations (traitement de transaction, contrôle de processus industriels, supports d'opérations de bureau et de communication) et des systèmes d'information supports de gestion (aide à la production de rapports, aide à la décision…). »

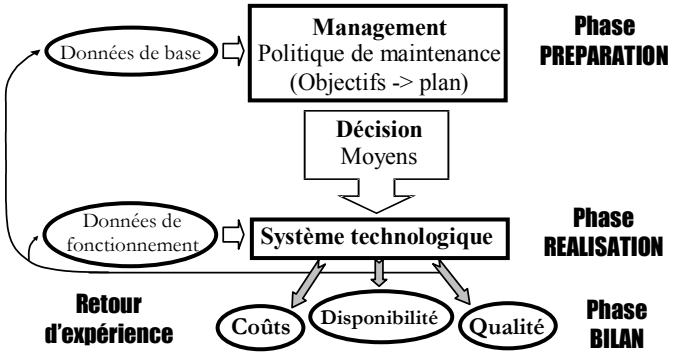


Figure 1.3. Analyse du système de gestion de la maintenance [Rasovska, 2002]

L'analyse du système de gestion de maintenance inspirée de différentes études dans la littérature [Boucly, 1990], [Francastel, 2001], [Monchy, 2000] est présentée à la figure 1.3. Dans la phase de préparation, le management définit la politique générale de maintenance ce qui revient à développer un plan de maintenance en fonction d'objectifs adoptés. Il décide ensuite des moyens nécessaires pour réaliser ce plan. Dans la phase de réalisation, un système technologique est mis en place, y compris l'organisation du système de maintenance, l'ordonnancement des travaux, la logistique des ressources humaines et matérielles, la gestion de la documentation, etc. Finalement, la phase de bilan permet de récapituler et comparer les objectifs fixés et moyens utilisés avec les résultats obtenus en analysant des indicateurs de performance. Le système de gestion opère avec un volume important de données et d'informations dissociées sur les données de base et données de fonctionnement.

3.1.1. Processus de maintenance

Un processus de maintenance selon [Spadoni, 2004] est « un enchaînement d'activités contrôlées ou interactives ». En collaboration étroite avec l'entreprise, nous avons modélisé par des scénarii les différentes activités d'un système de maintenance suivant les types de maintenance existants. Nous présentons ce processus de maintenance comme une suite logique et récurrente d'opérations techniques et gestionnaires, décrite à la figure 1.4, et valide pour tout type de maintenance.

Cet enchaînement d'activités sera plus détaillé dans le tableau 1.1 où les quatre types de maintenance, à savoir la maintenance corrective, préventive, proactive et enfin la maintenance améliorative sont prises en compte. Les activités, autrement dit les processus opérationnels, sont décrites plus en détail ci-après [Rasovska et al., 2004a].

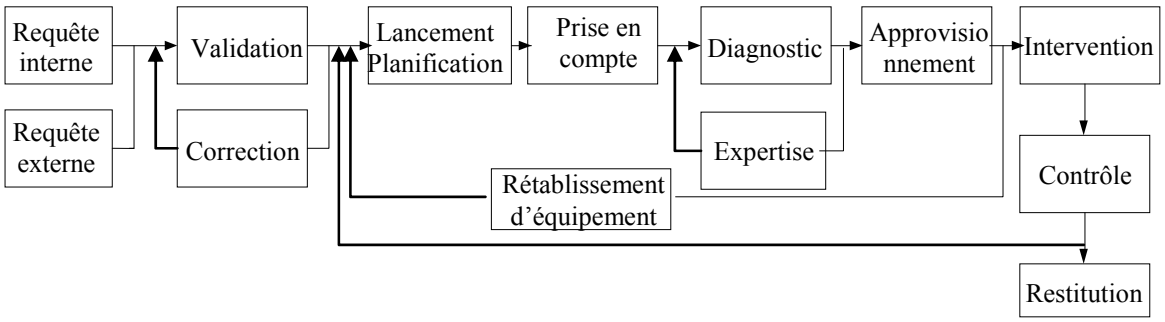


Figure 1.4. Processus général de maintenance

Type de maintenance	Demande	Déclenchement	Préparation	Validation & correction	Lancement planification	Ordonnancement	Prise en compte	Diagnostic Expertise	Approvisionnement	Intervention	Contrôle & restitution d'équipement
Corrective	Client	Requête interne externe		X	X		X	X	X	X	X
Préventive	Client	Requête interne	X	X	X	X	X			X	X
Proactive	Client	Requête du capteur	X	X	X	X	X	X	X	X	X
Améliorative	Client		X	X	X	X	X		X	X	X

Tableau 1.1. Le processus de la maintenance selon les quatre types de maintenance

La demande représente la formulation du besoin du client envers le prestataire des services de maintenance. Il s'agit de la demande exprimée comme la réalisation de la maintenance préventive ou corrective ainsi que toute externalisation de la maintenance ou de la demande pointue exprimée comme la maintenance améliorative, un devis sur un équipement, des travaux lourds etc.

Le déclenchement représente une signalisation souvent automatique d'un problème (défaillance ou panne) ce qui se traduit par une requête appelée demande d'intervention. La requête peut être externe, déclenchée pour la plupart du temps par le client comme c'est le cas de la maintenance corrective et interne, initiée par un opérateur de maintenance qui signale un problème après son contrôle ou une autre intervention. Elle peut être également déclenchée par le système de gestion des interventions préventives dans la GMAO ou par les capteurs ou plus généralement par le système de surveillance comme SCADA. Comme la maintenance améliorative

ne se fait pas régulièrement mais à titre exceptionnel, la phase déclenchement ne fait pas partie de son processus.

La phase ***préparation*** apparaît dans la maintenance préventive, proactive et dans la maintenance améliorative. Les deux premiers cas concernent une première étude de l'équipement, consistant en sa décomposition hiérarchique, de l'analyse fonctionnelle à l'AMDEC (Analyse des Modes de Défaillance, et de leurs Effets Critiques). Cette étude a pour but de développer la stratégie de maintenance la mieux adaptée à l'équipement et éventuellement d'installer les capteurs nécessaires pour assurer la disponibilité et la fiabilité de l'équipement. Dans le cas de la maintenance améliorative, la phase préparatoire contient une étude spécifique de l'équipement en vue de proposition de devis ou de l'amélioration du fonctionnement ou encore de la disponibilité de l'équipement.

L'opération de ***validation et de correction*** se fait après la réception de la demande d'intervention. Celle-ci est corrigée et validée par le prestataire de services de maintenance et renvoyée au client qui exprime les disponibilités ou son accord pour la date de l'intervention.

La planification et le lancement consécutif de l'intervention se font suivant la disponibilité de l'opérateur dont les compétences sont nécessaires pour effectuer cette intervention (ses compétences sont identifiées dans le type d'intervention) et suivant le planning de la production et donc de la disponibilité de l'équipement. La demande d'intervention complétée de la date précise d'exécution est communiquée à l'opérateur en tant qu'ordre de travail.

L'ordonnancement et l'approvisionnement à cette phase est requise dans le cas de la maintenance préventive, proactive et améliorative. Les outils et les pièces de rechange nécessaires pour la réalisation de l'intervention sont identifiés dans l'ordre de travail et peuvent être commandés si nécessaire par les acheteurs auprès des fournisseurs. Suivant le délai d'approvisionnement, la date d'intervention est re-planifiée ou non.

La prise en compte représente un moment important pour les calculs des indicateurs élémentaires pour la gestion de maintenance et notamment pour l'exploitation du retour d'expérience comme par exemple le temps de la réparation.

Le diagnostic et l'expertise de panne sur l'équipement concernent la localisation et l'identification de la cause ainsi que les actions conduisant à sa réparation. Actuellement il est réalisé par l'opérateur de maintenance intervenant sur l'équipement, en se basant sur le premier diagnostic fait par l'opérateur de production pendant la création de la demande d'intervention. Cette phase est présente dans la maintenance corrective et proactive ce qui est dû au fait que pour

les autres types de maintenance, le problème et sa solution, respectivement le diagnostic et l'action de réparation, sont identifiés dans la phase préparation et déclenchement.

Suite au manque d'information sur la défaillance de l'équipement dans la maintenance corrective, ***l'approvisionnement*** ne peut se faire qu'après avoir identifié la panne et sa cause ce qui nous amène à identifier les besoins en outils et pièces de rechange nécessaires pour réaliser la réparation. Dans le cas de la maintenance proactive, l'approvisionnement peut se faire suite au diagnostic et expertise fait précédemment.

L'intervention représente une action de réparation de l'équipement en panne dans la maintenance corrective, une action de l'entretien de l'équipement dans la maintenance préventive et proactive ou encore l'intervention de la maintenance améliorative. Après avoir effectué l'intervention sur l'équipement donné, l'opérateur de maintenance remplit obligatoirement le rapport d'intervention qui sert à l'exploitation du retour d'expérience et à la gestion de la maintenance.

Le contrôle et la restitution de l'équipement sont faits par l'opérateur de maintenance et le client (opérateur de production) qui vérifient le fonctionnement de l'équipement.

3.1.2. Données du processus

Les données de base découlent un inventaire exhaustif du bien, l'identification du patrimoine à maintenir, et les performances exigées de chaque équipement. Il s'agit en particulier de la définition des installations et des équipements en termes de données technologiques, de procédé industriel et de position géographique, précisant les performances requises pour satisfaire les objectifs de production. On peut également compter parmi ces données la hiérarchisation des équipements selon différents critères, des données financières, contraintes, fournitures, prestations, etc.

Les données de fonctionnement se révèlent essentielles pour les opérations du système technologique. Il s'agit des performances réelles des différents sites, d'unités, d'équipements, des sous-ensembles, des composants, etc., de l'état des équipements et composants, du comptage des unités d'œuvre, etc.

3.1.3. Acteurs du processus

Nous avons classé les acteurs de maintenance du point de vue d'un utilisateur de système informatique appliqué en maintenance. Le pouvoir d'avoir la bonne information où et quand j'en ai besoin devient un atout majeur dans les processus industriels d'aujourd'hui et facilite la

résolution des problèmes. Pour cela il est nécessaire d'identifier qui a besoin de quoi, quand et comment. Afin de simplifier la terminologie des acteurs du processus de maintenance souvent très caractéristique dans chaque entreprise, nous avons distingué trois classes générales, à savoir l'opérateur de maintenance, l'expert de maintenance et finalement le manager de maintenance.

L'opérateur de maintenance représente le spécialiste qui intervient directement dans la phase d'intervention sur un équipement. Dans le cas de la maintenance préventive, il s'agit de l'entretien ou de changement préventif, dans le cas de la maintenance corrective l'opérateur réalise le diagnostic et la réparation sur un équipement. L'opérateur est donc responsable de la réalisation et de la performance de l'intervention et il est chargé de la réalisation des rapports d'interventions nécessaires pour le retour d'expérience ultérieur. Il a besoin des informations concernant l'équipement comme la documentation technique, les rapports d'interventions précédentes, les données sur l'état des équipements, des mesures des systèmes de surveillance, etc.).

L'expert de maintenance fait partie de ce qu'on appelle souvent l'ingénierie de maintenance. Il intervient aussi bien dans la phase d'intervention (aide au diagnostic et à la réparation) comme étant l'expert dans un domaine spécifique que dans la phase préparatoire. Dans ce cas, il est chargé de l'analyse d'équipement et décide du choix de la stratégie de maintenance la mieux adaptée, il planifie les interventions préventives, propose des devis afin d'améliorer le fonctionnement ou la disponibilité de l'équipement et veille sur l'accomplissement des règles et des normes concernant la sécurité de l'équipement, du personnel et de l'environnement. Des fois l'expert de maintenance participe également à la réalisation du contrat de maintenance. Il a besoin d'informations plus complexes qu'un opérateur de maintenance dont les informations concernant des indicateurs de maintenance, le contrat de la maintenance, la documentation concernant les lois, règles et normes à respecter, etc. Certains experts sont spécialisés dans l'approvisionnement et négocient les contrats avec les fournisseurs des pièces de rechange ou des outils.

Le manageur n'intervient pas dans les phases techniques du processus de la maintenance mais il supervise la réalisation et fait le suivi du contrat de maintenance sur le site de production (analyse régulièrement les indicateurs du contrat de maintenance). Le manageur est chargé de la préparation des offres de prestations des services de maintenance et des projets à proposer, de la négociation du contrat et il est ensuite responsable de ce contrat et du suivi des engagements envers le client. Il a besoin des informations générales concernant le parc d'équipements dans un site de production, de la documentation concernant les lois, règles et normes à respecter, des indicateurs du contrat et accès au contrat lui-même, etc. Cette classe peut comprendre différents types de manageurs tels que le manageur du site, du projet ou du contrat, le commercial.

3.2. Différents systèmes informatiques en maintenance

Le terme informatique désigne littéralement, le traitement automatique de l'information par ordinateur [Wikipedia, 2005]. ***Le système informatique*** est une partie automatisée d'un ***système d'information***. Nous conviendrons que les systèmes informatiques regroupent les applications de gestion et ses éléments d'accompagnement, les logiciels supports et le matériel. Ce qui caractérise les systèmes informatiques en maintenance, c'est que la gestion et le traitement des informations s'effectuent dans les différentes phases du processus de maintenance en faisant partie du système technologique et représentent les différentes applications de la maintenance ou de la fiabilité. Le développement des systèmes informatiques dans le domaine de la maintenance industrielle a commencé lorsque la maintenance a été reconnue comme fonction fondamentale dans l'entreprise et un accent particulier a été mis sur l'étude approfondie et le développement des procédures de cette fonction.

Informatisation des procédures de maintenance

L'informatisation et l'automatisation de la gestion des entreprises a permis d'informatiser plusieurs procédures de maintenance. Des fichiers informatiques des équipements, des interventions, des stocks, des plans et schémas etc. ont ainsi été créés. L'intégration de ces fichiers et l'automatisation des activités de la maintenance ont été possibles grâce aux progiciels de GMAO (Gestion de Maintenance Assistée par Ordinateur). Les événements quotidiens de la maintenance ont été traités : la panne, l'exécution du préventif, la gestion des stocks.

Interfaçage avec des progiciels

Par la suite, ces progiciels ont dû s'interfacer avec les autres logiciels de l'entreprise telles que les achats et la comptabilité, déjà informatisés précédemment. Les grands progiciels de gestion intégrée (PGI) correspondant au sigle ERP en anglais (Enterprise Resource Planning) représentent une étape suivante dans la rationalisation des processus de l'entreprise et dans l'intégration de la maintenance avec les autres fonctions de l'entreprise.

Evolution du domaine technique

L'informatique a aussi progressé dans le domaine technique de la maintenance. Les techniques modernes d'analyse de maintenance et de contrôle ont vu le jour parallèlement à l'informatique: analyse vibratoire, analyse d'huile, thermographie IR, ultrasons à chaud, etc. Nous pouvons distinguer parmi ces systèmes deux grands groupes : les systèmes d'analyse et les systèmes d'acquisition et de contrôle.

- *Les systèmes d'analyse*, quelques fois couplés aux systèmes experts ont été décrits sous le sigle TTAO (travaux techniques assistés par ordinateur) ou TMAO (techniques de maintenance assistées par ordinateur). Les systèmes d'analyse sont également destinés à fournir de l'aide à la décision en diagnostic, pronostic et réparation des équipements aux opérateurs, etc.

- Parmi *les systèmes d'acquisition et de contrôle*, nous pouvons citer SCADA[1] - système de contrôle et d'acquisition des données, contrôles-commandes des équipements, systèmes de gestion des données techniques et de la documentation, etc.

Intégration de modules intelligents en architectures de maintenance

La présence de ces différents modules intelligents de maintenance nous amène à les faire communiquer et collaborer entre eux (cf. fig. 1.5). Cette construction de modules ou briques intelligentes doit concourir à donner des indicateurs pour prendre la bonne décision en matière de stratégie et de politique de maintenance.

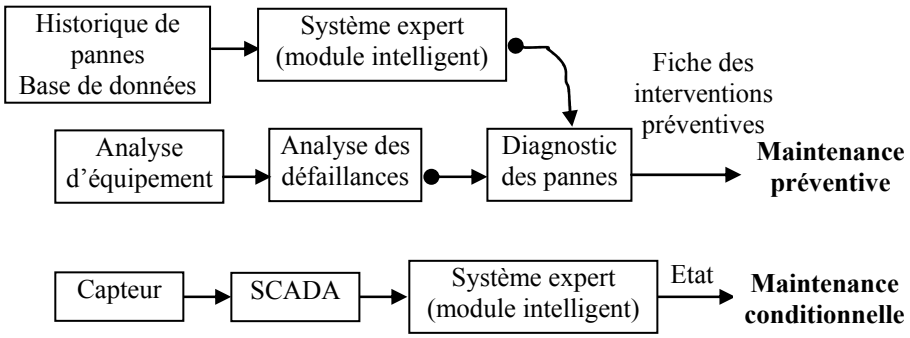


Figure 1.5. Coopération des différents systèmes informatiques

Développement des NTIC

Le développement de nouvelles technologies de l'information et des communications (NTIC), l'extension d'Internet dans l'entreprise, l'intégration des applications, l'émergence de nouvelles politiques de maintenance (RCM, MCO, etc.) indiquent une nouvelle période pour l'informatisation de la maintenance, celle que certains appellent la « maintenance intelligente ». Cela nous amène vers des architectures coopératives et distribuées des systèmes de maintenance communiquant entre eux ou sur une base de réseaux. L'implémentation de ces architectures de maintenance peut se faire à l'aide de plateformes de maintenance dont l'idée majeure est de proposer un service de maintenance via internet. Les plateformes de maintenance proposées dans les projets Proteus ou OSA/CBM peuvent servir d'exemples.

3.3. Analyse du processus de maintenance

Nous proposons de structurer le processus de maintenance afin d'automatiser les tâches attenant à ce processus dans l'optique de fournir des modules d'aide à la décision destinés aux acteurs de maintenance. A partir de l'analyse de la fonction et du processus de maintenance (cf. fig.1.2 et 1.4) nous avons recensé quatre champs imbriqués les uns dans le autres comme le montre la figure 1.6.

[1] Supervisory Control and Data Acquisition

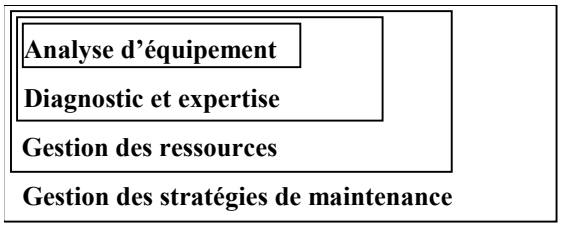


Figure 1.6. Analyse des blocs de maintenance

Analyse d'équipement

Au cœur du processus de maintenance industrielle se trouve un équipement à maintenir et à réparer si nécessaire, puis un parc d'équipements à gérer. Pour cela les analyses et les études de cet équipement constitueront les briques pour gérer la maintenance du parc. Sur le terrain les experts de maintenance à partir de la documentation technique, des données des systèmes de surveillance etc. préparent les plans des interventions préventives, proposent des devis pour la maintenance améliorative permettant d'installer un capteur éventuel en vue d'une maintenance conditionnelle, créent des manuels de réparation, des gammes de maintenance, etc. Les systèmes informatiques associés à ce bloc sont : les systèmes de surveillance, analyses automatisées comme AMDEC, arbres de défaillances, HAFOP.

Diagnostic et expertise

A partir de l'analyse d'équipement, les opérateurs de maintenance doivent pouvoir diagnostiquer pannes et défaillances et décider des actions de réparation. La panne doit être identifiée et localisée. Il faut trouver sa cause en fonction de ses symptômes, la réparer et finalement faire un rapport d'intervention en analysant la situation et sa solution. A ce niveau la documentation technique de l'équipement, les manuels de réparation contenant des procédures de réparation sont nécessaires et les systèmes d'e-documentation, des systèmes experts d'aide au diagnostic et à la réparation peuvent fournir d'aide aux opérateurs. Les deux blocs précédents couvrent la partie technique du processus de la maintenance.

Gestion des ressources

La partie managerielle de processus est représentée par la gestion des ressources et gestion des stratégies de maintenance qui utilisent les indicateurs techniques et financiers afin d'accomplir leurs objectifs. Les indicateurs viennent des rapports d'interventions fournis par les opérateurs de maintenance. La gestion des ressources humaines et matérielles et des stocks de pièces de rechange se fait en fonction du plan de la maintenance préventive et des indicateurs en question. L'objectif de la gestion des rapports d'intervention est de fournir les indicateurs techniques et financiers pour le retour d'expérience concernant le diagnostic et l'expertise sur l'équipement ainsi que pour actualisation et révision des informations précédemment acquises. Nous trouvons à ce niveau les systèmes d'aide à la décision concernant la gestion des ressources aussi bien humaines

que matérielles, la gestion des stocks, la gestion du parc machines, ainsi que les documents concernant l'organisation de l'entreprise, ses processus et activités.

Gestion des stratégies de maintenance

La gestion des stratégies de maintenance concerne des tâches de préparation de contrat de maintenance, l'établissement du budget et son contrôle continu, le choix de la stratégie de maintenance pour un parc d'équipements à maintenir. Les gestionnaires négocient avec le client leurs demandes, préparent les contrats et sont directement responsables de la réalisation des engagements et des résultats. Ils ont besoin des données générales concernant les spécifications de contrat, la législation sur la qualité, sécurité et protection de l'environnement. Ils ont besoin également d'estimation des coûts et des indicateurs techniques et financiers pour leur évaluation et contrôle.

4. Différentes architectures des systèmes de maintenance

Nous allons classifier les différentes architectures de systèmes et d'applications en maintenance y compris les nouvelles plates-formes émergentes, d'une part en fonction de la complexité et du niveau de structuration de l'information partagée, et d'autre part en fonction de l'intensité de la relation entre ces systèmes et applications intégrés dans ces architectures.

Evolution de l'information

L'information utilisée dans le domaine de la maintenance a changé en fonction de l'évolution des technologies d'information et en fonction de la complexité croissante de l'environnement industriel. Dans le passé, cette information a été saisie manuellement sur papier (plans, schémas, manuels) et a été échangée verbalement entre les opérateurs. L'information était donc informelle mise à part la forme papier, puisque le besoin ne s'en faisait pas ressentir. Par contre aujourd'hui, l'information est tout autre. Elle est devenue structurée et formalisée afin d'être manipulée par les systèmes informatiques.

En même temps, l'environnement de l'entreprise devient de plus en plus complexe et les systèmes de production deviennent plus dynamiques, ce qui rend le contexte d'utilisation de l'information plus variable et instable. L'information devient incertaine, elle évolue en fonction du contexte changeant. Une manière de réduire cette incertitude se fait par la mise en place de cette information dans un contexte avec un sens et une direction, par la transformer en connaissance suivant un objectif donné. Cette connaissance devient ensuite, avec d'autres informations et connaissances, la source d'acquisition d'une compétence donnée. Les systèmes informatiques d'aujourd'hui manipulent ces connaissances afin de fournir à ses utilisateurs une aide à la décision

pour la résolution de problème et en vue d'améliorer leurs compétences dans le domaine. Notre étude, que nous développerons au chapitre 4, portera sur l'aide au diagnostic et à la réparation d'équipement, destiné à améliorer la compétence d'un acteur en fonction de l'expertise fournie sous forme de connaissance. Nous passons alors d'une phase de gestion de bases de données à celle de gestion de l'information et enfin à la phase gestion des connaissances et des compétences. Nous allons caractériser plus en détail ces différents types « d'information » dans le chapitre 2 consacré entièrement à la gestion des connaissances dans une entreprise.

Relations entre les systèmes

Avec l'évolution technologique et informatique, les systèmes informatiques, au début indépendants et autonomes, commencent à communiquer, voir coopérer en échangeant et partageant les informations. Plus récemment, les nouvelles technologies de l'information et de la communication (NTIC) ont permis la migration de ces différents systèmes autonomes vers un système intégré où la coopération et la collaboration sont vitales pour tout fonctionnement. Il y a différents types de relations entre les systèmes que nous passons en revue, et qui seront à la base de la classification des différentes architectures en maintenance (cf. fig 1.7).

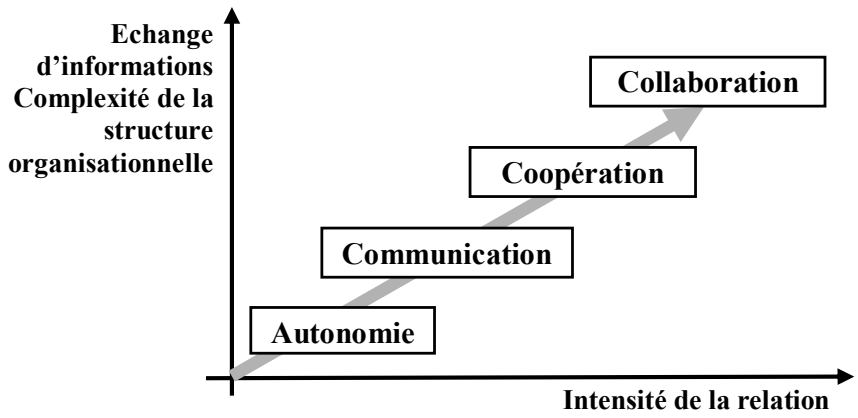


Figure 1.7. Intensité de la relation entre les systèmes

La relation d'autonomie représente un régime sous lequel un système dispose du pouvoir maximal de gestion et est indépendant de tous les autres systèmes et éléments. Il n'y a ni échange ni communication entre ce système et les autres et il doit être auto-suffisant au niveau des informations nécessaires.

La relation de communication est une liaison entre deux ou plusieurs systèmes qui permet des transferts ou des échanges. Les informations transmises lors de la communication ne se limitent plus aux caractères alphanumériques et comprennent également des images, du son et les séquences vidéo. En contexte, le terme communication est souvent employé comme synonyme de télécommunication.

La relation de coopération représente un travail coopératif qui est accompli par une division du travail dans laquelle chaque acteur est responsable d'une partie de la résolution du problème. Dans notre contexte, il s'agit surtout de la coopération technologique et industrielle donc un accord de coopération conclu entre des systèmes indépendants qui s'engagent à réaliser des projets communs de production des services de maintenance.

La relation de collaboration représente un partenariat stratégique en vue d'atteindre à l'excellence en combinant des compétences, des fournisseurs ou des produits divers. La collaboration implique un engagement mutuel des acteurs dans un effort coordonné pour résoudre ensemble le problème mettant en commun des ressources, des informations et des compétences en vue de mieux adapter les organisations à leur environnement.

4.1. Définitions des différents systèmes

Nous proposons une terminologie caractérisant les différents systèmes informatiques en maintenance et nous les classons sous deux axes : le type d'information utilisé dans le système et l'intensité d'une éventuelle relation avec d'autres systèmes informatiques (cf. fig. 1.8). Plus la relation est intense plus les systèmes sont reliés et intégrés et nous parlons d'architectures communes qui seront implémentées à travers des plateformes. Elles sont classées sur une exponentielle car la collaboration entre ces systèmes est atteinte plus tôt que le niveau de la compétence partagée. Le volume d'informations gérées automatiquement est concrétisé par la surface du carré de chaque système et augmente avec l'intensité de collaboration et aussi avec la complexité de l'information partagée. Nous tenons à signaler qu'il y a une parallèle entre notre classification des systèmes et la classification des entreprises telle qu'elle est présentée dans plusieurs travaux [Dedun et al., 2005]. Il s'agit de l'entreprise traditionnelle, distribuée, coopérative et étendue.

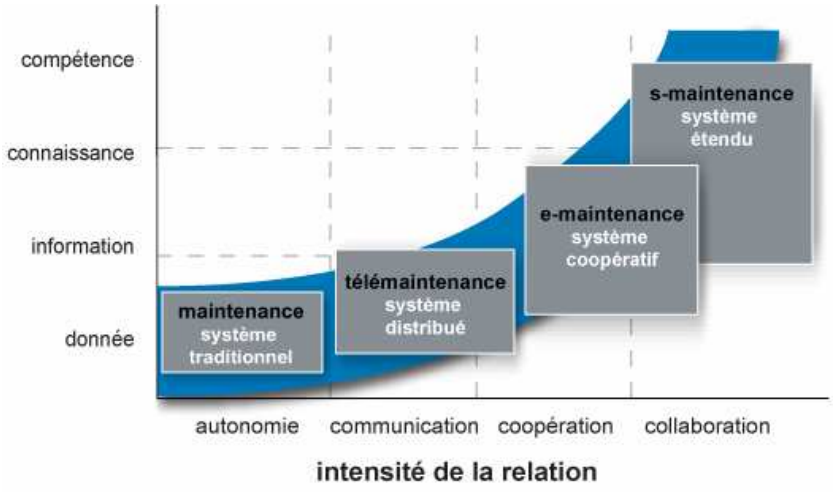


Figure 1.8. Classification de différentes architectures en maintenance

Le système de ***maintenance*** comprend un seul système informatique présent sur le site de production et utilisé sur le site de maintenance. Ce système est autonome sans échange de données avec d'autres systèmes. En parallèle avec la classification des entreprises, cela correspond à l'entreprise traditionnelle, donc nous parlons d'une architecture traditionnelle d'un système d'information.

Le système de ***télémaintenance*** est constitué d'au moins deux systèmes informatiques un émetteur et un récepteur de données et d'informations qui échangent à distance. Selon la définition d'AFNOR la télémaintenance est « la maintenance d'un bien exécutée sans accès physique du personnel au bien ». Nous parlons d'une architecture distribuée, basée sur la notion de distance qui permet de transférer les données par une radio, une ligne téléphonique ou par l'intermédiaire d'un réseau local.

Avec l'extension d'Internet, les systèmes de télémaintenance émergent vers le concept ***d'e-maintenance***. Le système d'e-maintenance sera implémenté sur une plateforme distribuée coopérative intégrant différents systèmes et applications de maintenance. Cette plateforme doit prendre appui sur le réseau mondial d'Internet (d'où le terme e-maintenance) et la technologie web permet d'échanger, de partager et de distribuer des données et des informations et de créer ensemble des connaissances. Ici le concept de la maintenance intelligente peut être exploité et les stratégies de maintenance proactives et coopératives sont mises en place.

Nous proposons une architecture encore plus performante au niveau de la communication et de l'échange des données entre les systèmes - la ***s-maintenance*** (où « s » signifie sémantique) [Rasovska et al., 2005c]. Ce système prend appui sur le concept de e-maintenance avec un échange d'informations basé sur la norme « OKC » issue du web sémantique. La sémantique de l'information échangée nécessite la création d'ontologie du domaine commune aux différents systèmes. Elle permet d'utiliser et créer des connaissances et des compétences ce qui aboutit à l'utilisation des techniques du management des connaissances et permet de capitaliser des connaissances acquises. Les systèmes collaborent, ce qui suppose un effort coordonné pour résoudre ensemble des problèmes. En parallèle avec la classification d'entreprise nous parlons d'une architecture intégrée ou étendue.

4.2. Maintenance

Un système de maintenance est représenté par une application de maintenance ou de fiabilité des différentes activités de la fonction de maintenance telles que logistique, planning des interventions, gestion des stocks (géré par la GMAO, ERP), diagnostic et réparation (systèmes experts, bases de données), surveillance d'un équipement (SCADA, commande numérique sur un équipement). L'architecture de ces systèmes peut varier selon les différents objectifs visés. Par

conséquent nous proposerons de décrire les architectures de ces systèmes par un schéma générique valable pour n'importe quel système d'entreprise composé de deux parties principales, à savoir du système physique et du système de gestion. Le système de gestion produit l'ensemble des résultats ou décisions en se basant sur les informations provenant du système physique [Kaffel, 2001] (cf. fig. 1.9). L'acquisition des informations est manuelle ou plutôt limitée dans son automatisation et les décisions se font donc par l'intermédiaire d'un système d'information.

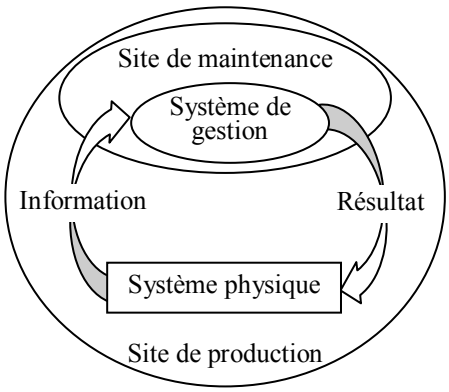


Figure 1.9. Architecture d'un système de maintenance

4.3. Télémaintenance

L'architecture de la télémaintenance est constituée de deux ou plusieurs systèmes ou sous-systèmes éloignés l'un de l'autre qui communiquent et échangent des données entre eux. Le mot télémaintenance est d'origine latino-grecque et est composé du mot grec « téle » signifiant loin et du mot latin «manuterer » qui contient le « manu » comme la main et « tenerer » comme tenir. L'un des systèmes peut fonctionner comme un système d'acquisition de données ; il représente l'émetteur de données structurées. Le deuxième système est le récepteur, fonctionnant comme un système de traitement de données. Le système émetteur peut envoyer les données automatiquement ou comme réponse à une requête de la part du système récepteur des données.

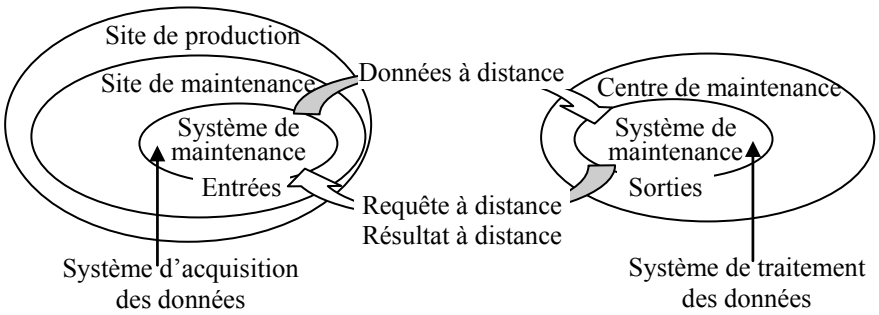


Figure 1.10. Architecture de télémaintenance

Les résultats du traitement de données (sorties) sont utilisés par les acteurs humains ou peuvent être renvoyés au système d'acquisition afin d'agencer l'acquisition des données. Pour que les données puissent être échangées elles doivent être structurées de façon qu'elles soient acceptables par les deux systèmes. Toujours en gardant l'aspect de distance, la télémaintenance peut être installée sur un seul site de production comme il peut être répartie entre différents site de production ou un site de maintenance et/ou un centre de maintenance.

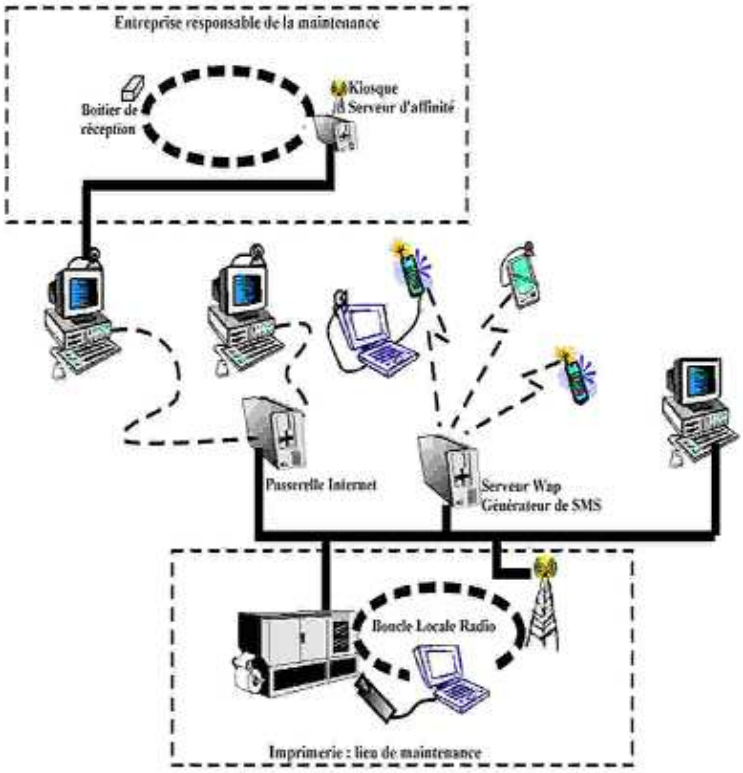


Figure 1.11. Exemple d'une architecture de télémaintenance (TEMIC)

Un exemple d'architecture de télémaintenance (cf. fig. 1.11) a été crée dans le projet TEMIC (TElé-Maintenance Industrielle Coopérative) qui permettait la télémaintenance coopérative : non seulement le personnel de maintenance peut effectuer son travail à distance (télé-maintenance) mais il peut le faire en collaboration avec d'autres experts (travail coopératif) [http://www.telecom.gouv.fr/rnrt/rnrt/projets/res_01_13.htm].

L'accent a été mis sur l'aspect mobilité des membres coopérants à plusieurs niveaux :

- Niveau distant : les acteurs de la télémaintenance seront joignables quel que soit l'endroit où ils se trouvent, via le réseau de téléphonie mobile (GSM/GPRS).
- Niveau local (nomadisme) : par détection de la présence des acteurs de la télémaintenance dans un périmètre prédéfini (de l'ordre de 100m) au sein de l'entreprise qui gère la maintenance, pour joindre le technicien le plus compétant sur un problème particulier.

4.4. E-maintenance

L'architecture d'e-maintenance se fait via un réseau web qui permet de coopérer, d'échanger, partager et de distribuer ces informations aux différents systèmes partenaires de ce réseau (cf. fig. 1.12). Le principe consiste à intégrer l'ensemble des différents systèmes de maintenance dans un seul système d'information [Muller, 2005]. Les systèmes proposent différents formats d'information qui ne sont pas toujours compatibles pour le partage ce qui nécessite la coordination et la coopération entre les systèmes pour les rendre interopérables. D'après [Spadoni, 2004], ***l'interopérabilité*** est « la capacité qu'ont deux systèmes de communication à communiquer de façon non ambiguë, que ces systèmes soient similaires ou différents. On peut dire que rendre interopérable, c'est créer de la compatibilité. » L'architecture d'e-maintenance doit alors assurer l'interopérabilité avec chacun de ces différents systèmes.

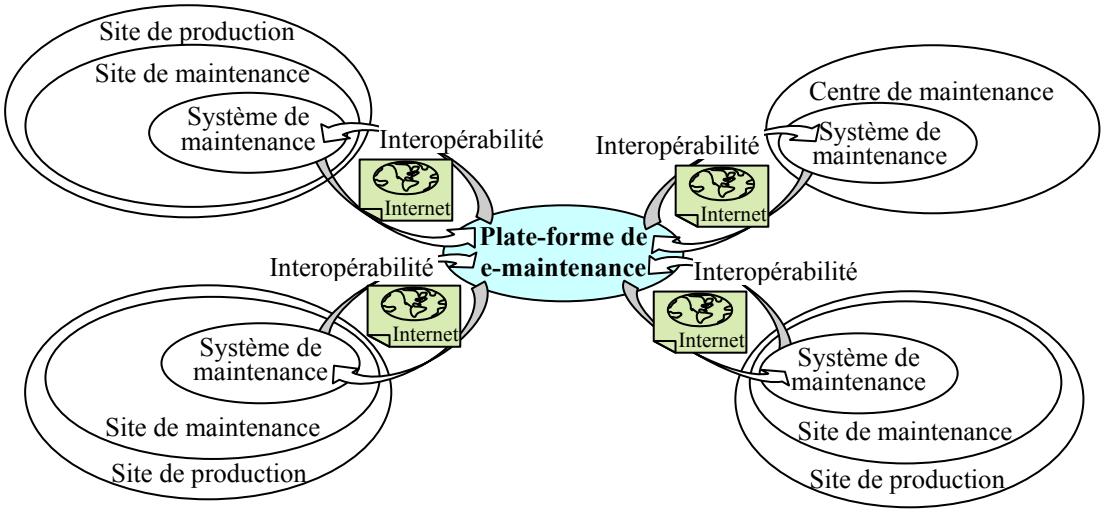


Figure 1.12. Architecture du concept de e-maintenance

Le projet MIMOSA (The Machinery Information Management Open Systems Alliance) a été le premier dans les années 90 aux Etats Unis à développer un système complexe d'information unique pour la gestion de la maintenance [Kahn, 2003]. Le projet est illustré à la figure 1.13. Il a eu pour objectif le développement d'un réseau de collaboration de maintenance en proposant la norme open de protocole EAI (Enterprise Application Integration). L'organisation préconise et développe des caractéristiques d'intégration de l'information pour permettre la gestion et le contrôle de la valeur ajoutée par les solutions ouvertes, intégrées et orientée industrie. Les solutions développées à partir des îlots d'information afin de créer la plateforme d'e-maintenance ont été proposées dans ce projet [Mitchell et al., 1998].

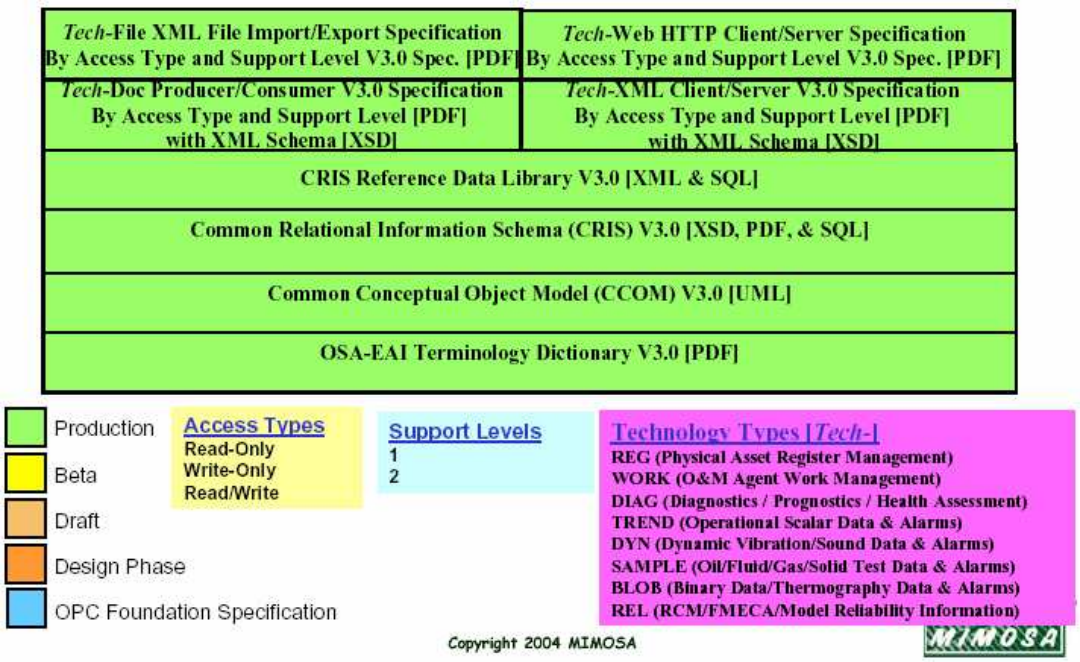


Figure 1.13. Projet Mimosa [Kahn, 2003]

Une architecture fonctionnelle OSA/CBM (Open System Architecture for Condition-Based Maintenance) dédiée au développement de stratégies de maintenance conditionnelle ou prévisionnelle [Lebold & Thurston, 2001] a été développée à partir du schéma relationnel MIMOSA CRIS. Elle contient sept modules flexibles dont le contenu (méthodologie et algorithmes) est configurable par l'utilisateur (cf. fig. 1.14). Elle peut être simplifiée et adaptée à chaque besoin industriel en réduisant des modules.

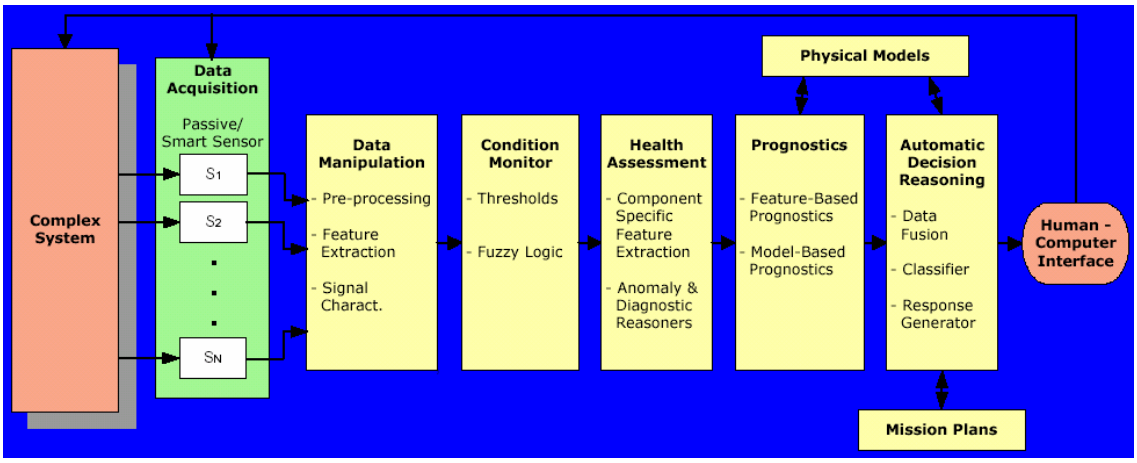


Figure 1.14. Projet OSA/CBM [Lebold & Thurston, 2001]

Une architecture d'e-maintenance a été présentée dans le projet européen Proteus (cf. fig.1.15) [www.proteus-iteaproject.com]. Le projet avait comme objectif de fournir une plateforme distribuée coopérative d'e-maintenance incluant les systèmes existants d'acquisition de données,

de contrôle commande, de gestion de la maintenance, d'aide au diagnostic, de gestion de la documentation, etc. Le concept de cette plateforme est défini par la description unique et cohérente de l'installation à maintenir (une ontologie), par l'architecture générique basée sur les concepts de Web services et en proposant des modèles et des solutions technologiques d'intégration. Ces techniques permettent de garantir l'interopérabilité de systèmes hétérogènes afin d'assurer l'échange et le partage des informations, des données ainsi que des connaissances. Le but de la plateforme est non seulement d'intégrer des outils existants, mais aussi de prévoir l'évolution de ceux-ci au travers de l'introduction de nouveaux services.

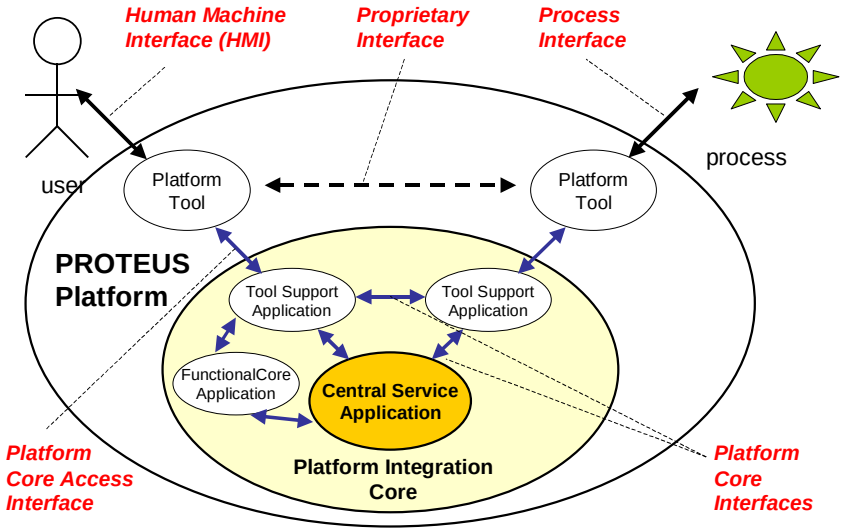


Figure 1.15. Plate-forme de e-maintenance Proteus [www.proteus-iteaproject.com]

4.5. S-maintenance

L'architecture d'une plateforme de s-maintenance prend appui sur l'architecture d'e-maintenance où l'interopérabilité des différents systèmes intégrés dans la plate-forme est garantie par un échange de connaissances représentées par une ontologie. Afin que le partage de l'information dans le réseau coopératif d'e-maintenance soit sans difficulté, nous sommes obligé de formaliser cette information d'une façon à pouvoir l'exploiter dans les différents systèmes faisant partie du réseau. Nous approfondissons donc la coordination entre les partenaires du réseau et nous élaborons une base ontologique du domaine de partage de l'information.

Les systèmes partagent la sémantique créée pour l'architecture commune de la plateforme d'e-maintenance (cf. fig.1.16). Cette base terminologique et ontologique modélise l'ensemble des connaissances d'un domaine. Elle jouera le rôle de la mémoire permettant de mettre en place un système de gestion et de capitalisation des connaissances et exploiter ainsi le retour d'expériences pour améliorer le fonctionnement de système de maintenance. Ce système va employer les outils

du domaine de l'ingénierie des connaissances présentés dans le chapitre suivant de cette thèse. L'outil logiciel doit jouer le rôle d'intégrateur de service capable de se connecter aux autres systèmes, propres aux entreprises. Ce système de connaissances permet d'identifier, capitaliser et restituer la connaissance nécessaire à la conduite, à l'aide d'un environnement d'assistance [Kramer, 2003]. La sémantique a trois niveaux, à savoir les concepts généraux de la maintenance, les concepts du domaine d'application, et les concepts spécifiques à chaque entreprise.

Nous avons développé sur la plateforme d'e-maintenance Proteus un outil d'aide au diagnostic et à la réparation basée sur la connaissance et les concepts de s-maintenance qui sera décrit au chapitre 4.

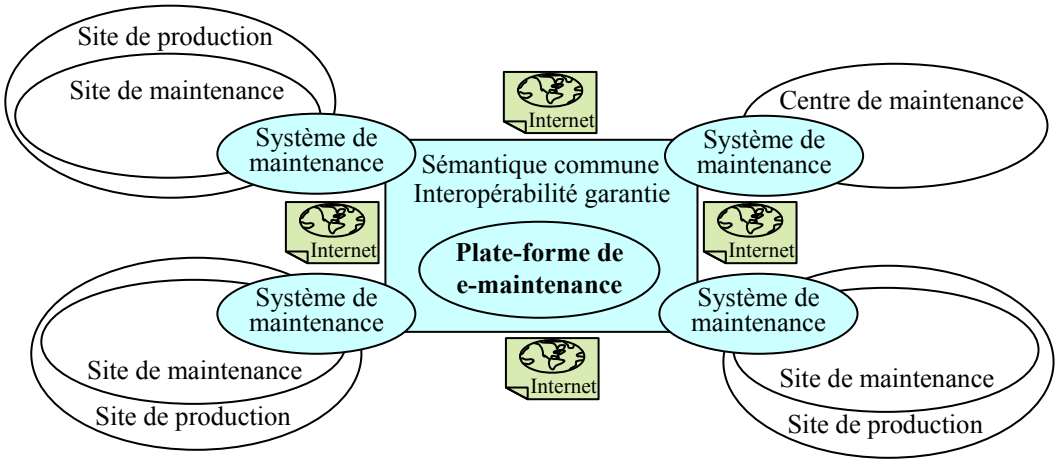


Figure 1.16. Architecture du concept de s-maintenance

Un autre exemple d'architecture de s-maintenance va être développé dans le cadre du projet Européen nommé Shiva (début janvier 2006 – fin 2009) qui succède au projet européen Proteus et se base alors sur sa plateforme d'e-maintenance (cf. fig. 1.17). Cette architecture s'étendra à toute fonction de la maintenance et ne se réduira pas comme dans notre cas au diagnostic et à sa réparation. L'objectif du projet est de proposer une approche basée sur les modèles afin de configurer automatiquement ou à l'aide d'un outil les interfaces des systèmes intégrés dans la plateforme grâce aux applications développées dans le projet. Les informations concernant la configuration vont être générées automatiquement en se basant sur la modélisation des workflows.

Le projet devrait aboutir à trois finalités. La première est la formalisation des stratégies de maintenance et le développement des outils pour dériver automatiquement les processus de modélisation élémentaires. La deuxième est la création des modèles d'équipements/dispositifs utilisant des techniques du langage (morphosyntaxique, sémantique, etc.) et le développement d'un langage de description de l'interface de l'outil général. La troisième concerne les outils, l'assistance et les processus automatiques pour créer la plate-forme de maintenance par les ICAs (Intelligent Core Adapters), Proteus Information Repository and workflows.

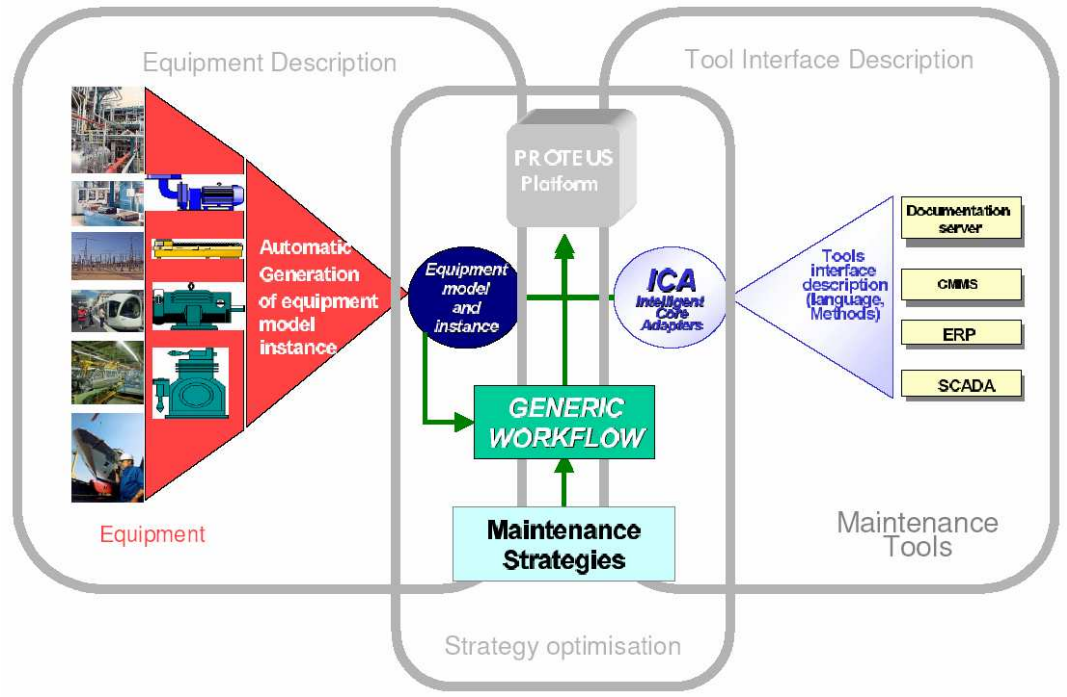


Figure 1.17. Plate-forme de s-maintenance dans le projet Shiva

5. Conclusion

Dans ce chapitre, après avoir rappelé les notions de base liées à la sûreté de fonctionnement et à la maintenance, nous avons défini ce que sont une fonction et un service de maintenance, afin de pouvoir proposer un ensemble d'e-services correspondant, via une plateforme distribuée communiquant sur le web. Pour ce faire, nous avons analysé le processus de maintenance et nous avons mis en évidence quatre champs imbriqués les uns dans les autres, qui permettent de fournir des modules d'aide aux acteurs de maintenance. Ces aides peuvent être proposées suivant différentes architectures de maintenance, allant de la maintenance à e-maintenance en passant par la télémaintenance.

Les nouvelles technologies de l'information permettent d'intégrer ces différents systèmes d'aide via des plateformes. Nous avons proposé une classification des différentes architectures en maintenance en fonction de l'intensité des relations entre les systèmes (autonomie, communication, coopération, collaboration) dans une architecture particulière. Nous avons mis en évidence un type d'architecture de maintenance émergent, la s-maintenance qui prend appui sur des ontologies communes aux différents systèmes. En effet, ces ontologies sont les témoins d'une démarche de gestion des connaissances, que nous introduirons au chapitre 2 et que nous appliquerons au processus de maintenance analysé.

Avant de modéliser ce domaine plus avant, nous passons par un état de l'art des méthodes de gestion des connaissances et plus particulièrement des outils de modélisation et de représentation des connaissances ainsi que les méthodes de raisonnement permettant de manipuler les connaissances modélisées. Nous employons les outils du domaine de l'ingénierie des connaissances présentés dans le chapitre suivant.

Chapitre 2

Capitalisation des connaissances en maintenance

1. Introduction 40

2. Gestion des connaissances 41

2.1. Connaissances dans l'entreprise 42

2.2. Différents modèles de gestion 45

2.3. Tableau de comparaison 48

2.4. Cycle de capitalisation des connaissances 50

3. Capitalisation des connaissances et mémoire d'entreprise 51

3.1. Méthodes dédiées à la construction de la mémoire d'entreprise 52

3.2. Outils et méthodes de l'ingénierie des connaissances 55

3.3. Représentation des connaissances et ontologies 59

4. Formalismes de représentation des connaissances 60

5. Notre démarche de capitalisation des connaissances 62

5.1. Mémoire d'entreprise 63

5.2. Méthode de représentation 65

5.3. Méthode de raisonnement 71

6. Conclusion 72

1. Introduction

Nous avons déjà évoqué différents aspects de l'évolution du contexte industriel, à savoir la mondialisation des marchés, l'instabilité de ces marchés ainsi que l'explosion des technologies de l'information avec les nouveaux modes organisationnels appliqués par les entreprises. Cette évolution accentue l'intérêt porté aux connaissances de l'entreprise. La démarche des responsables d'entreprise d'aujourd'hui vise à augmenter la productivité en améliorant l'organisation, et en particulier en diminuant le gaspillage par une réutilisation des savoirs et des savoirs faire et des compétences développées au cours du temps.

A côté de l'automatisation et de l'informatisation des équipements industriels les entreprises d'aujourd'hui demandent la préservation et la capitalisation du savoir faire des experts. Un aspect important est donc, en plus d'une bonne gestion du personnel, de mettre en place un système permettant de fournir à un individu l'information utile au moment où il en a besoin, dans les meilleurs délais, et de façon exploitable. La gestion des connaissances est devenue un enjeu stratégique capital pour de nombreuses entreprises pour répondre à un environnement économique, technique et informationnel en perpétuelle évolution. Les entreprises doivent être capables de développer des qualités d'adaptation permanente par le développement des compétences de leurs ressources humaines.

Ce deuxième chapitre introduit d'abord les définitions de base de la connaissance ainsi que différents modèles de gestion des connaissances dans une entreprise. Dans le cadre d'offre de services de maintenance via une plateforme web, notre objectif est de mettre à disposition d'acteurs de maintenance, pendant leurs activités professionnelles, différents types de connaissances. La visée de cette mise à disposition, est de faire partager la connaissance métier, de la capitaliser et de la faire vivre. L'expertise dans une entreprise est un bien immatériel à identifier, formaliser et modéliser ainsi à transmettre continuellement [Grundstein, 1994]. Ce capital, une fois en place, fait partie de la mémoire d'entreprise. Charlet [2005] souligne qu'une mémoire d'entreprise s'acquiert par un processus de capitalisation des connaissances qui évolue et vit dans une gestion des connaissances. Afin d'employer ou de développer une méthode adaptée à nos besoins de capitaliser des connaissances dans un processus de maintenance et d'élaborer une mémoire d'entreprise, nous analysons, dans le paragraphe 2, les techniques de gestion de connaissances existantes.

La mémoire d'entreprise contient des connaissances du domaine, de l'entreprise ainsi que différentes sources de connaissances basées sur une ontologie commune. Cette ontologie permet d'utiliser les connaissances stockées dans cette mémoire par les différentes classes d'acteurs et selon leurs différentes tâches à faire dans le processus de maintenance. Elle permet la

communication et l'échange des informations entre eux. La modélisation des connaissances, réalisée dans le cycle de capitalisation, est inséparable de la gestion des connaissances. Nous présentons alors dans le paragraphe 3, une analyse des différentes méthodes de représentation et modélisation des connaissances suivie des différents formalismes de représentation des connaissances dans le paragraphe 4. Cette analyse nous amène à développer une démarche de gestion des connaissances. Elle prend appui sur le cycle de capitalisation des connaissances représentées dans une mémoire d'entreprise basée sur l'ontologie commune du domaine d'expertise. Les connaissances sont représentées par objet ce qui permet d'exploiter l'aspect hiérarchique des concepts représentés. Pour manipuler les connaissances dans la mémoire une méthode de l'intelligence artificielle - le raisonnement à partir de cas - est employée et présentée dans le troisième chapitre de cette thèse.

2. Gestion des connaissances

Nous introduisons d'abord quelques définitions du terme gestion des connaissances selon différentes sources dans la littérature.

La gestion des connaissances désigne un ensemble de concepts et d'outils permettant aux membres d'une organisation de travailler ensemble et de faire le lien entre les informations disponibles, la production de connaissances et le développement des compétences individuelles, collectives et organisationnelles [Ermine, 2001].

Selon [Grundstein, 2000] ***le management, ou la gestion des connaissances*** en français, correspond à l'expression « knowledge management » en anglais et couvre « toutes les actions managériales visant à répondre à la problématique de capitalisation des connaissances dans son ensemble. Il faut aligner le management des connaissances sur les orientations stratégiques de l'organisation, sensibiliser, former, encourager, motiver tous les acteurs de l'organisation, organiser et piloter les activités et les processus spécifiques conduisant vers plus de maîtrise des connaissances, susciter la mise en place des conditions favorables au travail coopératif et encourager le partage des connaissances, élaborer des indicateurs permettant d'assurer le suivi et la coordination des actions engagées, de mesurer les résultats et de déterminer la pertinence et les impacts de ces actions ».

Selon [Charlet, 2005] ***la gestion des connaissances*** vise à rassembler le savoir et le savoir faire sur des supports facilement accessibles, faciliter leur transmission en temps réel à l'intérieur de l'établissement, en différer à nos successeurs et garder la trace de certaines activités ou actions sur lesquelles on peut devoir rendre des comptes à l'avenir.

La connaissance est considérée comme un capital de plus en plus important dans les entreprises et les organisations, notamment dans le secteur des services des industries basées sur la connaissance. Cette connaissance, les expériences et le savoir faire des employés sont stockés et capitalisés afin de créer le capital intellectuel de l'entreprise. Il est défini d'après [Rosario, 1996] comme « la somme totale et la valeur accumulée du partage de la connaissance et de l'expertise de l'entreprise ». Selon la définition précédente, pour que la connaissance devienne le capital intellectuel, elle doit être partagée. Ce capital immatériel devient essentiel pour la création de la valeur dans l'entreprise et représente ainsi un profit.

Une distinction de capital intellectuel de l'entreprise en capital structurel et capital humain a été faite par Lotus Consulting :

- « **Le capital structurel** comprend toute la connaissance codifiée (base de données, procédures et règles de gestion…) provenant de sources internes et externes. »
- « **Le capital humain** comprend les connaissances, les compétences et l'expertise des membres de l'organisation qui sont difficilement formalisables. »

Cette distinction correspond à la division de la gestion des connaissances en gestion de l'information et gestion des hommes présentée dans [Sveiby, 1997]. Les deux groupes diffèrent dans leurs techniques et outils. Le capital structurel concerne la partie technologique de l'information et emploie les logiciels et l'Internet afin de saisir des informations dans les bases de données. Le capital humain améliore la communication intérieurement (dans l'organisation) et extérieurement en créant un environnement afin d'expédier l'innovation et l'utilisation de la compétence.

2.1. Connaissances dans l'entreprise

Plusieurs typologies de la connaissance à l'entreprise ont été proposées dans la littérature [Grundstein, 1995], [Nonaka et Takeuchi, 1995], [Sveiby, 1997]... Une distinction se porte sur la connaissance tacite et explicite, individuelle et collective ou encore organisationnelle, sur la connaissance procédurale et déclarative, etc. Les typologies sont utilisées pour déterminer la connaissance essentielle que l'organisation a besoin de capitaliser [Durstewitz, 1994].

Deux catégories générales de la connaissance sont distinguées dans [Nonaka et Takeuchi, 1995], à savoir les connaissances :

- **tacites** (implicites, non formalisables) dont il est difficile de les décrire - des compétences, des capacités, la connaissance historique de l'organisation, ... Nous intervenons sur le mode de leur prise en compte.

- **explicites** (formalisables) - les connaissances codifiées notées en tant qu'un manuel, des procédures, des plans, des modèles, des documents d'analyse, données, ... Nous intervenons sur le mode de leur formalisation.

Charlet [2005] et Grundstein [2000] distinguent, dans leurs définitions de la gestion des connaissances, deux types des connaissances à gérer (cf. fig. 2.1.) :

- **Le savoir** de l'entreprise et de ses employés, ce qui représente les compétences individuelles et collectives, les connaissances des faits et des dépendances, les plans, les documents, les modèles, les données, etc. Le savoir a le caractère des connaissances explicites et correspond plutôt à la partie gestion de l'information représentée par des objets.
- **Le savoir-faire** représente la connaissance de la résolution des problèmes, des procédures, des stratégies, du contrôle des actions, des expériences individuelles et d'organisation et a plutôt le caractère tacite. Cela correspond à la gestion des hommes et est représentée par les processus.

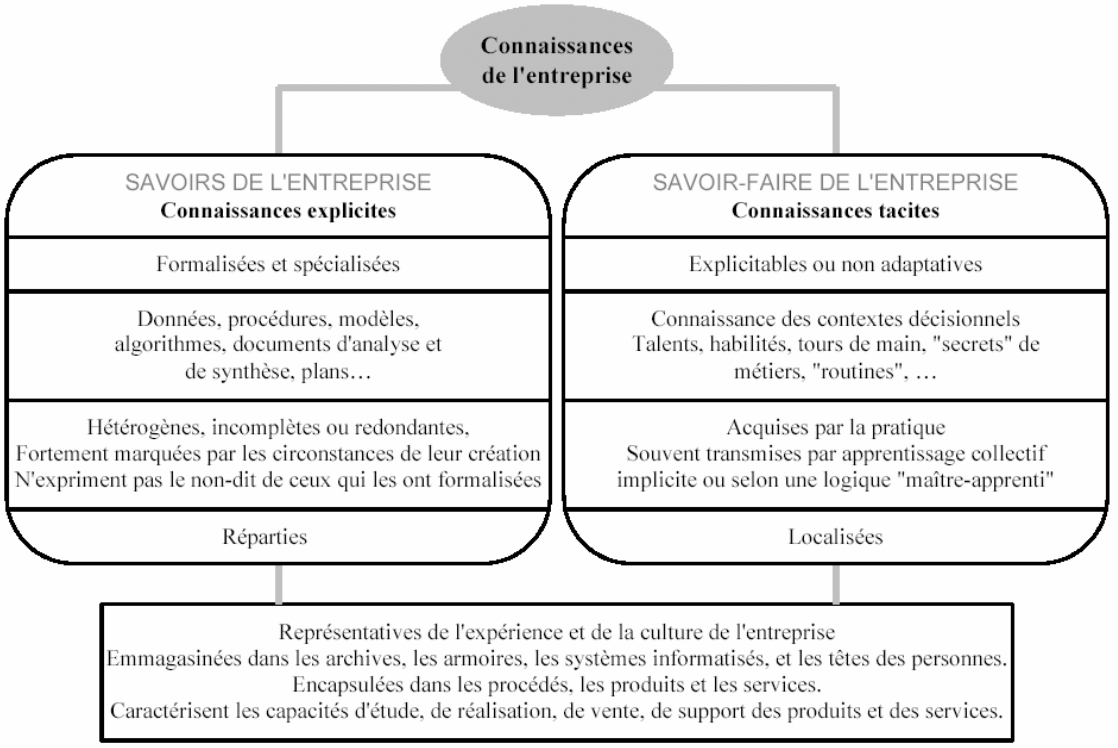


Figure 2.1. Deux catégories de connaissance dans l'entreprise [Grundstein, 2000]

Une autre distinction porte sur la connaissance organisationnelle décrite par Sveiby [1997] comme « intangible asset » et est divisée en trois structures, à savoir externe (les relations avec clients et fournisseurs, l'image de l'organisation, etc.), interne (les brevets, les concepts, les modèles, la culture etc.) et les compétences des employés (leurs expériences et aptitudes).

Dans sa définition de la gestion des connaissances, [Jarboe, 2001] précise que les organisations capturent et communiquent « ressources, perspectives et possibilités tacites et explicites, données, informations, connaissances et peut-être compétence ». Cette dernière définition nous amène à caractériser les différents types d'« information » ou de « connaissance » en fonction de leur complexité, de leur relation respective et de leur niveau de compréhension. Sur la figure 2.2. nous schématisons le lien entre ces différentes entités. Nous avons adopté les définitions selon plusieurs références dans le domaine telles que [Jarboe, 2001], [Prax, 1997].

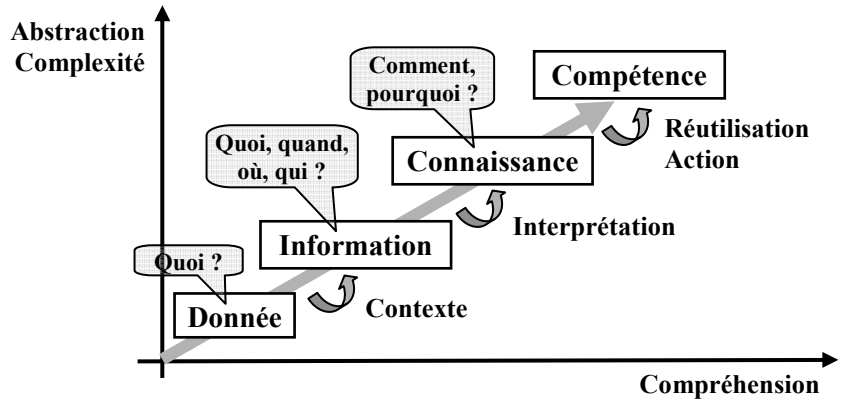


Figure 2.2. Classification des différents types d'information

La donnée est un élément de base qui peut être utilisé pour représenter une information dans des bases de données, à savoir une mesure ou une caractéristique. Nous distinguons deux types de données : les données brutes et les données structurées permettant d'être transférées par un système informatique à distance. Un exemple de données peut être 100 °C.

L'information est une donnée interprétée qui représente un fait réel. C'est la donnée complétée par une description qui indique le contexte : de quelle mesure s'agit-il, quand, où et par qui elle a été prise, etc. Pour notre exemple c'est la température de l'eau dans le moteur prise sur un équipement.

La connaissance est une information assimilée, affinée et synthétisée se rapportant à un contexte spécifié. C'est l'information interprétée et contextualisée ayant un sens ou une signification (pourquoi et comment cette mesure a été prise et le contexte entier de cette mesure souvent structuré pour les besoins de son exploitation). Pour notre exemple, la température de l'eau à 100 °C d'un moteur est trop élevée et signifie un symptôme d'une défaillance.

La compétence est, dans ce cas, une connaissance structurée pouvant être directement exploitée par les utilisateurs afin d'accomplir une certaine tâche ou action. Cela représente « la connaissance dans l'action » donc une résolution d'un problème qui est après cette action validée

ou révisée (actualisée) par le retour d'expérience. Pour notre exemple cela serait un remède pour la défaillance de moteur.

2.2. Différents modèles de gestion

Le domaine de la gestion des connaissances a intéressé pas mal d'auteurs, suivant différents objectifs liés à la connaissance, par exemple certains travaux s'intéressent à l'explicitation de la connaissance tacite, d'autres au partage de celle-ci, etc. Différents modèles et structures ont été proposés par les chercheurs afin de mieux comprendre le phénomène lié à la connaissance et à sa gestion dans une entreprise. Bien que ces structures de gestion des connaissances aient été élaborées suivant différentes perspectives, elles présentent des similarités sur beaucoup d'aspects, notamment dans leur objectif d'observation des organisations gérant les connaissances.

Une première étude de ces différentes structures a été réalisée dans [Holsapple & Joshi, 1999] complétée dans [Lai & Chu, 2000] et que nous actualisons par les travaux présentés dans [Laverde et al., 2003] et les travaux français présentés dans [Grundstein, 2000]. [Lai & Chu, 2000] a proposé une unification des modèles dans une structure générale décrivant le cycle de vie de la connaissance. Nous reprenons cette structure dans un tableau récapitulatif permettant de comparer les différents travaux au nombre de dix-sept que l'on décrira succinctement ci-dessous.

Les différents modèles de connaissances

Wiig [1993] a présenté une structure reposant sur trois piliers concernant l'identification des connaissances du domaine, l'évaluation et l'appréciation de ces connaissances relatives à leurs activités, et enfin la gestion des activités relatives à la manipulation des connaissances (basées sur la création, manifestation, utilisation et transfert de cette connaissance).

Leonard-Burton [1995] a introduit une architecture comprenant quatre compétences de base qui influencent quatre activités de construction des connaissances. Parmi les compétences de base figurent les systèmes physiques (compétences stockées dans les bases de données, logiciels, etc.), les connaissances et les expériences des employés, les systèmes de gestion (les flux d'informations et leurs échange) et finalement les valeurs et normes de l'organisation. Les quatre activités sur les compétences sont la résolution de problèmes partagée et créative, l'implémentation et intégration des nouvelles méthodologies et outils, la création des expérimentations et prototypes, l'importation et l'appropriation des technologies externes.

Andersen [1996] présente un modèle qui est constitué de sept processus coopérant sur l'organisation de la connaissance, à savoir la création, l'identification, la collection, l'adaptation, l'organisation, l'application et le partage. Dans ce modèle ni la nature de la connaissance ni la nature des processus individuels ne sont spécifiés.

Choo [1996] a présenté un modèle de l'organisation « intelligente ». Cette organisation utilise des informations auxquelles elle donne le sens, créé les connaissances et résout des problèmes. Ces trois processus sont reliés comme un continuum des activités définissant une organisation qui possède des informations et connaissances afin de réagir intelligemment.

Le modèle de Van der Spek et Spijkervet introduit dans [Van der Spek et Spijkervet, 1997] identifie un cycle comportant quatre phases de gestion des connaissances. La première phase – la conceptualisation comprend la recherche, la classification et la modélisation de la connaissance. La deuxième phase – le reflet - comprend l'évaluation de la connaissance selon différents critères. La troisième phase – l'action - comprend l'amélioration de la connaissance en créant des nouvelles connaissances, en les distribuant et sauvegardant finalement la rétrospection signifiant l'évaluation des résultats, les effets des actions et comparaison des situations avant et après.

Nonaka et Takeuchi [1995] ont proposé un modèle identifiant quatre types de « conversion des connaissances » tacites et explicites, par création de connaissance organisationnelle. Dans leur modèle de spirale les connaissances organisationnelles sont crées par socialisation (tacites aux tacites), externalisation (tacites aux explicites), internalisation (explicites aux tacites) et combinaison (explicites aux explicites).

Un modèle de transfert des connaissances, présenté dans [Szulanski, 1996], analyse les difficultés du transfert des connaissances dans une organisation. La structure identifie quatre phases : l'initiation, l'implémentation, l'accélération et l'intégration ainsi que quatre facteurs du transfert des connaissances : caractéristique du transfert (ambiguïté causale), caractéristique de la source de la connaissance (manque de motivation et incrédulité observée), caractéristique du destinataire de la connaissance (manque de motivation, manque de capacité de compréhension, etc.), et caractéristique du contexte (relations, contexte organisationnel, etc.).

Alavi [1997] a décrit le processus de la gestion des connaissances dans un service des consultations – KPMG Peat Marwick. Le processus est composé d'une séquence de six phases : l'acquisition, l'indexation, le filtrage, l'enchaînement, la distribution et l'application.

Demarest [1997] a proposé quatre processus de production des connaissances, à savoir : la construction (découverte et structuration des connaissances), la dissémination (processus humain et infrastructure technique), l'incorporation des connaissances et leur utilisation (production des valeurs commerciales des clients).

Taylor [1997] a décrit un cycle de gestion des connaissances par le développement et l'utilisation des connaissances. Le développement comprend la conceptualisation, la revue,

l'internalisation et le partage des connaissances. L'utilisation se réfère au stockage, à la distribution, à l'application et à la révision des connaissances.

Beckman [1996] a introduit une architecture prescriptive contenant neuf pas pour la gestion des connaissances, qui se décline par les verbes suivants : identifier, capturer, sélectionner, stocker, partager, appliquer, créer et vendre.

Davenport et Prusak [1997] ont proposé un modèle de gestion des connaissances basé sur quatre phases : la détermination des objectifs, l'acquisition, la distribution et l'utilisation des connaissances.

Le modèle de Tannembaum et Alliger [2000] examine quatre aspects de la gestion des connaissances, à savoir leur partage, leur accessibilité, leur assimilation (le fait d'apprendre) et leur application (le fait de les utiliser et faire des décisions). Chacun de ces aspects contribue à l'application des connaissances.

Rastogi [2000] affirme que l'organisation doit implémenter un ensemble des opérations afin d'accomplir les objectifs du processus de la gestion des connaissances. Les connaissances doivent être identifiées, tracées, capturées, acquises, stockées, partagées, appliquées afin de générer des nouvelles connaissances comme la dernière étape la plus avancée dans la gestion des connaissances dans une organisation.

Probst [2002] regardent le cycle de la gestion des connaissances comme un processus dynamique (comme Nonake et Takeuchi) constitué de deux cycles – l'un externe et l'autre interne. Le cycle externe est composé de la détermination des objectifs visés par ces connaissances et leur évaluation permettant de contrôler la gestion des connaissances. Le cycle interne est composé des six blocs suivants : identification, acquisition, développement, distribution, utilisation et préservation.

Mark W. McElroy [2002] a défini une architecture concernant le cycle de vie de la connaissance qui suppose l'existence de la connaissance une fois produite, sa capture, sa codification et son partage. Le modèle divise donc le processus de la création des connaissances en deux grands processus, l'un de production de connaissances et l'autre d'intégration de connaissances.

[Grundstein, 2000] présente un cycle qui permet de distinguer les notions de gestion des connaissances et de capitalisation des connaissances. Les quatre processus : Repérer, Préserver,

Valoriser, Actualiser concernent le cycle de capitalisation. Le dernier processus : Manager, concerne la gestion de ce cycle – donc la gestion des connaissances.

Synthèse

Nous pouvons regrouper ces différents modèles selon la spécificité des modèles.

Le premier groupe des modèles traite des « architectures générales » du fait qu'ils décrivent en général le processus de la gestion des connaissances et les connaissances utilisées. Dans ce cas, nous parlons d'un cycle de vie de la connaissance. Il s'agit de la plupart des modèles présentés, à savoir de celui de Wiig, Leonard-Burton, Andersen, Choo, Van des Spek et Spijkervet, Davenport et Prusak, Tannenbaum et Alliger, Rastogi, Probst, McElroy et Grundstein. Les modèles diffèrent dans leurs objectifs, les ressources des connaissances utilisées, les activités de la manipulation des connaissances ainsi que dans leur influence sur la conduite de la gestion des connaissances [Holsapple & Joshi, 1999].

D'autres modèles sont classés plus « spécifiques » et ne proposent pas toujours une architecture sous forme d'un cycle. Nous pouvons pourtant identifier les concepts correspondant aux phases du cycle de vie des connaissances afin d'introduire ces modèles dans le tableau comparatif. Les modèles de Alavi et de Beckman proposent quand même un cycle de vie de la connaissance. Ces cycles sont plus spécifiques du fait que Alavi décrit un processus de la gestion des connaissances dans une organisation particulière et Beckman présente une structure prescriptive et non descriptive comme les autres.

Nonaka et Takeuchi ainsi que Szulanski s'intéressent dans leurs modèles au transfert des connaissances dans une organisation. Demarest et Taylor, de leur côté, s'intéressent aux aspects de la production des connaissances dans l'organisation.

Deux modèles parmi les dix-sept montrent une dissociation entre le cycle de la gestion des connaissances et le cycle de la capitalisation. Il s'agit de modèle proposé par Probst dans les travaux anglophones et le modèle proposé par Grundstein dans les travaux français.

2.3. Tableau de comparaison

A partir du modèle générique de gestion des connaissances proposé par [Lai & Chu, 2000], les dix-sept modèles sont réécrits en fonction du contenu des activités décrites. Ce modèle générique permet de fédérer tous ces modèles et mettre en évidence les points les plus particulièrement développés dans telle ou telle méthode (tableau 2.1). Ce modèle contient donc huit phases.

	Activités						
Structure générale (Lai & Chu, 2000)	Initiation	Elaboration	Modélisation	Préservation Stockage	Distribution & transfert	Utilisation	Révision
Wiig (1993)	Création		Manifestation		Utilisation		Transfert
Leonard-Barton (1995)		Résolution des problèmes Import et adoption des technologies		Implémentation et intégration des nouvelles méthodologies et outils		Experiments et prototypes	
Noenaka & Takeuchi (1995)	Partage des connaissances tacites	Création des concepts	Justification des concepts		Mise à niveau des connaissances	Création des archetypes	
Andersen (1996)		Identifier Collecter Créer	Organiser		Partager	Appliquer	Adapter
Choo (1996)	Donner sens	Création des connaissances				Prise des décisions	
Szulanski (1996)	Initiation				Implémentation	Accéleration	Intégration
Taylor (1996)	Développement des connaissances			Utilisation des connaissances (stockage, distribution, application, révision)			
Alavi (1997)		Acquisition	Indexation Filtrage Enchainement		Distribution	Application	
Beckman (1997)		Identifier Créer	Capturer Sélectionner	Stocker	Partage	Appliquer Vendre	
Demarest (1997)		Construction			Dissemination Incorporation	Utilisation	
Van der Spek & Spijkervet (1997)	Conceptualisation Réflection				Action		Rétrospection
Davenport & Prusak (1997)	Déterminer besoins	Capturer			Distribuer	Utiliser	
Tannembaum & Alliger (2000)				Partage	Accès	Assimilation	Application
Rastogi (2000)	Identification Trace	Caption	Acquisition	Stockage	Partage	Application	Création des nouvelles connaissances
Probst (2002)	Identification	Acquisition	Développement		Distribution	Utilisation	Préservation
McElroy (2002)	Production des connaissances				Intégration des connaissances		
Grundstein (2000)	Détecter : Identifier Localiser		Préserver : Modéliser Formaliser Stocker		Capitaliser: Accès Diffuser Utiliser Intégrer		Actualiser Mise à jour Enrichir

Tableau 2.1. Tableau de comparaison

L'initiation (le déclenchement), en général, implique l'identification des demandes (besoins, exigences, objectifs) sur la gestion des connaissances dans l'entreprise et la création des stratégies de gestion de ces connaissances.

L'élaboration (la génération) a pour objectif de produire les connaissances ce qui revient premièrement à identifier la connaissance dans l'entreprise et à qui elle appartient, ensuite à rassembler puis importer cette connaissance comme l'étude de la connaissance déjà existante.

La modélisation concerne l'organisation de la connaissance et sa représentation dans un dépôt (***stockage*** dans une base de connaissance) afin de la retrouver plus tard. Ce dépôt de la connaissance (base de connaissances) a comme principal objectif de décider quelle connaissance et comment elle devrait être préservée.

La distribution et le transfert consistent à décider de la façon de distribuer et partager la connaissance aux utilisateurs finaux.

L'utilisation résout le problème de « comment utiliser cette connaissance afin de produire la valeur ajoutée ».

Finalement, ***la révision*** concerne le processus, la performance et l'impact de la gestion de connaissance ainsi que la détection des nouvelles connaissances créées pendant le cycle. Il n'est pas nécessaire que ce cycle des activités de gestion de la connaissance soit un processus séquentiel. Chaque activité peut bénéficier du retour d'expérience des autres et vice-versa.

2.4. Cycle de capitalisation des connaissances

Un des modèles se rapprochant du modèle générique et semblant le plus approprié à nos objectifs, est celui proposé par Grundstein et que nous présentons à la figure 2.3. Ce cycle de capitalisation des connaissances est managé par la gestion des connaissances. Il est composé de quatre phases de repérage, préservation, valorisation et actualisation et reflète également les besoins du développement d'un système d'aide à la décision, chacune déclinée sur plusieurs tâches.

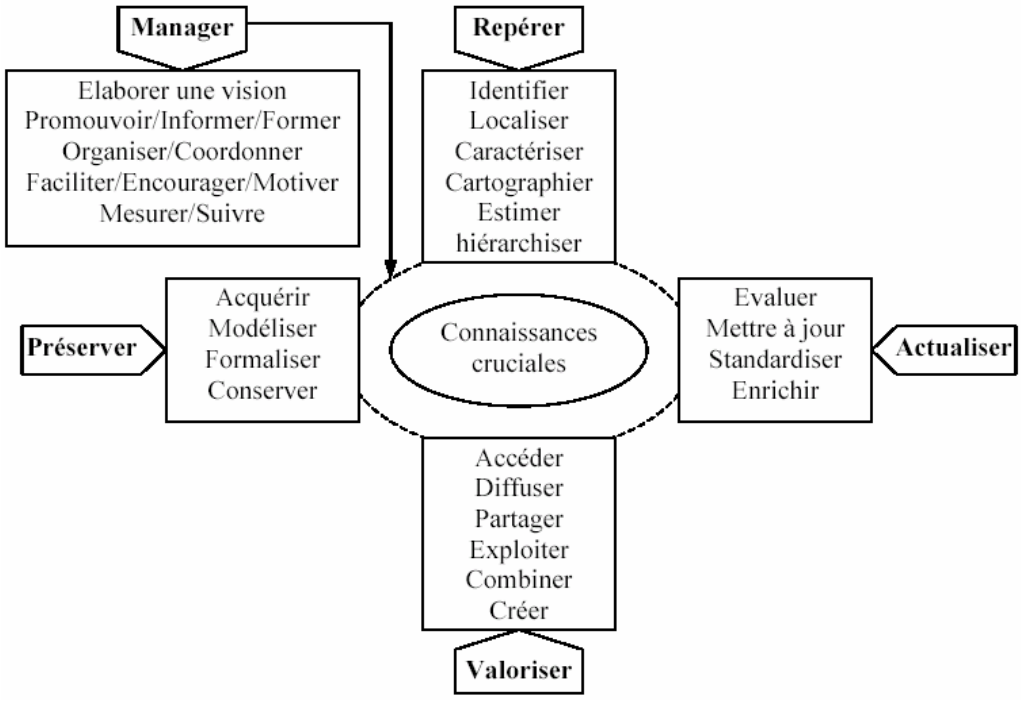


Figure 2.3. Cycle de capitalisation des connaissances [Grundstein, 2000]

Capitaliser les connaissances est définie dans [Grundstein, 2000] comme « considérer certaines connaissances utilisées et produites par l'entreprise comme un ensemble de richesses et en tirer des intérêts contribuant à augmenter la valeur de ce capital ». La première tâche est de repérer les connaissances cruciales en les identifiant, localisant, caractérisant et hiérarchisant.

Ensuite ces connaissances doivent être préservées, donc modélisées, formalisées et conservées. La troisième tâche concerne la valorisation des connaissances, c'est-à-dire qu'il faut les rendre accessibles, les diffuser, les faire partager, les exploiter, les combiner et créer des nouvelles connaissances. Enfin, les connaissances doivent être actualisées, c'est-à-dire évaluées, mises à jour, et enrichies au fur et à mesure de retours d'expérience.

Capitaliser les connaissances implique la constitution d'un capital intellectuel qui sera ensuite valorisé. On ne peut pas capitaliser toutes les connaissances et donc il est nécessaire de ne considérer que les connaissances stratégiques ou cruciales pour certaines activités. Le processus de capitalisation des connaissances permet de réutiliser, de façon pertinente, les connaissances d'un domaine donné, précédemment stockées et modélisées, afin d'accomplir de nouvelles tâches [Simon, 1996].

La capitalisation de la connaissance est définie dans [Matta et al., 2001] comme « une formalisation d'expérience gagnée dans un domaine spécifique ». Le but est de « localiser et rendre visibles les connaissances de l'entreprise, être capable de les conserver, y accéder et les actualiser, savoir comment les diffuser et mieux les utiliser, les mettre en synergie et les valoriser » [Grundstein, 2000]. Il y a plusieurs méthodologies de capitalisation des connaissances combinant les aspects techniques, humains et d'organisation.

3. Capitalisation des connaissances et mémoire d'entreprise

Selon [Baizet, 2004] *la capitalisation des connaissances* désigne le passage d'une mémoire de travail à une mémoire d'entreprise. Ce concept de mémoire d'entreprise tend à caractériser un système qui inclurait toutes les données concernant l'entreprise, les archives, son passé technique, financier, légal, donnant par là une image fidèle de son histoire. Cette mémoire d'entreprise contient des connaissances stratégiques qu'on peut ensuite capitaliser dans un processus de capitalisation des connaissances.

La construction d'une mémoire d'entreprise peut être facilitée par l'emploi d'outils ou de méthodes spécifiques comme REX, MEREX... et d'autre empruntées à l'ingénierie des connaissances adaptées à la définition de mémoire d'entreprise. Nous présentons dans cette partie un résumé de ces méthodes et les principes les guidant. La plupart de ces techniques de capitalisation des connaissances préconisent la création d'un recueil d'information à partir de documents et d'entretiens avec des experts, d'une modélisation du domaine suivie d'une représentation qui consiste à traduire la modélisation dans un formalisme [Matta et al., 2001].

3.1. Méthodes dédiées à la construction de la mémoire d'entreprise

Pour toute opération de capitalisation des connaissances, il est important d'identifier les connaissances stratégiques à capitaliser. Ces connaissances peuvent être rassemblées dans une mémoire d'entreprise qui devrait fournir la bonne connaissance à la bonne personne au bon moment et au bon niveau pour que cette personne puisse prendre la bonne décision. Par ***mémoire d'entreprise*** nous comprenons « un ensemble structuré de la connaissance reliée à l'expérience d'une entreprise dans un domaine donné » [Grundstein & Barthès, 1996].

[Van Heijst et al., 1996] considère ***la mémoire d'entreprise*** comme « représentation explicite et persistante des connaissances et des informations dans une organisation, afin de faciliter leur accès et leur réutilisation par les membres adéquats de l'organisation pour leur tâche ». Cette représentation peut inclure par exemple, les connaissances sur les produits, les procédés de production, les clients, les stratégies de vente, les résultats financiers, les plans et buts stratégiques, etc…

La construction d'une mémoire d'entreprise repose sur la volonté de « préserver, afin de les réutiliser plus tard ou plus rapidement possible, les raisonnements, les comportements, les connaissances, même en leurs contradictions et dans toute leur variété » [Pomian, 1996]. La gestion de la mémoire d'entreprise consiste à détecter les besoins en mémoire d'entreprise, à la construire, diffuser, utiliser, évaluer, maintenir et faire évoluer [Dieng et al., 1998] ce qui rentre parfaitement dans le cycle de la capitalisation des connaissances. Différentes méthodes et techniques existent et sont étudiées par les chercheurs dans chacune de ces phases.

Différents types de la mémoire d'entreprise

Nous pouvons trouver différentes typologies de la mémoire d'entreprise dans la littérature [Pomian, 1996], [Tourtier, 1995], [Grundstein & Barthès, 1996], [Dieng et al., 1999]. Généralement, la mémoire d'entreprise implique :

- *Une mémoire technique (métier)* composée de référentiel, documents, outils et méthodes utilisés dans la profession donnée et des fois appelée mémoire professionnelle dans les cas où le savoir faire des employés de l'entreprise y est compris.
- *Une mémoire organisationnelle (de société)* également appelée mémoire de la société composée de l'organisation de l'entreprise, ses activités, de ressources humaines, management, etc.
- *Une mémoire individuelle* composée de statut, compétences et activités des employés d'entreprise, occasionnellement de leur savoir faire si ce n'est pas le cas de la mémoire technique.
- *Une mémoire de projet* composée de la définition du projet, activités, histoire, résultats et les expériences capitalisées de ce projet.

Différentes méthodes de construction de mémoire

Nous présentons les méthodes les plus connues dédiées à la construction de la mémoire d'entreprise, à savoir REX, MEREX, CYGMA, atelier FX et Componential Framework.

La méthode REX (retour d'expérience) [Malvache & Prieur, 1993] a été définie au départ dans le but de capitaliser les expériences de conception de réacteurs nucléaires au sein du CLA (cf. fig. 2.4). La finalité requise est la constitution d'une base de connaissances pour les futures équipes et le principe consiste à constituer des « éléments d'expériences » extraits d'une activité quelconque et à restituer ces éléments pour qu'un utilisateur puisse les valoriser. Ces éléments peuvent être de trois types – élément de connaissance documentaire (résumé d'un document), élément d'expérience (formalisée au cours d'un entretien) et élément de savoir-faire (d'une personne dans une activité particulière).

REX se décompose en trois étapes : l'analyse des besoins et l'identification des sources de connaissances (spécification et dimensionnement du futur système de gestion des connaissances), construction et mémorisation d'éléments de connaissances sous forme d'une base de données ou une mémoire d'entreprise appelée CEMem, et enfin la mise en place et l'exploitation du système de gestion des connaissances créé. Pour constituer la mémoire d'expérience un modèle descriptif et un réseau terminologique sont définis.

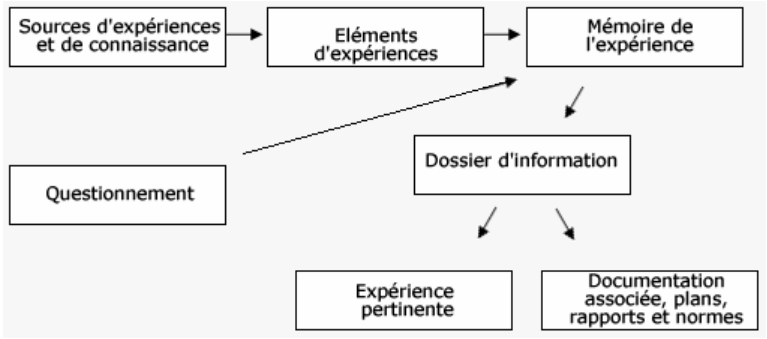


Figure 2.4. Principes de la méthode REX [Malvache & Prieur, 1993]

La méthode MEREX (Mise en Règle de L'expérience) [Corbel, 1997] est une méthode de l'expression. Elle a été mise au point par Jean Claude Corbel chez Renault pour l'amélioration continue des processus de l'ingénierie de ses véhicules (cf. fig. 2.5). Il s'agit d'un système de fichiers simples pouvant être remplies par tous les acteurs de l'entreprise. Le principe de cette méthode consiste à rédiger les fiches d'abord sur simple papier, les fiches sont ensuite informatisées et mises à la disposition des acteurs via l'intranet de l'entreprise. Les fiches contiennent les solutions techniques, leur contexte, preuves de validité et le nom de la personne ayant rédigé la fiche. MEREX permet de capitaliser des connaissances et des expériences portant sur les produits et les processus dans l'industrie automobile afin de les améliorer.

<table>
<tr><td>RENAULT

N° d'ordre :
ECR :
Crée le :</td><td colspan="2">MEREX Produit

Titre de la fiche MEREX Produit/Process (doit être clair et concis)</td><td>Fonction</td></tr>
<tr><td colspan="4">Enoncé/croquis : Il représente le corps de la fiche, puisqu'il énonce la solution au problème exprimé. Le but est d'être clair et concis (environ 15 lignes). La solution doit être applicable, il faut donc qu'elle ait déjà fait ses preuves. La méthode privilégie le concret, l'opérationnel, et non le théorique.

Conséquence d'un non-respect : Conséquences d'une non application de la solution en termes de coût, ou de délai, de qualité…

Support pour vérifier la règle : Explique comment la solution a été trouvée (est-ce grâce à un prototype, des essais en laboratoire…)

Jalon : En cas de besoin, prévoit d'indiquer les dates ou tout autre repère qui permette de savoir quand ce savoir-faire doit être intégré dans le projet/processus

Contexte : Elément très important de la fiche, car les solutions proposées ne sont pas valables tout le temps ni partout

Solution de retouche : Si la solution préconisée est impossible à mettre en œuvre, ou inadaptée, la fiche indique une deuxième solution – moins pertinente, mais qui permet de maintenir le même niveau de qualité pour le client

Eléments de validation : apporte les preuves de l'efficacité de la solution, grâce à des expériences réussies, des lois physiques, des exemples issus de la concurrence…

1/ du respect de la règle :
+
2/ de la règle :
+

Origine et document de référence

Se substitue à…</td></tr>
<tr><td>Rédigée par</td><td>Validée par</td><td>Gérée par</td><td>Exploitée par</td></tr>
<tr><td>1/ Auteur
2/</td><td>1/
2/</td><td></td><td>Le responsable fonction
de chaque projet</td></tr>
</table>

Figure 2.5. Exemple d'une fiche de retour d'expérience MEREX [Corbel, 1997]

La méthode CYGMA (Cycle de vie et gestion des métiers et des applications) [Bourne, 1997] a été développée par la société KADE-TECH. Cette méthode a été appliquée dans les industries manufacturières et spécialement dans l'activité de conception (bureau d'études, de méthodes et d'industrialisation) pour faire des systèmes experts. Cette méthode analyse le contexte où sont situées les connaissances, l'organisation des acteurs, les enjeux existants et leur pondération.

Elle prévoit six catégories de connaissances industrielles : connaissances singulières, terminologiques, structurelles, comportementales, stratégiques et opératoires. En se basant sur ces catégories la méthode définit des référentiels métiers et des bases de connaissances exploitables par des algorithmes de raisonnement déductif. La méthode préconise des entretiens avec les experts et une étude de la documentation de l'entreprise afin de définir un « bréviaire de connaissances » structuré en quatre documents, à savoir le glossaire métier, le livret sémantique, le cahier de règles et le manuel opératoire.

L'atelier FX [Fouet, 1997] est basé sur une méthode appelée M3A qui s'inspire de travaux issus des sciences sociales pour définir une mémoire d'entreprise. La méthode part de la technique « observateur-apprenti » pour capitaliser les connaissances du processus industriel d'un expert afin de formaliser les connaissances d'entreprise. Dans la première étape un observateur est choisi et formé dans les techniques d'observation et de description des connaissances préconisées par la méthode M3A. L'observateur ensuite suit un technicien et analyse ses activités dans une notice d'instruction décrivant les connaissances acquises. Cette notice sera informatisée dans une base technologique. Le logiciel NOMINO a été développé pour indexer les documents dans cette base.

L'approche Componential Framework [Steels, 1993] a été proposé dans le cadre de l'acquisition des connaissances pour développer des systèmes à base de connaissances. Il a été ensuite adapté pour supporter la gestion des connaissances dans une entreprise [Matta et al., 1999]. Ici, une activité est définie selon trois perspectives : tâche (objectifs à atteindre), information (construites pour réaliser les tâches) et méthode (comment les informations sont utilisées pour réaliser les tâches).

Les informations sont organisées dans « Componential Framework » sous forme de deux modèles : du domaine (d'application) et de cas (spécialisation de certaines informations). La perspective tâche utilise les informations appelées sources pour atteindre un but concrétisé par un état cible. Un diagramme de dépendances permet de représenter le flou de données entre les informations « sources », les tâches et les informations « cibles ». Le logiciel KREST a été développé comme support de la méthode.

Ces méthodes présentent l'avantage de ne pas être trop compliquées à mettre en œuvre mais ont comme inconvénient majeur de ne pas suffisamment développer toute la manipulation de connaissances et d'exploitation automatique de ces connaissances. C'est pour cela que notre intérêt se porte sur les méthodes et modèles de l'ingénierie des connaissances.

3.2. Outils et méthodes de l'ingénierie des connaissances

L'ingénierie des connaissances est le domaine qui correspond à l'étude des concepts, méthodes et techniques permettant de modéliser et/ou d'acquérir les connaissances pour des systèmes réalisant ou aidant les humains à réaliser des tâches se formalisant a priori peu ou pas [Charlet et al., 2000].

Des outils et des méthodes de l'ingénierie des connaissances tels que modélisation cognitive et conceptuelle, des langages de représentation et de modélisation, etc., ont été développés pour permettre d'acquérir des connaissances à partir des textes, pour résoudre des problèmes de recherche d'informations sur le web ou pour mettre en œuvre des indicateurs de gestion dans des systèmes d'information ou bien pour gérer et capitaliser des connaissances en entreprise [Charlet et al., 1996].

Ces méthodes peuvent être subdivisées en deux approches différentes : ***approche ascendante et approche descendante*** [Motta et al., 1990]. Ces deux approches de construction de modèle d'expertise ont été comparées dans [Duribreux-Cocquebert & Houriez, 2000]. Nous retiendrons de cette comparaison que l'approche descendante présente l'avantage majeur de séparer la connaissance du domaine de son utilisation et permet ainsi de définir des composants génériques donc réutilisables, tandis que l'approche ascendante présente un modèle structuré d'expertise, certes, mais décrit dans une terminologie propre au problème et manquant d'abstraction. Notre étude s'est donc orientée vers une approche descendante.

3.2.1. Méthodes ascendantes

L'approche ascendante consiste à faire un recueil maximum des données verbales d'un expert et à les regrouper pour former un modèle de connaissance.

Un exemple de méthodes ascendantes est ***la méthode KOD*** ((Knowledge Oriented Design) qui a été développée par Claude Vogel [1988] pour fournir un support à l'activité de l'intelligence artificielle pour une société de services (CISI). Cette méthode est fondée sur une approche anthropologique et propose des techniques d'analyse de discours d'un expert afin d'organiser les données acquises en remplissant les tableaux prédéfinis. A partir d'un discours d'expert, de sources documentaires, d'observations et d'analyse la méthode produit un modèle de représentation de ces connaissances comprenant un ensemble de concepts de base qui sont mis en relation par des liens conceptuels établis a priori.

On distingue deux concepts élémentaires, les objets (éléments statiques) et les actions (éléments dynamiques) sous forme de texte. Cette méthode fait appel à l'analyse des textes et utilise le sens des mots pour organiser les connaissances.

La méthode Métodac (Méthode d'acquisition et de modélisation des connaissances) développée par la société EDIAT est destinée à supporter l'ensemble des activités de spécification de systèmes à base de connaissances, en particulier l'analyse des besoins, l'analyse du réel, l'écriture des spécifications techniques [Le Priol, 2000]. Il existe un certain nombre de méthodes ascendantes basées sur la création d'ontologie à partir des textes et terminologies comme des outils ANA, STARTEX ou encore SYCLADE [Condamine & Rebeyrolle, 2000].

3.2.2. Méthodes descendantes

L'approche descendante « ou approche dirigée par les modèles » se focalise sur la définition du modèle de l'expertise afin de filtrer les connaissances acquises et de guider efficacement le processus d'acquisition.

Une des méthodes de référence dans cette approche est ***la méthode KADS*** (Knowledge Acquisition and Design Structuring) [Wielinga et al., 1992] et son évolution CommonKADS [Schreiber et al., 1999]. La structure de KADS en couche utilise des méthodes génériques de résolution de problèmes qui décrivent les mécanismes de raisonnement à un bon niveau d'abstraction (cf. fig. 2.6). C'est un modèle qui comprend trois couches : les concepts du domaine d'application représentés par le niveau domaine, leurs transformations représentées par le niveau inférence et la structure de contrôle représentée par le niveau tâche.

On distingue deux types de connaissances : un en rapport avec le domaine d'application et l'autre lié aux mécanismes de résolution. Cette méthode est plutôt orientée sur la tâche à réaliser en fonction de laquelle les connaissances du domaine sont choisies. Cette dissociation des connaissances permet la réutilisation des concepts du domaine dans les différentes activités. La place des méthodologies, des outils et des textes y est minimisée.

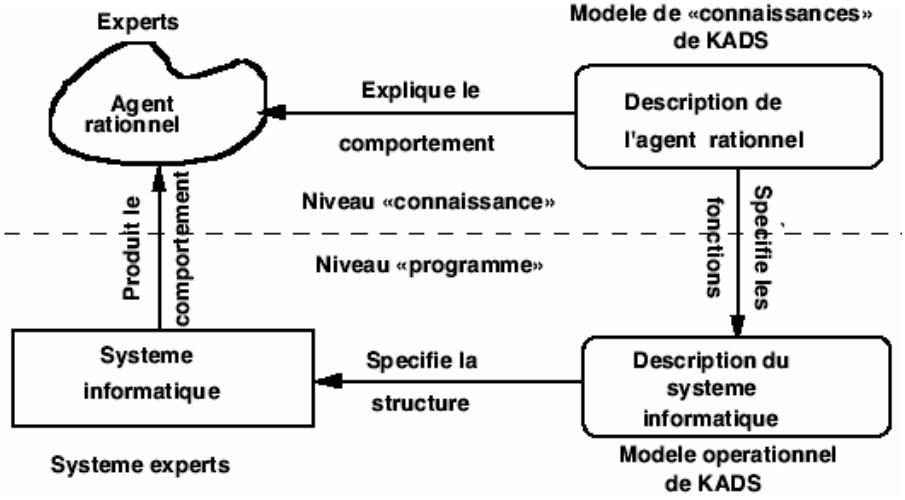


Figure 2.6. Méthodologie de KADS [Wielinga et al., 1992]

La méthode CommonKADS [Wielinga et al., 1993] repose sur la communication, la création de la connaissance et le partage de la connaissance. Cette méthode propose le développement de six modèles pour analyser la connaissance : l'organisation, la tâche, l'agent, la communication, la connaissance et la conception qui donnent différent points de vue sur la connaissance du domaine. Ils sont utilisés dans le cycle de management de projets à base de connaissances qui comporte quatre phases : revue (évaluation de l'état courant et établissement des objectifs), évaluation des risques, planification et supervision du travail. CommonKADS utilise les conventions graphiques d'UML (Unified Modelling Language) pour représenter les modèles et la notion d'ontologie pour la réutilisabilité des connaissances du domaine.

La méthode MKSM (Methodology for Knowledge System Management) proposée par Ermine au CEA (Commissariat à l'énergie atomique) a été introduite dans [Ermine, 1996]. La méthode considère que l'ensemble des connaissances dans une organisation est un système complexe. La modélisation d'un tel système se fait selon trois points de vue : syntaxique, sémantique et pragmatique. Ces trois points de vue comptent encore chacun trois onglets différents, à savoir la structure, la fonction et l'évolution. Les trois composants de ce système de connaissances sont l'information, la signification (modélisation des tâches), et le contexte (modélisation des activités). La méthode propose cinq phases de modélisation : la modélisation du système des connaissances, du domaine, d'activités, du concept et des tâches.

La méthode MASK (Méthode d'Analyse et de Structuration des Connaissances) [Van Craeynest et al., 2000] est une extension de MKMS et est basée sur des principes d'analyse cognitive et sur des retours d'expérience. Elle permet de représenter une expertise sur trois points de vue complémentaires : le contexte (approche métier concerné), les savoirs et savoirs faire (approche par l'ingénierie cognitive), l'informatisation (approche par le génie logiciel). Cette méthode prend en compte l'aspect dynamique de l'acquisition des connaissances.

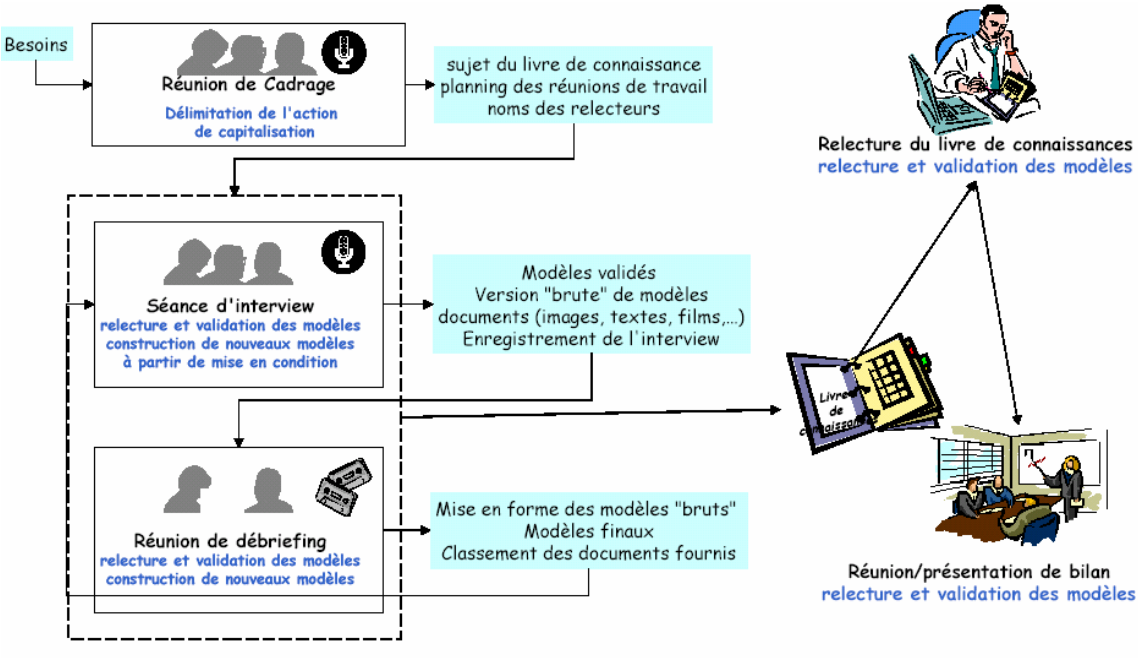


Figure 2.7. Le processus MASK [Van Craeynest et al., 2000]

Ces méthodes de base ne sont pas, à elles seules, complètement satisfaisantes et ne peuvent pas prendre en compte la particularité de chaque type de problèmes. Elles sont utilisées dans différents travaux moyennant des adaptations.

3.2.3. Réutilisation des méthodes existantes

La réutilisation d'une méthode, aussi générique soit elle, nécessite des adaptations. Des perspectives de construction de modèle d'expertise basée sur l'intégration des approches ascendante et descendante se dessinent. Duribreux-Cocquebert et Houriez [2000] proposent une approche mixte combinant la méthode KADS et une méthode de modélisation à partir d'un corpus KOD. Quant aux méthodes descendantes, Talbi et Laublet [2000] ont construit pour une classe de problèmes donnés une solution méthodologique à partir de l'outil OPENKADS.

Toutefois, la réutilisation de ces méthodes peut être partielle et se faire sur les spécifications d'éléments du domaine ou sur le raisonnement utilisé dans la méthode. Reynaud et Tort [2000]

déclarent : « … Appliquée aux éléments du domaine, cette technique (la réutilisation) repose sur la définition d'ontologies décrivant explicitement les éléments du domaine. Appliquée aux éléments du raisonnement, elle repose sur l'identification de descriptions abstraites de méthodes de résolution de problème. ». Dans cette étude, nous nous situons dans cette réutilisation de méthodes.

Comme Reynaud et Tort [2000] qui proposent de construire un modèle de résolution de problèmes génériques à partir d'ontologies du domaine spécifiées formellement, nous définissons une ontologie du domaine d'expertise à laquelle nous allons associer comme méthode de résolution le raisonnement à partir de cas.

3.3. Représentation des connaissances et ontologies

Construire une ontologie d'un domaine revient à modéliser ce domaine, c'est-à-dire à définir quels en sont les objets, à les décrire et à structurer les descriptions obtenues. Cela implique qu'un langage de description ait été précisé et qu'une structure d'accueil permette d'organiser la description. Lors de la réalisation d'une application, il faut passer d'une description informelle des objets du domaine à une description formelle [Biébow & Szulman, 2000]. La représentation des connaissances permettant de traiter le problème à résoudre doit être formalisée dans un langage formel ce qui est possible qu'à condition que le problème des ontologies soit résolu [Bachimont, 2000].

[Charlet et al., 1996] donne la définition suivante d'une ontologie «***faire une ontologie***, c'est décider des individus qui existent, des concepts et des propriétés qui les caractérisent et des relations qui les relient». Les relations les plus classiques sont les relations d'héritage, les relations de composition. Les autres relations associant les termes n'ont pas de sémantique implicite évidente. Schreiber et al. [1994] distingue l'ontologie du domaine qui contient les termes et expressions spécifiques d'un domaine applicatif, de l'ontologie du modèle qui décrit la structure imposée sur la connaissance du domaine, par la tâche et par la méthode de résolution de problème.

Une grande partie des travaux en ontologie se fait par une description lexicale des domaines de spécialités. Les textes comme la documentation technique sont la source d'acquisition des connaissances et concernent les chercheurs en linguistiques et en terminologie [Biébow & Szulman, 2000]. Ces travaux font partie des méthodes ascendantes de modélisation, que nous n'avons pas exploitées. Par contre nous nous sommes tournés vers la construction de l'ontologie du domaine d'expertise à partir de l'étude du processus de maintenance et de l'expertise associée.

Avant de présenter notre propre méthode de capitalisation des connaissances nous introduisons dans un bref résumé les différents formalismes de représentation des connaissances nécessaire pour les modéliser.

4. Formalismes de représentation des connaissances

Le formalisme de représentation de la connaissance donne la structure de cette connaissance telle qu'elle est manipulée par le système à base de connaissances. La représentation définit une série de symboles et une série d'opérations sur ces symboles qui modélisent le raisonnement que la représentation supporte. Les symboles ont un contenu sémantique mais les opérations qu'on leur associe sont déterminées par leur syntaxe.

Le formalisme de représentation des connaissances est donc un langage formel donné par l'interprétation sémantique de sa structure syntaxique [Biébow & Szulman, 2000]. Différentes techniques de représentation des connaissances ont été proposées dans le domaine de l'intelligence artificielle telles que les prédicats logiques, les systèmes à base de règles, les réseaux sémantiques, les schémas (frames) et la représentation orientée objet.

Formalisme logique

Le formalisme logique a été l'un des premiers formalismes proposés pour représenter la connaissance en intelligence artificielle [Kaufman, 1987]. Différents langages logiques ont été proposés comme la logique des propositions (définie par un alphabet, des règles de construction des phrases et un calcul de valeurs de vérité pour ces phrases appelées formules), la logique des prédicats du premier ordre (son alphabet inclut des symboles de fonctions), la logique floue (introduit degré dans la valeur de vérité d'une formule), modale (introduit des modalités telles que la possibilité ou nécessité, passé ou futur), linéaire (propositions jouent le rôle de ressources consommables), etc.

Systèmes à base de règles

L'élément de base de ces systèmes est la règle de production qui a la forme suivante :
SI <condition> ALORS <action>. La partie condition est exprimée par un prédicat logique qui correspond à une affirmation de la base de connaissances qui doit être vraie au moment de validation de la règle. Dans ce cas l'action, qui est la partie exécutable de la règle et exprime des ajouts ou modifications à faire, peut être déclenchée. Ce système comporte trois parties : une base de règles (la connaissance opératoire de l'expert), un contexte ou base de faits (l'état actuel du système) et un moteur d'inférence (contrôle les actions couplées à un module expliquant le raisonnement) [Rousseau, 1988].

Réseaux sémantiques

Un réseau sémantique est un graphe composé d'un ensemble de nœuds qui représentent des concepts d'entité, attribut, objet, événement, état, etc. et d'un ensemble d'arcs orientés, étiquetés, qui relient deux nœuds en représentant les relations binaires entre ces concepts. Nous pouvons

structurer la connaissance dans une hiérarchie de concepts par les relations sorte-de et est-un qui représente le lien de spécialisation de concepts. Un certain nombre de modes de raisonnement est mis en œuvre, comme l'inférence par héritage ainsi que le filtrage [Capponi, 1995]. Le formalisme de réseaux sémantiques est le premier à structurer la connaissance et à rendre visibles différentes relations existantes entre les objets, ainsi que la notion de distance entre deux concepts.

Graphes conceptuels

Les graphes conceptuels ont été proposés par Sowa [1984] qui a enrichi le concept des réseaux sémantiques par l'aspect du concept s'appuyant sur l'étude de la perception en psychologie. Le sens d'un concept se réduit à sa position relative par rapport aux autres concepts dans le réseau sémantique modélisant les connaissances générales du système. A la différence des réseaux sémantiques représentant plusieurs propositions et définitions de concepts dans un même réseau, les graphes conceptuels organisent les différents types d'éléments dans des structures différentes avec la hiérarchie de spécialisation des graphes. Cela permet de manipuler et de vérifier chaque structure par les règles de cohérence.

Schémas (frames)

Dans les années 60, Bartlett [1964] a présenté l'idée des paquets de connaissances que l'homme utilise dans son activité cognitive quotidienne pour représenter ses expériences précédentes pour interpréter une nouvelle connaissance. A la fin des années 70 cette idée est développée par Schank dans les modèles de représentation du langage naturel et Minsky dans le premier formalisme informatique. La théorie développée par Minsky présente la notion « frame » et signifie une structure dynamique remémorée qui doit être adaptée pour correspondre à la réalité d'une nouvelle situation rencontrée. Il s'agit donc de la représentation de connaissance pour servir de support de raisonnement sur le monde réel.

Représentation objet

La représentation par objet permet de décrire la connaissance sous forme d'objets eux-mêmes décrits par leurs caractéristiques. Un objet est une description d'un élément qui est constitué de deux parties – statique et dynamique. La partie statique concerne l'état de cet élément et l'ensemble de ces propriétés (des données et d'informations). La partie dynamique concerne son comportement (les méthodes ou procédures qui manipulent les données et informations).

Les objets collaborent entre eux de deux manières [Capponi, 1995]: plusieurs objets communiquent à l'aide d'envoi des messages, un objet fait référence à un autre objet au niveau de la description de son état. Les objets similaires dans leur caractérisation et comportement sont regroupés dans les classes ordonnées par la relation de spécialisation permettant de construire une hiérarchie des classes. Cette hiérarchie permet le mécanisme d'héritage qui consiste à récupérer pour une classe donnée les caractéristiques (l'état et les propriétés) et les méthodes définis dans sa

surclasse. Il existe différentes relations d'héritages plus ou moins spécialisées. Une classe permet également de créer des instances : représentations des objets concrets ou réels (instanciation), c'est-à-dire elle fournit à l'objet son interface et sa structure.

5. Notre démarche de capitalisation des connaissances

Notre démarche est destinée à mettre en place un ensemble d'outils d'aide à la décision destiné aux opérateurs de maintenance dans leurs tâches quotidiennes et intégré dans la plateforme d'e-maintenance Proteus. Le concept d'un système d'aide à la décision est basé sur l'expertise du domaine ce qui nécessite la modélisation cognitive du processus et de la connaissance utilisée par les opérateurs pendant leurs activités professionnelles.

Nous avons mis au point notre approche sur le cycle de capitalisation de Grundstein, et créé une mémoire d'entreprise complexe. Une étude globale du processus de maintenance, nous a permis de mettre en évidence quatre types de mémoires existantes, (technique, individuelle, organisationnelle, et de projet). Nous présentons une ontologie faisant le lien entre ces différentes mémoires. Toutefois, nous nous sommes intéressés aux deux premières mémoires techniques liées à l'équipement et aux systèmes d'aide au diagnostic et à la réparation sur une plateforme d'e-maintenance.

L'étude des autres mémoires demande à être approfondie et peut faire l'objet de thèses suivantes, ce qui permettrait d'offrir un service complet de systèmes interactifs d'aide à la décision en maintenance basé sur la connaissance experte du domaine. Nous proposons une démarche empruntée par les experts du domaine pour l'élaboration d'un modèle d'expertise lié au domaine de la maintenance industrielle. Ce modèle sera formalisé dans une représentation objet par un diagramme de classe en UML, et implémenté dans Protégé, un éditeur d'ontologie.

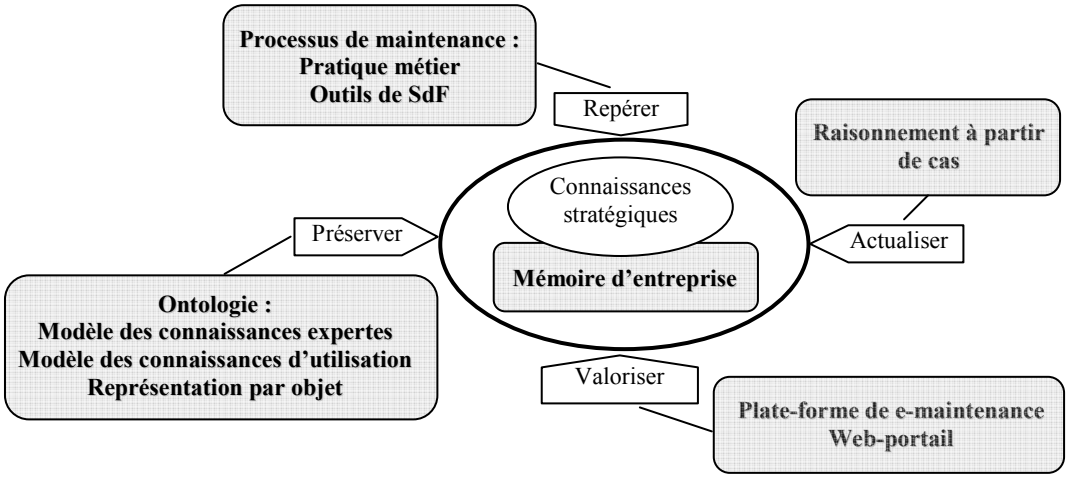


Figure 2.8. Cycle de capitalisation des connaissances

Nous avons associé à chaque phase de cycle de capitalisation des connaissances présenté dans [Grundstein, 2000] des méthodes appropriées pour le domaine de la maintenance comme le montre la figure 2.8.

Le repérage des connaissances stratégiques s'est fait par l'étude du processus de maintenance et est basé sur la pratique des gens du métier à CEGELEC Belfort et sur les différents outils liés à la sûreté de fonctionnement (AMDEC, arbres de défaillances, système SOA: Symptôme Origine Action).

La préservation des connaissances se fait par le développement d'une mémoire d'entreprise basée sur l'ontologie commune des connaissances expertes et des connaissances d'utilisation. Le formalisme de représentation est la représentation orientée objet.

La valorisation des connaissances se fait par le biais d'une plateforme distribuée de e-maintenance de technologie web développée par des partenaires informaticiens. L'échange et la diffusion des connaissances se font via un portail générique qui fait l'interface entre l'acteur de maintenance et notre outil interactif d'aide à la décision. Pour l'utiliser nous avons développé des web services liés à l'acquisition de la connaissance.

L'actualisation des connaissances se fait par la méthode choisie de résolution des problèmes.

5.1. Mémoire d'entreprise

Notre système prend appui sur l'expertise capitalisée extraite du terrain et de l'analyse faite dans le premier chapitre. Cette expertise en forme de connaissance est représentée dans une mémoire d'entreprise dont la structure rejoint la structure du processus de maintenance présentée dans le paragraphe 3.3. du premier chapitre. Cette structure permet aux différents systèmes d'aide à la décision potentiels de retrouver effectivement l'information dont l'utilisateur a besoin dans ses tâches de maintenance. Quatre couches de connaissances correspondent aux quatre processus techniques et manageriels dans le système de maintenance.

La mémoire est centrée sur l'équipement à maintenir, par un premier niveau de surveillance lié à l'équipement (la supervision des données mesurées à l'aide de capteurs) et des analyses de l'équipement, suivi d'un deuxième niveau élaborant le diagnostic de pannes et leur réparation. Ces deux couches de la mémoire couvrent la partie technique du processus de maintenance. La partie managerielle de processus est couverte par les couches de gestion des ressources humaines, matérielles et de gestion des stratégies de maintenance. Dans la gestion des ressources, la gestion des rapports d'intervention y est comprise ce qui alimente un calcul d'indicateurs devant servir à l'élaboration des différentes stratégies de maintenance à mettre en œuvre sur le matériel surveillé.

Dans chaque couche de la mémoire d'entreprise les différents types de données sont présents. Contrairement à la distinction des types de données utilisées dans le processus de

maintenance présentée dans le premier chapitre, nous dissocions cette-fois-ci deux types de données, les données concernant l'équipement individuel et les données générales concernant un parc d'équipement à maintenir dans le cadre d'un contrat de maintenance. Parallèlement, nous faisons la différence entre donnée, information, connaissance et compétence présentées à la figure 2.2. Nous illustrons la disposition des différents types de données dans la mémoire à la figure 2.9.

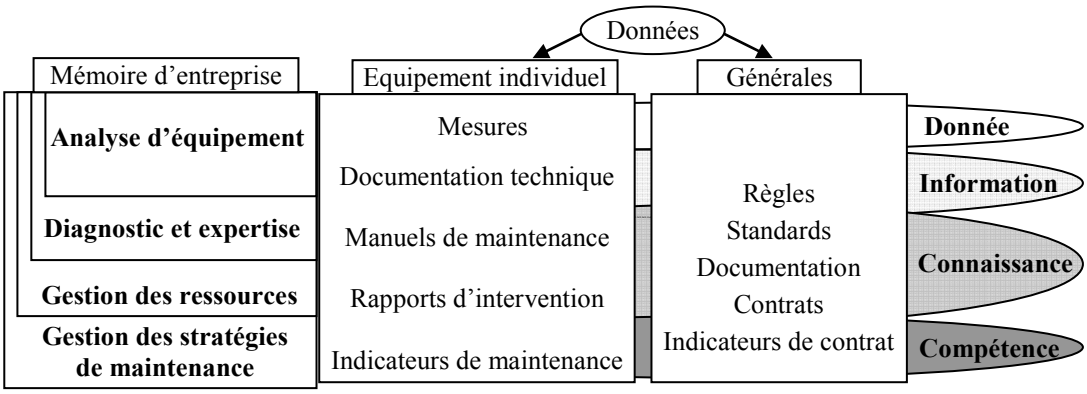


Figure 2.9. La structure de la mémoire d'entreprise et types de données

Dans la partie ***analyse d'équipement***, les opérateurs et les experts de maintenance doivent se servir des données et des informations techniques, des mesures des systèmes de surveillance ainsi que de la documentation technique fournie par le constructeur de l'équipement. Cette couche de la mémoire est de type technique (mémoire métier) et elle emploie des méthodes basées sur les documents. A partir de ces données et informations les experts de maintenance créent des documents contenant les connaissances (les manuels de réparation, les plans des interventions préventives, les gammes de maintenance, etc.) utilisées dans la deuxième couche de la mémoire.

La partie ***diagnostic et expertise*** fournit des informations et des connaissances aux systèmes d'aide à la décision ainsi qu'aux opérateurs et experts de maintenance. Les systèmes destinés au diagnostic et à la réparation ainsi que les systèmes de gestion d'e-documentation (contenant les manuels, les gammes, les plans de la maintenance préventive etc.) proposent l'aide aux opérateurs dans les interventions et les réparations des équipements. Après chaque intervention l'opérateur remplit un rapport d'intervention qui pourra être exploité pour le retour d'expérience et fournit des indicateurs pour la gestion des ressources ainsi que pour la gestion des stratégies de maintenance. Cette couche de mémoire est de type mémoire individuelle ou encore technique. Elle emploie des méthodes basées sur les documents ou sur les connaissances sous forme des cas.

La partie ***gestion des ressources*** comprend la gestion des ressources humaines et matérielles, la planification des interventions de la maintenance préventive et la gestion des rapports d'intervention. La mémoire à ce niveau est constituée de rapports d'intervention qui fournissent les indicateurs techniques et économiques, des documents concernant les règles, les standards, les

normes, l'organisation de l'entreprise, ses processus et ses activités. Les experts de maintenance créent et généralisent à ce niveau les connaissances et compétences sur les spécifications de la maintenance pour chaque type d'équipement.

Les systèmes d'aide à la décision destinés à la planification des ressources, la gestion des stocks, l'approvisionnement des ressources ainsi qu'à la gestion des rapports d'intervention se placent à ce niveau de la mémoire. Le caractère de cette partie de la mémoire est organisationnel et les méthodes pour sa création sont basées sur les connaissances.

La dernière couche de la mémoire concerne ***la gestion des stratégies de maintenance*** et globalement du contrat de maintenance. Les spécifications de maintenance pour chaque type d'équipement créées à partir des indicateurs des rapports d'intervention sont introduites dans les contrats de maintenance qui représente « des projets ». Cette partie de la mémoire est donc de type projet où la définition de projet, les activités, l'histoire, les évaluations et les expériences du projet sont enregistrés.

Nous proposons un premier modèle de ces différentes mémoires issues des quatre champs de maintenance mis en évidence au chapitre 1 au paragraphe 3.3. Ce modèle sera issu d'une représentation objet.

5.2. Méthode de représentation

Les méthodes de conception basées sur la représentation des connaissances orientée objet répondent mieux à l'identification des besoins des utilisateurs et offrent une architecture plus simple des systèmes d'informations qui sont en même temps facilement maintenables. De telles méthodes créent un ensemble de modèles et diagrammes selon le domaine d'application et la spécification de ses fonctions. Ces diagrammes proposent des perspectives multiples du système en cours de l'analyse ou du développement pour les utilisateurs aussi bien que pour des développeurs de logiciel.

Aamodt [2001] prouve que ***la représentation des connaissances orientée objet*** est bien adaptée à manipuler les entités composées et en particulier assure deux points de vue de la modélisation :

- ***Modélisation au niveau des symboles*** représente le niveau des structures de données et des langages de programmation où le comportement s'interprète par une séquence des procédures de programme. C'est le point de vue du développeur d'un système informatique.
- ***Modélisation au niveau de la connaissance*** se traduit comme la description du système informatique (physique). Le système est décrit comme un agent intelligent avec ses propres objectifs et la connaissance de comment les réaliser. C'est le point de vue de l'expert du domaine.

Bézivin [2000] souligne que les métamodèles au sens où ils sont utilisés à l'OMG (Object Management Group) répondent au concept de la représentation et plus particulièrement à la définition d'ontologie présentée dans [Charlet et al., 1996]. OMG a approuvé le système de notation ***UML*** (Unified Modeling Language) qui permet de spécifier, construire, visualiser et documenter les objets des systèmes informatiques ainsi que des systèmes non informatisés sous forme de diagrammes lisibles par les utilisateurs et les développeurs [Rumbaugh et al., 1999].

UML donne une vision complète de représentations des connaissances par ces diagrammes différents. Il existe neuf diagrammes permettant de modéliser les aspects structurels (ou statistiques), les aspects dynamiques (changements d'états et réponses aux messages venant des autres objets), et les aspects propres à la représentation des exigences fonctionnelles (diagramme des cas d'utilisation). Notamment le diagramme de classes et le digramme d'objets permettent de représenter une ontologie des connaissances du domaine. Les descriptions de ces diagrammes sont présentées en annexe B. Nous utilisons ainsi ces diagrammes afin de construire l'ontologie du domaine de maintenance.

5.2.1. Ontologie du processus de maintenance

A partir de l'étude de processus de maintenance, des concepts de sûreté de fonctionnement et de la pratique des experts de maintenance de Cegelec, nous avons développé une ontologie du domaine d'expertise de maintenance englobant le modèle d'équipement à maintenir associé aux composants du système d'aide à la décision sous la forme d'un diagramme de classes en UML.

Nous avons construit notre propre cadre afin que dès sa conception il tienne compte des différents champs d'application du processus de maintenance. Cette ontologie a été établie comme outil de partage de la connaissance entre les différents acteurs de la plateforme d'e-maintenance. L'ontologie du domaine, bien qu'établie indépendamment des méthodes de raisonnement a une structure qui dépend de la manière dont les connaissances acquises seront utilisées pour raisonner car les experts délivrent des connaissances adaptées à leur raisonnement.

Le modèle du domaine est composé de cinq parties correspondant à la fois à la structure de la mémoire d'entreprise et au processus de maintenance (cf. fig. 2.10). Il s'agit du système de surveillance, de l'expertise sur l'équipement, de la gestion des ressources, la gestion des interventions et enfin de la gestion des stratégies de maintenance.

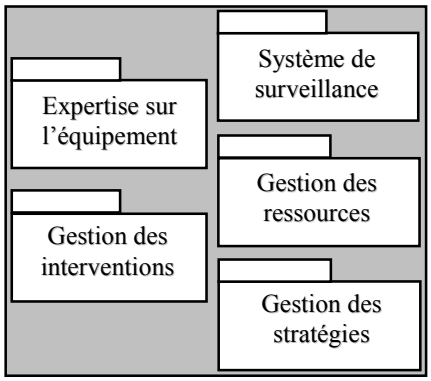


Figure 2.10. Les paquets du modèle général

Le modèle de l'équipement (cf. fig. 2.11) est caractérisé, d'une part, par les composants et sous composants de cet équipement sous une forme arborescente (*Component*). D'autre part, l'analyse fonctionnelle et le modèle associé (*Functional equipment model*) caractérisent le fonctionnement d'équipement par les classes *MainFunction* et *SecondFunction*. Elles représentent la fonction principale de l'équipement et les fonctions secondaires assurant le bon déroulement de la fonction principale.

Chaque équipement peut subir des pannes et des défaillances décrites dans la classe *Failure* et analysées dans l'analyse des défaillances (*Failure equipment model*). Une défaillance est identifiée par des symptômes (*Symptom*) provoqués par des origines (*Origin*) et remédiée avec une action de réparation (*Action*). Elle a également des caractéristiques (*Characteristics*) telles que la criticité, la fréquence d'apparition, la non-détectabilité et la gravité qui sont évaluées dans l'AMDEC.

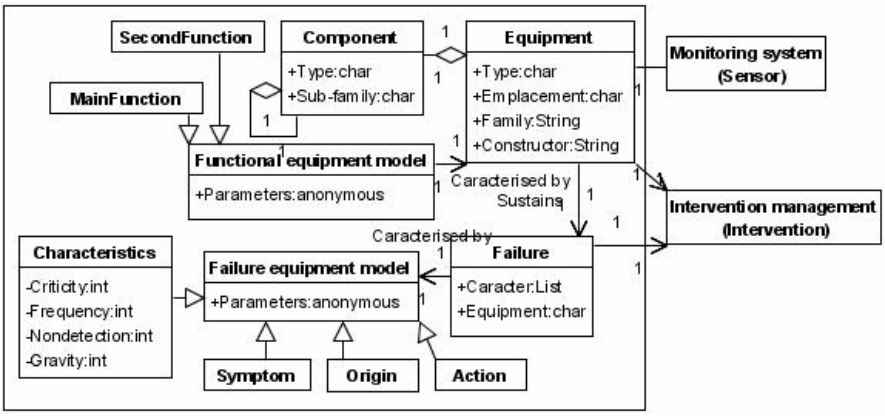


Figure 2.11. Modèle de l'expertise sur l'équipement

Le système de surveillance (cf. fig. 2.12) est constitué de capteurs (*Sensor*) installés sur les équipements et des différentes mesures (*Measure*) venant de ces capteurs. Un modèle d'acquisition des données (*Data acquisition model*) gère l'acquisition et l'exploitation de ces mesures. Ce

modèle peut déclencher la procédure de demande d'intervention en fonction d'un seuil des mesures et est donc relié avec le modèle de la gestion d'intervention.

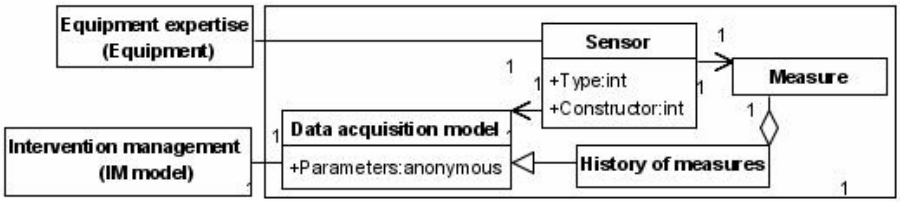


Figure 2.12. Modèle du système de surveillance

Le modèle de la gestion des interventions (cf. fig. 2.13) est centré sur l'intervention de maintenance. Une intervention (*Intervention*) doit remédier à la défaillance de l'équipement et est décrite par un rapport d'intervention (*Intervention report*) qui est composé d'observations techniques (*Observation*), d'indicateurs de la consommation (*Consumptions*) et de documents (*Document*) aidant l'opérateur à effectuer cette intervention.

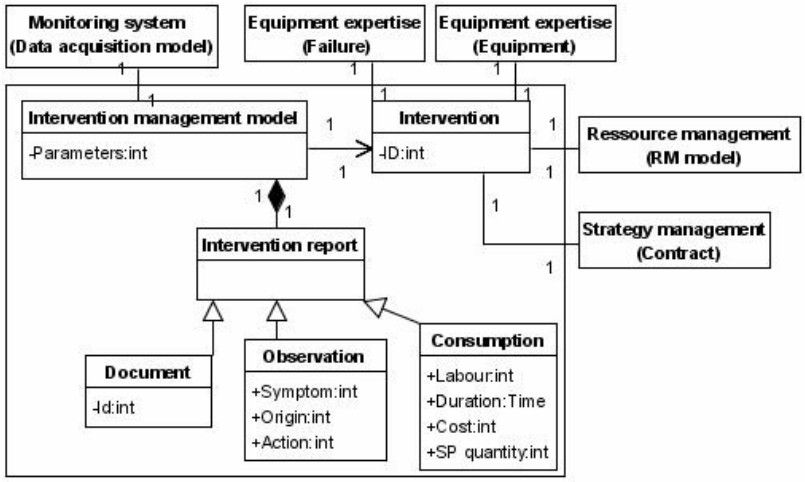


Figure 2.13. Modèle de la gestion des interventions

Le modèle de gestion des ressources (cf. fig. 2.14) décrit les ressources utilisées dans le système de maintenance, à savoir humaines et matérielles ainsi que leurs sous-classes l'opérateur (*Operator)*, l'expert (*Expert*), le manageur (*Manager*) et les outils (*Tool*) et les pièces de rechange (*Spare part*).

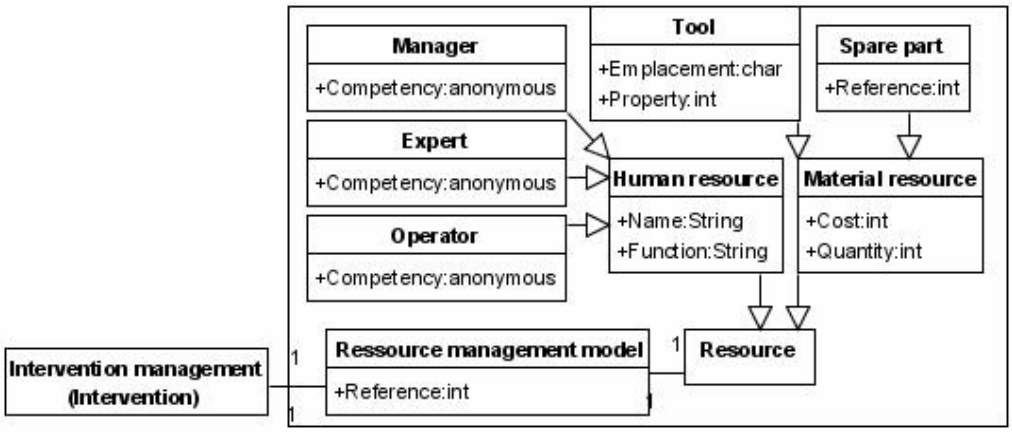


Figure 2.14. Modèle de la gestion des ressources

Le modèle de gestion des stratégies de maintenance (cf. fig. 2.15) est basé sur les indicateurs techniques (*Technical indicator*) et financiers (*Financial indicator*) concernant chaque équipement dans un contrat de maintenance (*Contract*).

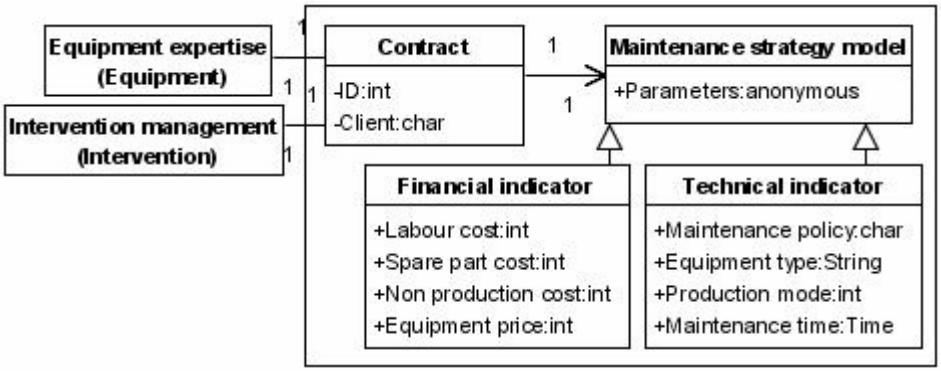


Figure 2.15. Modèle de la gestion des stratégies de maintenance

5.2.2. Modèle général du domaine

Le modèle complet général est montré à la figure 2.16. Nous présentons également un aperçu de l'ontologie commune développée dans le cadre de projet Proteus sur la plateforme d'e-maintenance à la figure 2.17. Cette ontologie est développée à l'aide d'un éditeur d'ontologie Protégé [http://protege.stanford.edu].

Protégé est un outil informatique intégré pour construire l'ontologie du domaine ainsi que pour adapter les formes d'acquisition des connaissances et pour développer les systèmes à base de connaissances. Les concepts concernant le domaine d'application sont spécifiés dans la classe GMOBasicElements (GMO comme General Maintenance Ontologie).

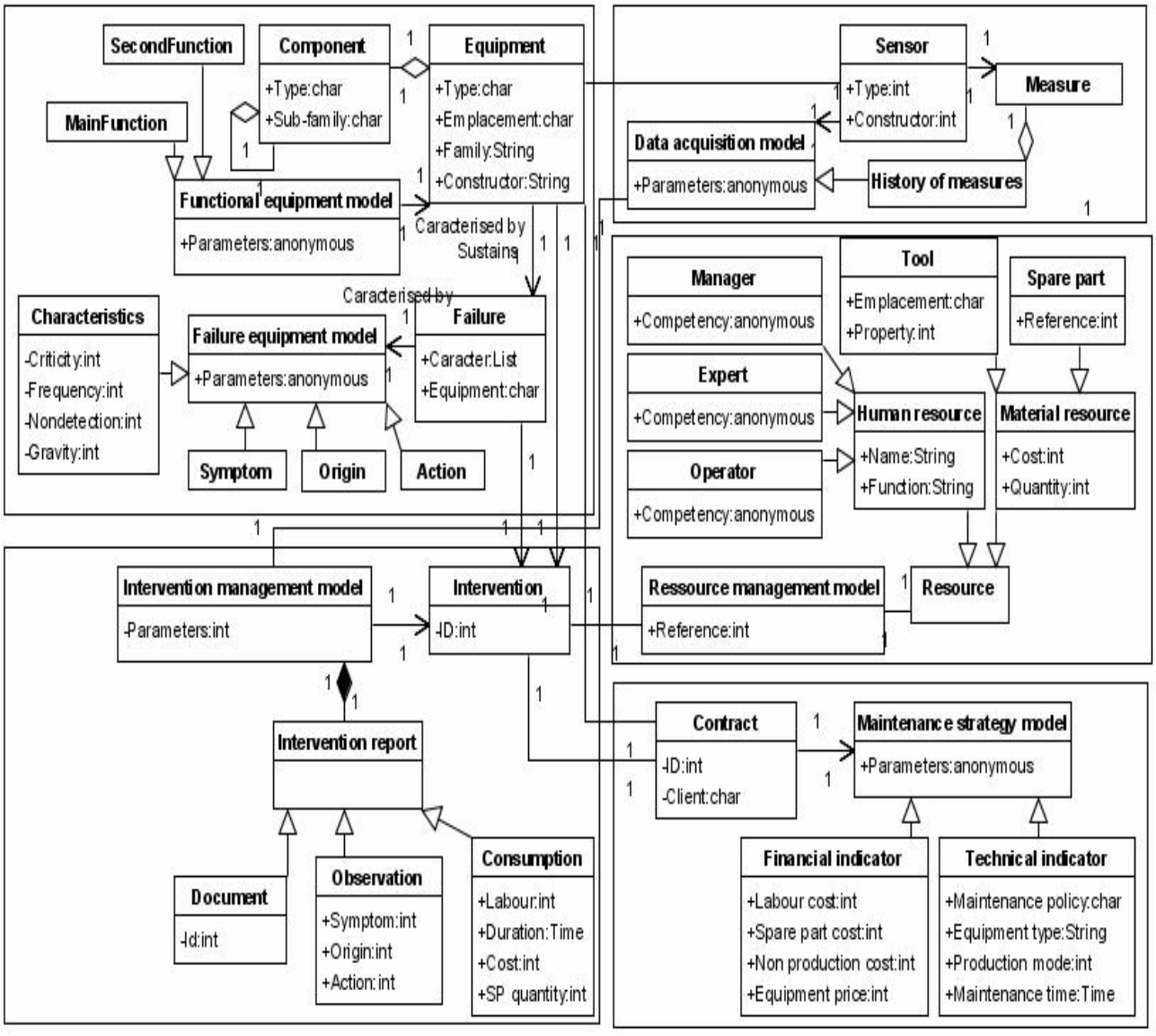


Figure 2.16. Modèle du domaine

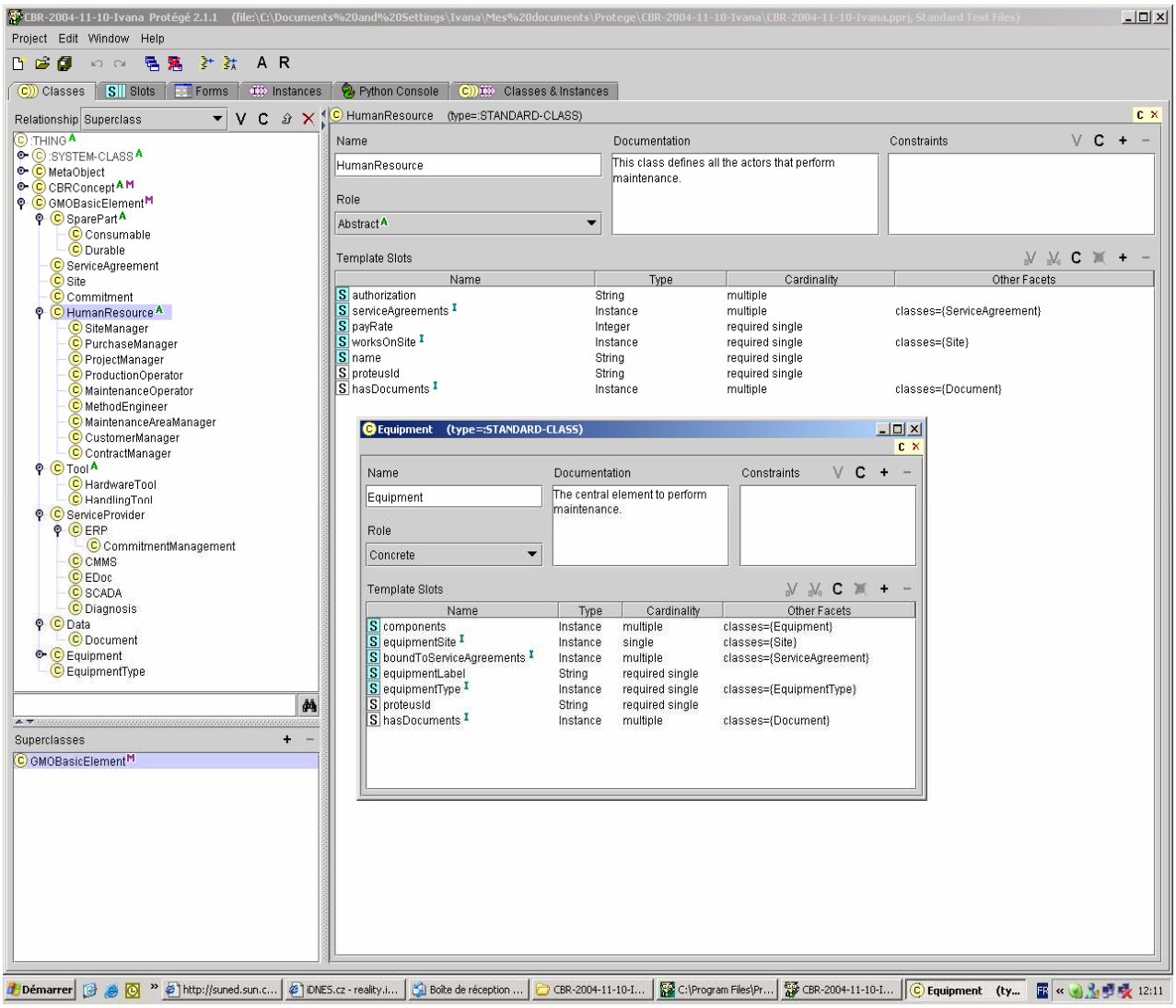


Figure 2.17. Un aperçu de l'ontologie dans Protégé

5.3. Méthode de raisonnement

Une première étude réalisée sur les systèmes d'aide à la décision à appliquer sur la plateforme d'e-maintenance nous a amené à chercher des méthodologies adaptées pour le développement de tels systèmes. Parmi les techniques utilisées dans le domaine spécifique au diagnostic et à la réparation, il existe différentes méthodes telles que les réseaux de neurones, les systèmes à base de règles, les systèmes utilisant les réseaux de Pétri, les graphes causaux et contextuels, les modèles Markoviens, les réseaux Bayésiens, les réseaux neuro-floues, le raisonnement à partir de cas, etc.

Toutefois ces méthodes ne sont pas couplées à des bases de connaissances, et n'utilisent pas une modélisation profonde des connaissances expertes, ce qui nous a amené à nous tourner vers d'autres types de systèmes en l'occurrence les systèmes de raisonnement à partir de cas.

Cette méthode permet également la gestion et la capitalisation des connaissances du domaine, point essentiel de notre système d'aide à la décision. Bien que les bases de connaissances soient tournées vers la connaissance de l'utilisateur, la dimension utilisateur est traitée indépendamment du reste du système et reste isolée des bases de connaissances et de la recherche associée [Perron, 2000].

Par contre, le raisonnement à partir de cas se présente comme proche du fonctionnement cognitif humain, en utilisant des cas analogues pour prendre une décision. Les systèmes du raisonnement à partir de cas sont bien adaptés à la coopération avec des utilisateurs humains. Ils sont donc censés fournir un support pour la résolution de problèmes et avoir une présentation conviviale et agréable à utiliser. La méthode de raisonnement à partir de cas est présentée dans le chapitre 3 de cette thèse.

6. Conclusion

Un système assurant l'aide aux opérateurs et ingénieurs dans leurs tâches quotidiennes représente dans un premier temps une économie considérable dans le budget de service de maintenance et dans un deuxième temps assure la transmission du savoir faire dans l'entreprise.

Afin de proposer des e-services intelligents en maintenance, nous avons étudié la connaissance experte, mise en jeu lors de toute activité de maintenance par les acteurs du domaine. Notre objectif pour être atteint passe par l'élaboration d'une mémoire d'entreprise destinée à des systèmes d'aide à la décision en maintenance. De ce projet trop ambitieux pour une seule thèse, nous avons donné une première ébauche d'une mémoire d'entreprise multi niveau, et nous avons développé plus spécifiquement une première pierre, à savoir l'élaboration d'un système d'aide au diagnostic et à la réparation.

La solution que nous avons adoptée n'est pas issue de méthodes dédiées à la mémoire d'entreprise mais est empruntée à l'ingénierie des connaissances. Nous avons fait pour cela un état de l'art des différentes méthodes utilisées dans le domaine, ce qui nous a permis de sélectionner le cycle de capitalisation de Grundstein et de l'appliquer au processus de maintenance et ainsi d'élaborer une mémoire d'entreprise. La démarche adoptée comporte une réutilisation de méthodes et utilise deux ontologies, une du domaine et une de manipulation. Dans ce chapitre, nous avons proposé une ontologie du domaine de maintenance, et nous aborderons dans le chapitre suivant les méthodes de manipulation des connaissances, à savoir le raisonnement à partir de cas.

Chapitre 3

Raisonnement à partir de cas en diagnostic industriel

1. Introduction 74

2. Approche de raisonnement à partir de cas 75

2.1. Historique 76

2.2. Principes fondamentaux 77

3. Méthodes utilisées dans les phases du cycle de RàPC 81

3.1. Conception du système et phase d'élaboration 81

3.2. Remémoration 84

3.3. Adaptation 86

3.4. Révision 87

3.5. Mémorisation 88

4. Domaines d'applications 89

4.1. Typologie d'applications 90

4.2. Notre domaine de diagnostic 92

5. Etude comparative des différents systèmes de diagnostic existants 94

5.1. Systèmes développés à l'aide d'un système de développement 94

5.2. Systèmes développés dans un système de représentation des connaissances 97

5.3. JColibri 101

5.4. Tableaux récapitulatifs 103

6. Nos choix et notre démarche 105

7. Conclusion 107

1. Introduction

Dans le cadre de la conception d'outils d'aide à la décision destinés à fonctionner sur une plateforme d'e-maintenance, nous avons développé au chapitre 2 une démarche de capitalisation des connaissances adaptée aux besoins du domaine de maintenance. Notre objectif se situe à deux niveaux :

- Le premier niveau est de mettre en œuvre un système de gestion des connaissances expertes sur la maintenance d'équipements clés permettant de préserver le savoir et le savoir faire de l'entreprise.
- Le deuxième niveau est de proposer un système d'aide à la résolution de problèmes de diagnostic et de réparation destiné aux acteurs de maintenance.

Une mémoire d'entreprise multi-niveaux permettant d'identifier les connaissances cruciales a été proposée. Elle a été modélisée par des diagrammes de classes UML et a permis ainsi de définir une ontologie du processus de maintenance qui pourra être consultée par les différents acteurs de la plateforme d'e-maintenance. En ce qui concerne le système d'aide au diagnostic et à la réparation, la démarche de capitalisation appliquée au chapitre 2 sur la première phase du cycle de capitalisation demande à être complétée par les phases de valorisation et d'actualisation du cycle. En effet, les connaissances doivent être manipulées afin de pouvoir être valorisées et actualisées, et ce chapitre 3 aborde la méthode choisie pour la manipulation de ces connaissances, le raisonnement à partir de cas (RàPC).

Le RàPC est une méthode d'apprentissage et de résolution de problèmes basée sur les expériences passées. Aujourd'hui cette technique s'apparente de plus en plus à une méthode de gestion des connaissances. Toutefois, cette méthode est généralement utilisée quand on a une faible connaissance du domaine étudié, que les problèmes ne sont pas complètement compris et l'expérience représente la plus grande partie des connaissances sur le sujet traité. Lors de la modélisation de la connaissance du domaine, les experts interviewés, loin de maîtriser leur domaine, ne s'expriment qu'en citant des exemples issus de leurs expériences. Les expériences identifiées sont représentées par des cas.

Un système informatique de RàPC met en œuvre une base de cas composée d'expériences passées dans laquelle peuvent être recherchées des expériences similaires au problème à résoudre. Les recherches actuelles dans ce domaine se focalisent sur une représentation approfondie des connaissances. En effet, on associe aux cas un ensemble de connaissances, qui permet de dire à certains auteurs que le RàPC est une technique d'implémentation des systèmes à base de connaissances [Althoff, 2001]. Cette méthode est également bien adaptée aux problèmes d'acquisition de nouvelles connaissances.

Notre étude passe, dans le paragraphe 2, par une présentation des principes fondamentaux concernant la structure d'un système de RàPC, son cycle de raisonnement et le principe associé à celui du carré d'analogie. Une deuxième vue sur le système de RàPC peut être décrite par les connaissances nécessaires au développement du système.

Nous introduisons, dans le paragraphe 3, les différentes méthodes traitant les données dans le cycle, et posons ainsi les jalons permettant de comparer différentes applications possibles. Après avoir abordé au paragraphe 4 les différents types d'applications possibles de RàPC, nous nous intéressons au domaine de diagnostic technique dont nous présentons les définitions.

Le paragraphe 5 est consacré à un état de l'art de différents systèmes de diagnostic basé sur le RàPC. En effet, dans le domaine de diagnostic technique de nombreux systèmes de RàPC ont été expérimentés et certains d'entre eux sont commercialisés et utilisés avec succès. Ces systèmes de diagnostic sont décrits puis comparés selon différents critères correspondant aux phases de RàPC. Deux tableaux récapitulatifs synthétisent cette analyse et permettent de dégager les avantages et inconvénients de chacun des systèmes présentés.

Fort de cette comparaison, nous présentons une démarche suivant le cycle de RàPC où un soin particulier a été mis dans la phase de création et d'élaboration des cas. Ce qui nous a permis de rejoindre la démarche de gestion des connaissances du chapitre 2.

2. Approche de raisonnement à partir de cas

Le RàPC touche principalement deux types de communautés : celles de l'intelligence artificielle et des sciences cognitives. La méthode se situe entre ces deux communautés de recherche et constitue un pont « naturel » entre elles [Mille, 1999]. En effet :

- ***Les sciences cognitives*** étudient la théorie de la compréhension, la résolution et l'apprentissage des problèmes par les hommes et essaient donc de modéliser les connaissances, les expériences et le raisonnement d'un être humain afin d'expliquer les motifs et le fonctionnement de son cerveau.
- Les systèmes ***d'intelligence artificielle*** à base de connaissance évoquent l'insuffisance des méthodes prenant appui sur la généralisation et essaient d'adopter le raisonnement humain et de l'apprendre aux machines automatiques artificielles. C'est une méthodologie pour développer des systèmes intelligents.

Récemment les deux communautés se rejoignent et les systèmes de raisonnement à partir de cas développés par les chercheurs de la communauté informatique de l'intelligence artificielle font souvent l'objet d'applications industrielles en gestion et capitalisation des connaissances dont les principes.

2.1. Historique

Le paradigme du RàPC doit son évolution à deux facteurs principaux :

- d'une part aux travaux en sciences cognitives sur la mémoire humaine, plus particulièrement sur la théorie de mémoire dynamique de R. Schanck,
- d'autre part aux évolutions engendrées par les limites rencontrées par les systèmes experts.

Cette recherche concerne les problèmes d'acquisition des connaissances « knowledge acquisition bottleneck », de mémorisation d'expériences passées et des limites dues à la connaissance exclusivement sous forme de règles.

Les bases du raisonnement à partir de cas ont été tracées par les travaux de Minsky et Schank à la fin des années 70. La théorie développée par Minsky présente la notion « frame (script, schéma) » qui correspond à une structure remémorée qui doit être adaptée pour correspondre à la réalité d'une nouvelle situation rencontrée.

Nous avons ainsi une représentation particulière des connaissances servant de support de raisonnement sur le monde réel. Dans son modèle de mémoire Minsky présente un cadre (frame, schéma) comme un réseau de nœuds et de relations entre ces nœuds. Toutefois l'auteur doute quant à la flexibilité du raisonnement logique et d'une représentation des connaissances ordinaires sous une forme synthétique de propositions indépendamment vraies.

R. Schank s'inspire des travaux de Minsky et formule, pour la première fois, le paradigme de raisonnement basé sur les cas. D'après Schank le processus de compréhension correspond à un processus d'explication qui s'applique d'une manière itérative [Mille, 1999]. Il présente la mémoire dynamique comme un réseau dense d'expériences au degré de généralité varié connu comme « Memory Organization Packets » (MOPS). R. Schank avec son équipe a travaillé à l'Université de Yale et est considéré comme l'auteur du terme « Case-Based Reasoning ». D'autres chercheurs ont suivi ses travaux et la tendance de recherche s'est orientée vers le développement des systèmes et leurs applications dans divers domaines.

La recherche dans le domaine du raisonnement à partir de cas s'est affirmée dans les années 80, d'abord aux Etats-Unis avec les conférences « DARPA » commencées en 1988 [Kolodner, 1988], avant de s'imposer en Europe avec la première conférence Européenne en 1993 à Kaiserslautern [Richter et al., 1993], puis avec la première conférence internationale à Lisbonne en 1995 [Veloso et al., 1995]. Janet Kolodner a popularisé cette méthode à un large publique dans son livre [Kolodner, 1993] et a développé le premier système de RàPC nommé CYRUS qui représente l'implémentation du modèle de la mémoire dynamique de R. Schank. Ce modèle a ultérieurement servi de base pour de nombreux systèmes, comme MEDIATOR [Simpson, 1985], CHEF [Hammond, 1986], PERSUADER [Sycara, 1987], JUDGE [Koton, 1988], JULIA [Hinrichs, 1988], PROTOS [Bareiss et al., 1993].

2.2. Principes fondamentaux

Dans ce paragraphe nous présentons les théories fondamentales concernant le fonctionnement et la structure d'un système de raisonnement à partir de cas. D'abord nous présentons le cycle de raisonnement à partir de cas fondé par les chercheurs A. Aamodt et E. Plaza [1994].

Ensuite nous introduisons le modèle de carré d'analogie présenté dans [Mille et al., 1996] qui montre le raisonnement analogique et représente le lien entre la description du cas et sa solution. Enfin nous nous intéressons aux connaissances nécessaires pour le développement d'un système de raisonnement à partir de cas.

Le cycle de raisonnement à partir de cas

L'objectif du raisonnement à partir de cas est de résoudre des problèmes courants à partir d'expériences passées. La méthode remémore les situations similaires passées et réutilise des informations et connaissances sur ces situations. Un système informatique de RàPC est un système particulier de gestion des connaissances. Il met en œuvre une base de cas composée d'expériences passées dans laquelle peuvent être recherchées des expériences similaires au problème à résoudre (problème courant).

L'utilisation d'un tel système répond à un cycle de raisonnement, le cycle du RàPC. Selon les sources bibliographiques, ce cycle est décomposé en 3, 4 ou 5 phases. [Fuchs et al., 2006] distingue les étapes de remémoration, adaptation et mémorisation. [Aamodt, 1994] propose de considérer également une phase de « révision » entre « l'adaptation » et la « mémorisation » d'un cas. [Mille, 1999] complète ce processus par une étape amont "d'élaboration" du cas cible (problème courant).

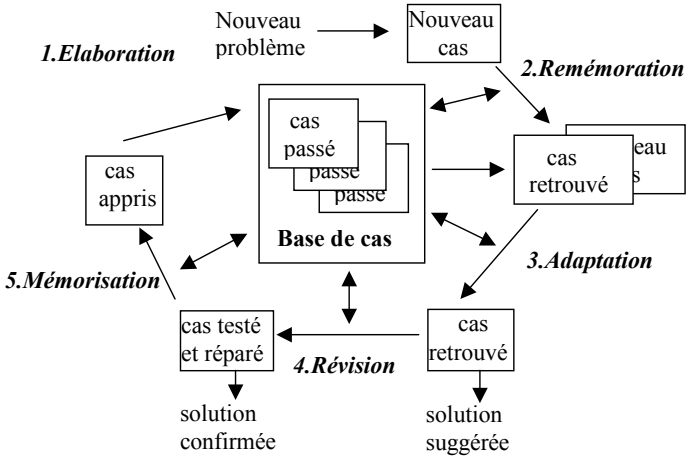


Figure 3.1. Le cycle de raisonnement à partir de cas [Mille, 1999]

1. ***L'élaboration*** d'un nouveau problème (cas cible) représente l'acquisition des informations connues sur le nouveau problème. Le cas doit être représenté d'une manière similaire à un cas source.

2. ***La remémoration*** des cas (sources) les plus similaires signifie la recherche des correspondances entre descripteurs des cas de la base et du cas à résoudre. Différentes techniques peuvent être utilisées : calcul d'un degré d'appariement des descripteurs (similarité entre deux cas), pondération éventuelle des descripteurs d'un cas, etc.

3. ***L'adaptation*** des cas afin de résoudre un nouveau problème se fait en réutilisant la solution d'un cas similaire.

4. ***La révision*** de la solution proposée signifie l'évaluation de la solution dans le monde réel (bilan d'un cas), la vérification par introspection de la base de cas en considérant la qualité des cas.

5. ***La mémorisation*** d'un nouveau cas représente l'ajout éventuel du cas cible dans la base de cas. Dans cette phase nous pouvons également synthétiser et apprendre des nouvelles connaissances.

Décomposition méthode – tâche

Armengol [1993] présente une approche en termes de tâches génériques. C'est une méthode plus fine pour la description du raisonnement à partir de cas qui propose une décomposition méthode-tâche. Les phases de raisonnement à partir de cas du cycle, à l'exception de l'élaboration, sont décomposées sur la figure 3.2 en fonction de la méthode utilisée pour accomplir la tâche que chaque phase de raisonnement à partir de cas doit assurer.

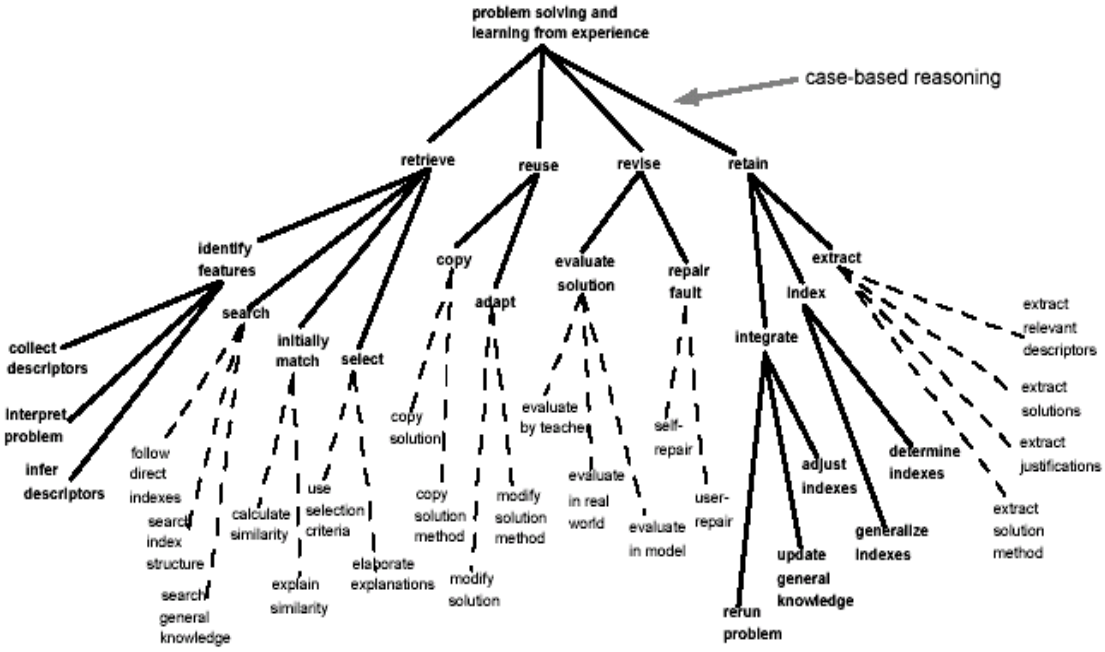


Figure 3.2. La décomposition méthode-tâche du RàPC [Aamodt & Plaza, 1994]

Les tâches ont des noms de nœuds écrits en « gras », alors que les méthodes sont écrites en caractères normaux. Les liens entre les nœuds de tâche (lignes pleines) sont les décompositions de tâche, c'est à dire les parties des relations. Les liens entre les tâches et les méthodes (lignes pointillées) identifient des méthodes alternatives pour la résolution de la tâche.

La tâche supérieure concerne la résolution des problèmes et apprentissage par expérience (« *problem solving and learning from experience* » en anglais) et la méthode pour accomplir cette tâche est le raisonnement à partir de cas (« *case-based reasoning* » en anglais).

Carré d'analogie

Le raisonnement de RàPC peut également être présenté par le carré d'analogie introduit dans [Mille et al., 1996]. Ce carré d'analogie explique :

- Dans un premier lieu le lien entre la description d'un cas et sa solution (la trace du raisonnement menant à la solution).
- Dans un deuxième lieu, les liens entre la description et la solution du cas source de la base de cas et du cas cible représentant un nouveau problème à résoudre (similarité entre deux problèmes). Dans ce cas nous parlons d'adaptation de la solution du cas cible en fonction de la similarité et de l'adaptation des descripteurs des cas sources similaires de la base de cas au cas cible.

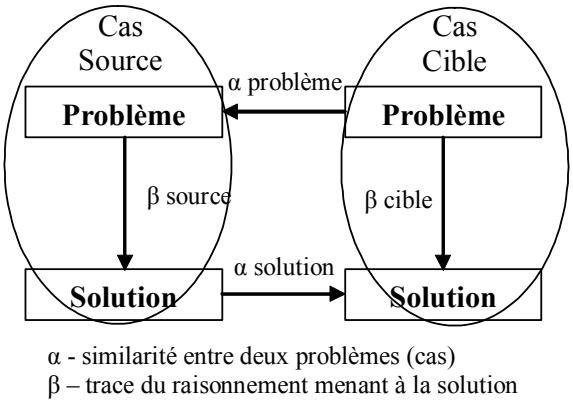


Figure 3.3. Carré d'analogie [Mille et al., 1996]

Un cas source est sélectionné et dit similaire à partir des valeurs de descripteurs du problème cible et en utilisant la mesure de similarité α. Ensuite les descripteurs solution qui doivent être adaptés car dépendant de descripteurs problèmes source, différents des descripteurs problèmes cible, sont mis en évidence par les relations de dépendance β entre valeurs de descripteurs problème cible et valeurs de descripteurs solution.

En d'autres termes, si une valeur de descripteur source dépend d'une valeur de descripteur problème, une modification de la valeur du descripteur problème entraînera une modification « analogue » à la dépendance du descripteur solution correspondant. Cette connaissance est nécessaire pour l'adaptation. En fonction de ces dépendances et des écarts α constatés à corriger,

l'adaptation permet de proposer une solution cible candidate qui pourra être vérifiée par rapport à sa conformité aux dépendances particulières qui pourraient exister entre problème et solution cible.

Conteneurs de connaissances

La mise en œuvre d'un outil d'aide à la décision basé sur le système de RàPC nécessite des connaissances particulières. [Richter, 1995] présente quatre conteneurs de connaissances utilisés dans un système de raisonnement à partir de cas : le vocabulaire, les mesures de similarité, la transformation de la solution et la base de cas (cf. fig. 3.4).

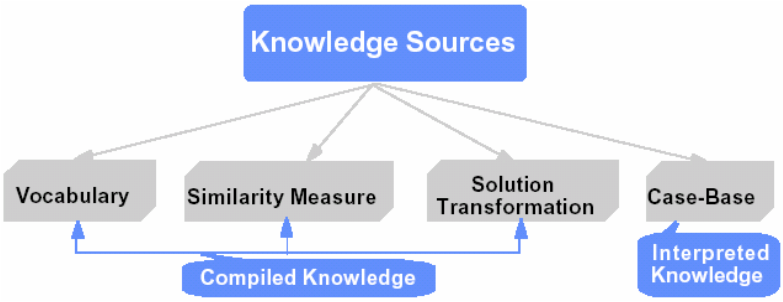


Figure 3.4. Les conteneurs de connaissances [Richter, 1995]

Les conteneurs relatifs aux mesures de similarité et à la transformation de la solution portent essentiellement sur l'expertise qu'il est par définition difficile à modéliser. Ces deux derniers conteneurs de connaissance, bien que spécifiés dans une phase amont de configuration de l'outil, peuvent être modifiés lors de l'exploitation du système. Ils traduisent en effet une certaine expertise sur les connaissances du domaine qu'il est nécessaire d'affiner au cours du temps. En principe, chaque containeur peut fournir pratiquement toute la connaissance disponible.

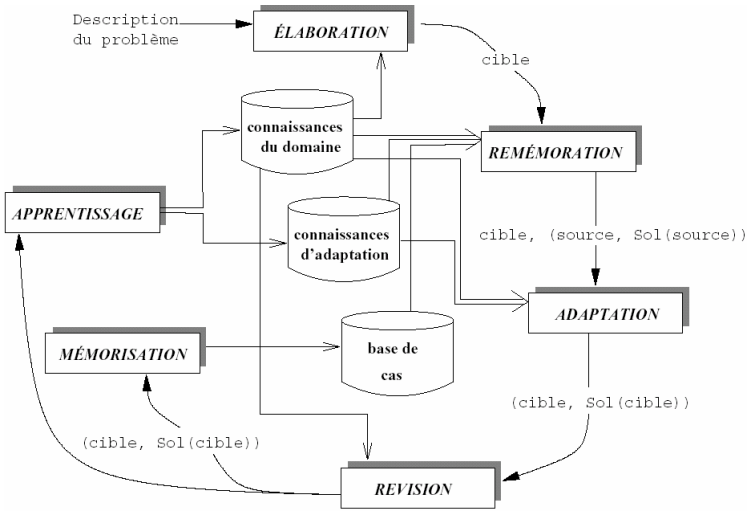


Figure 3.5. Les étapes de RàPC guidé par les connaissances [d'Aquin, 2005]

Lorsqu'il s'agit du raisonnement guidé par les connaissances, les connaissances utilisées dans la mise en œuvre du RàPC sont décrites au sein d'un modèle formel, dans un langage de représentation des connaissances. D'Aquin [2005] distingue trois types de connaissances : les cas, les connaissances du domaine et les connaissances d'adaptation.

Un modèle des connaissances du domaine est nécessaire afin de définir les notions relatives au domaine d'application d'un système de RàPC. Ce modèle est proche du terme vocabulaire présenté par Richter. Les connaissances d'adaptation correspondent aux connaissances relatives à la similarité entre problèmes et à l'adaptation des solutions utilisées par Richter.

3. Méthodes utilisées dans les phases du cycle de RàPC

Dans chaque phase de raisonnement à partir de cas, le traitement des données peut être de différentes natures selon le type de données et le modèle des connaissances adoptés. Aussi, dans chaque étape du RàPC, d'autres outils (d'intelligence artificielle notamment) peuvent être utilisés pour assurer ces traitements. Nous présentons dans ce chapitre un bref résumé des méthodes utilisées en fonction du domaine d'application concernant la maintenance et de la tâche de diagnostic.

3.1. Conception du système et phase d'élaboration

La conception du système se fait en amont de la « mise en service » du système RàPC et donc en amont des autres phases du cycle de RàPC. Elle inclut la structuration, la modélisation et la représentation du cas dans la base de cas ainsi que la mise en place des connaissances de similarité, d'adaptation, etc. Ce processus est souvent effectué manuellement par le concepteur du système à partir des documents, bases de données et autres ressources d'information sur le domaine d'application. Aux problèmes « d'élaboration de cas » s'ajoutent ensuite les problèmes éventuels d'indexations de cas dans la base, de stockage de cas dans les différents modèles de mémoire (hiérarchique ou plate) ou encore de l'exploitation de cette base de cas.

Phase élaboration

Une première formalisation de la phase d'élaboration est présenté dans [Fuchs et al., 2006] et définie comme « une étape qui consiste, à partir de l'entrée du système de RàPC, à construire le problème cible » :

$$\text{Elaboration} : \text{pré - cible} \rightarrow \textit{cible}$$

où $\textit{pré - cible}$ dénote l'entrée du système de RàPC, c'est-à-dire le problème cible avant mise en forme.

La phase d'élaboration concerne l'acquisition des informations sur le nouveau problème ce qui représente l'élaboration de sa description et se fait régulièrement au début du cycle d'utilisation du système RàPC. L'élaboration de la description du nouveau problème dépend essentiellement du modèle de raisonnement à partir de cas utilisé. Cela peut se concrétiser par l'envoi de requêtes automatiques vers d'autres systèmes de connaissances, par exemple, l'interrogation de l'utilisateur sur la description de la nouvelle situation ou une simple liste des caractéristiques ou symptômes de la nouvelle situation. Dans cette phase les méthodes d'acquisition des informations et des connaissances sont ainsi employées.

3.1.1. Représentation des cas

Un cas est une connaissance contextualisée qui représente une expérience. Les connaissances sont stockées sous forme de cas dans une base de cas qui peut donc être considérée comme une base de connaissances. Les cas dans la base de cas sont créés par la représentation et la modélisation approfondie des connaissances du domaine. La définition d'un cas dans la base de cas est basée sur les trois étapes suivantes [Mille, 1995] :

- La synthèse - trouver une structure permettant de satisfaire des spécifications.
- L'analyse - trouver le comportement résultant d'une structure particulière.
- L'évaluation - vérifier que le comportement est conforme à ce qui est attendu.

Un cas est généralement défini par une paire : [P, Sol(P)]) le problème et la solution associée. Un cas peut comprendre :

- la description du problème qui décrit les conditions dans lesquelles le cas apparaît,
- la solution du problème qui indique la solution dérivée de problème présent,
- la conclusion qui représente l'évaluation de la solution, des conséquences ou des commentaires ou encore l'explication qui relie la description et la solution.

Dans le cas de diagnostic, un cas décrit une situation diagnostiquée caractérisée par les symptômes observés et les valeurs mesurées. L'objectif est de trouver la cause de la défaillance (panne) et la stratégie de réparation.

Selon [Fuchs et al., 2006] le cas est la description informatique d'un épisode de résolution de problème. Souvent les problèmes en RàPC sont représentés sous la forme d'un ensemble de descripteurs. Un descripteur est une expression du formalisme de représentation. Cette représentation peut être dans le cas de diagnostic :

- sous forme de texte (description textuelle d'une défaillance puis de sa réparation).

- de type « attribut-valeur » où les attributs sont de différentes forme (numérique, symbolique, etc.). Les cas sont souvent représentés par les vecteurs (listes) des couples attribut-valeur qui peuvent être structurés (avec une structure hiérarchique) ou plats.
- sous forme d'objets stockés dans la mémoire hiérarchique.
- sous forme d'arbres de décision (cette structure de données est utilisée notamment pour la classification).
- dans d'autres cas et pour les tâches spéciales sous forme de contraintes, règles, graphes et plans.

Il existe de nombreux travaux sur la représentation des cas, mais la représentation la plus communément reprise dans le diagnostic est la représentation structurée en liste de descripteurs qui peuvent être des objets complexes. Dans [Bergmann et al., 2003] trois différents types de systèmes de RàPC par rapport à la représentation des cas sont distingués.

Dans ***les systèmes textuels*** [Lamontagne, 2002] les cas sont, soit non structurés sous forme de texte libre, soit semi structurés sous forme de texte découpé en plusieurs parties telles que « problème », « solution ». Il existe deux directions de travail dans ce type de système : la structuration de cas textuels qui concerne la représentation par un nombre limité d'indexes basés sur les caractéristiques du domaine (concepts, catégories, sujets, mots-clés) et l'extension du modèle de recherche d'information qui travaille sur l'élaboration de mécanismes de recherche sophistiqués avec des index les plus simples possibles (approche basée sur la fréquence de mots-clés). Des mesures de similarité sémantiques sont définies.

Dans ***les systèmes structurels*** [Richter, 1998] les cas sont représentés sous forme de paires <attribut, valeur>. Un attribut représente une caractéristique importante du domaine applicatif. La représentation des cas peut être sur un seul niveau (plate) ou sur plusieurs niveaux (hiérarchiques). Des mesures de similarité sont définies et mesurées en fonction de la distance entre les valeurs de mêmes attributs. Cette catégorie est la seule où les travaux sur l'adaptation de cas sont menés.

Dans ***les systèmes conversationnels*** [Aha et al., 1999] les cas sont représentés sous forme d'attributs qui sont spécifiés au fur et à mesure et l'ensemble de ces caractéristiques est connu comme le modèle du domaine. Dans le modèle structurel la description de cas doit être connue avant que la recherche des cas similaires commence. Contrairement à cela, le modèle conversationnel est interactif et emploie à côté de la description du problème et de la solution – action, une série de questions et de réponses pour compléter la description du problème. Le modèle conversationnel est actuellement le plus répandu parmi les applications commerciales mais il ne présente pas de phase d'adaptation car la solution de cas n'est pas structurée.

3.1.2. Organisation des cas

Pour organiser les cas dans la base de cas et afin de faciliter la remémoration des cas similaires nous sommes amenés à les indexer. Les méthodes d'indexation peuvent être manuelles ou automatisées. Le choix manuel des indices de cas suppose que l'objectif d'utilisation des cas et surtout des circonstances dans lesquelles les cas seront utiles soit déterminé précisément. Néanmoins les méthodes d'indexation sont de plus en plus automatisées et nous pouvons citer trois méthodes :

- ***Les méthodes d'indexation basées sur la similarité*** produisent l'ensemble des indices d'un cas abstrait issu des caractéristiques communes des cas ressemblants.
- ***L'indexation basée sur l'explication*** détermine les caractéristique appropriées pour chaque cas. Cette méthode analyse les cas afin de trouver les attributs prédictifs pouvant être utilisés comme des indexes.
- ***Les méthodes basées sur l'apprentissage inductive*** qui choisissent les attributs pouvant jouer le rôle des indexes par les algorithmes ID3, C4.5.

Les cas sont sources (des cas déjà résolus dans le passé) sont stockés dans une base appelée base de cas. Les cas sont mémorisés et organisés en fonction de critères bien déterminés permettant de les retrouver efficacement [Fuchs, 1997]. De plus, l'étape d'acquisition d'un nouveau cas permet de faire évoluer la connaissance. En fonction de la représentation du cas nous distinguons deux organisations principales de la base de cas :

- ***La mémoire plate*** où les cas sont organisés de manière linéaire. Cette organisation est la plus simple et les cas sont stockés dans une liste séquentielle.
- ***La mémoire hiérarchique*** où les cas sont structurés et organisés suivant une hiérarchie donnée.

L'organisation de la mémoire est très importante car le type d'organisation influencera les étapes du RàPC et notamment les étapes de recherche et d'apprentissage qui exploitent directement la mémoire. Un équilibre doit être trouvé entre les méthodes de stockage de cas qui préservent la richesse sémantique des cas et de leurs indexes et les méthodes qui simplifient l'accès et la récupération des cas appropriés [Mille et al., 1996].

3.2. Remémoration

La remémoration dépend essentiellement de la représentation de cas, de leur indexation et de leur organisation de la base de cas. Cette phase prend appui sur le calcul de similarité entre les cas de la base de cas appelés sources et le cas à résoudre appelé cible.

Mesures de similarité

Ces mesures de similarité cherchent des correspondances entre descripteurs de ces cas qui sont ensuite retrouvés à l'aide d'un algorithme de recherche. L'objectif de ces mesures de

similarité est de retrouver le cas de la base de cas similaire au problème actuel au sens qu'il est facilement adaptable à ce nouveau problème. Ainsi le degré de similarité est la fonction d'utilité/adaptabilité de la solution.

Il existe de nombreuses mesures qui dépendent de la représentation de cas. Nous distinguons les mesures :

- ***de similitudes*** qui rendent le degré de similarité entre deux cas.
- ***de dissimilitudes*** qui rendent le degré de différence entre deux cas.

Les mesures de similarité peuvent être :

- ***locales*** où les mesures sont établies au niveau des caractéristiques de cas. Les mesures les plus communes sont basées sur la notion de la distance. Elles dépendent du type de descripteur (numérique, symbolique, taxonomique). :
  - pour *les valeur numériques* : $sim(a,b) = 1 - \frac{|a-b|}{range}$
  - pour *les valeurs symboliques* : $sim(a,b) = \begin{cases} 1 poura = b \\ 0 poura \neq b \end{cases}$
  - pour *les taxonomiques* : $sim(a,b) = \frac{h(commonnode(a,b))}{\min(h(a),h(b))}$, où

  range est la valeur absolue de la différence entre la borne supérieure et la borne inférieure de l'ensemble des valeurs,
  h est le nombre de niveaux dans l'arbre taxonomique (dans l'hiérarchie).
- ***globales*** où les mesures sont établies au niveau de cas ou d'objet par agrégation de similarités locales. Le système calcule la similarité globale entre le cas cible et les différents cas sources de la base de cas. Aucune mesure de similarité n'est parfaitement appropriée à tous les domaines, donc différentes mesures sont utilisées, comme par exemple :
  - *weighted block-city* : $sim(A,B) = \sum_{i=1}^{n} \omega_i sim_i(a_i, b_i)$
  - *la mesure Euclidienne* : $sim(A,B) = \left[\frac{1}{n} \sum_{i=1}^{n} sim_i(a_i, b_i)^2 \right]^{1/2}$
  - de *Minkowski* : $sim(A,B) = \left[\frac{1}{n} \sum_{i=1}^{n} sim_i(a_i, b_i)^r \right]^{1/r}$, où

  n est le nombre d'attributs,
  ω_i est le poids (évalué en fonction de l'importance) de l'attribut *i,*
  sim_i est la similarité locale calculée pour l'attribut *i.*

Dans le cadre des ***similarités locales*** une distinction [Le Bozec et al., 2000] porte sur :

- la *similarité de surface* qui ne tient compte que des attributs des objets.
- *la similarité structurelle* qui s'intéresse aux relations entre attributs correspondants.

D'autres mesures de similarité tiennent compte des historiques et des séquences dans les cas ainsi que de l'adaptabilité des cas ou encore le coût de l'adaptation.

Algorithmes de recherche des cas similaires

Plusieurs algorithmes ont été mis en place pour rechercher des cas appropriés. Citons quelques exemples :

- ***Les K plus proches voisins*** (KPPV) est la méthode la plus habituellement utilisée. Cette approche implique l'évaluation de la similarité entre un cas cible (nouveau problème) et un cas sources de la base de cas. K est le nombre de cas sources voisins considérés d'être proches (autour) du cas cible.
- ***Les approches inductives*** déterminent les caractéristiques pertinentes dans la discrimination des cas et génèrent une structure arborescente des décisions afin d'organiser les cas dans la mémoire. Un exemple qui combine le RàPC avec l'induction est l'approche de KD-arbres basée sur les arbres de décision et développée spécialement pour les tâches de diagnostic et d'aide à la décision. Un autre exemple concerne les algorithmes d'apprentissage automatique ID3, C4.5.
- ***L'induction basée sur la connaissance*** utilise la connaissance dans le processus d'induction en identifiant manuellement les caractéristiques des cas.
- ***La recherche basée sur la structure*** (« template retrieval ») est similaire aux requêtes SQL où l'algorithme recherche des cas correspondant à certains paramètres.
- Les algorithmes appliqués à ***l'historique des séquences***.
- L'algorithme de ***chemin de similarité***.

Pour les grandes bases contenant un volume important de cas les structures indexées de la base de cas sont nécessaires. Différentes méthodes d'indexation sont utilisées comme réseaux de recherche de cas similaires (Case Retrieval Nets) présenté dans [Lenz, 1998], ou réseaux discriminatoires utilisés dans la représentation logique, etc. Un autre algorithme basé sur l'exploitation des différentes vues sur les cas - Fish & Shrink - combine dynamiquement différentes mesures de similarité pendant la recherche [Börner, 1998].

3.3. Adaptation

La phase d'adaptation permet de résoudre un nouveau problème par la réutilisation de la solution du cas source, dans le nouveau contexte propre au cas cible. Globalement, il y a trois types d'adaptation, à savoir l'adaptation par copie, l'adaptation manuelle et l'adaptation

automatique à l'aide d'algorithme, de formules, de règles, etc. Les principaux types d'adaptation automatique sont les suivants [Wilke & Bergmann, 1998]:

- ***L'adaptation transformationnelle*** suppose que nous avons des éléments de réponses à un problème qui nécessite des modifications selon des écarts de contexte observés entre cas source et cas cible. Cette transformation se fait à l'aide d'un ensemble de règles d'adaptation,
  - *adaptation par substitution* contient des modifications concernant le changement des valeurs d'attributs,
  - *adaptation structurelle* comprend un changement de la structure de la solution ou la réorganisation des éléments de la solution : ajout ou suppression de caractéristiques par exemple,
- ***L'adaptation générative*** part du fait que nous avons toutes les connaissances pour résoudre le problème à partir de zéro. Le cas retrouvé retrace le raisonnement ayant mené à la solution. La méthode détermine quelles parties de la trace de solution peuvent être réutilisées dans le contexte de la nouvelle situation,
- ***L'adaptation compositionnelle*** utilise l'ensemble des cas similaires pour l'adaptation.
- ***L'adaptation hiérarchique*** où les cas sont stockés à plusieurs niveaux d'abstraction.

D'autres méthodes d'adaptation existent telles que la remémoration guidée par l'adaptation [Smyth & Keane, 1996], l'adaptation mémoire [Leake et al., 1996], l'adaptation plan [Koehler, 1996] ou encore l'analogie par dérivation et l'analogie par transformation [Carbonell, 1986] et [Veloso, 1994].

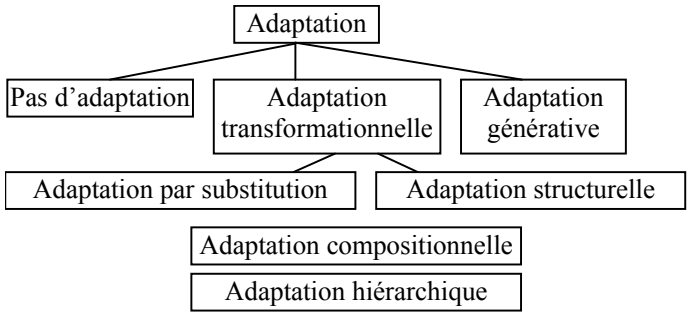


Figure 3.6. Types de l'adaptation dans les systèmes de RàPC

3.4. Révision

Dans la phase de révision nous pouvons employer différentes actions [Mille, 1999] :

- essayer la solution dans le monde réel.
- faire une rétrospection dans la base de cas en utilisant cette fois ci l'ensemble des descripteurs « problème + solution » pour vérifier que les cas similaires ont donné entière satisfaction.
- utiliser une autre méthode d'évaluation de la solution (simulateur, système expert classique, etc.).

La révision consiste à continuer éventuellement l'élaboration de la solution cible si besoin: Cette phase s'avère cruciale pour l'apprentissage du système et son évolution. Dans tous les cas, en cas de divergence entre ce qui a été construit comme solution et ce qu'il aurait fallu faire, il est important :

- de corriger de façon à ce que le cas soit un cas avec une solution correcte,
- éventuellement de tracer l'écart et l'expliquer sous forme d'exception (ce qui revient à particulariser une forme d'adaptation à une zone plus petite pour la classe de solutions)

Ce cas avec les explications de révision devient une source d'apprentissage importante pour faire évoluer les connaissances mobilisées par le raisonnement. Il n'existe pas encore de méthode standardisée pour rendre compte de la tâche de révision dans le RàPC.

3.5. Mémorisation

La phase de mémorisation d'un nouveau cas ajoute le nouveau cas appris dans la base de cas et synthétise des nouvelles connaissances afin d'optimiser la base de cas. La mémorisation du nouveau cas résolu est l'occasion d'enrichir la base de connaissances [Mille, 1999]:

- Un cas supplémentaire résolu augmente l'expérience du système.
- En cas de révision, la connaissance générale peut être modifiée et en particulier les connaissances duales liées aux tâches « retrouver » et « adapter ».
- La mesure de similarité peut être affinée pour éviter de sélectionner une classe de solution erronée.
- Les connaissances d'influences peuvent être affinées pour piloter l'adaptation. Ces connaissances d'influence sont duales des connaissances de similarité, elles sont directement liées aux poids utilisés pour pondérer la mesure globale de similarité.
- Les nouvelles dépendances peuvent être découvertes, etc.

Il peut être utile de garder la trace de l'ensemble du cycle avec le détail des corrections faites. Même si on n'a pas encore pu mettre à jour les connaissances du système, cette trace pourra être utilisée pour considérer ce cas comme un modèle pour « corriger » en s'inspirant de cette correction une nouvelle adaptation qui se ferait avec le même type de similarité (on découvrirait ce cas résolu et corrigé au moment de la révision).

Ces différentes tâches de la phase de mémorisation peuvent être résumées sous une tâche générale nommée : maintenance de la base de cas. La maintenance de la base de cas est définie comme étant le processus d'affinement de la base de cas d'un système de RàPC. Elle met en œuvre des politiques pour réviser l'organisation ou le contenu de la base de cas afin de faciliter le raisonnement futur pour un ensemble particulier d'objectifs de performance [Leake & Wilson,

1998]. Deux critères peuvent être employés pour la caractérisation et l'évaluation de la maintenance de base de cas :

- ***La performance*** du système mesurée par le temps nécessaire pour proposer une solution.
- ***La compétence*** mesurée par le nombre de problèmes différents résolus correctement par le système.

La plupart des approches proposées pour la maintenance de la base de cas s'orientent vers l'optimisation du contenu de la base de cas ce qui permet à la fois de réduire la taille de la base donc la performance du système et de préserver la compétence de ce système. Cette réduction peut s'effectuer par des suppressions de cas [Smyth & Keane, 1995] en catégorisant les cas dans la base de cas selon leurs compétences ou en considérant la performance du système [Leake & Wilson, 2000] ainsi que par des ajouts de cas [Smyth & McKenna, 1999], [Yang & Zhu, 2001] en construisant une nouvelle base de cas réduite.

D'autres méthodes portent uniquement sur la stratégie des mesures de performance [Leake & Wilson, 2000] et sont généralement basées sur le coût d'adaptation (dans la phase de réutilisation). Une autre stratégie [Yang & Wu, 2000] consiste à partitionner la base de cas en groupes de petites bases où la recherche de cas est moins coûteuse.

Nous avons introduit les principes et définitions de base afin de présenter l'étude comparative des différents systèmes de RàPC appliqués dans le domaine du diagnostic technique. Les différentes phases du cycle de RàPC doivent être en parfaite symbiose. Elles sont interdépendantes et le choix de la méthode d'une phase agit sur le choix des méthodes des autres phases. La conception du système de RàPC se place au centre du processus ce qui influence directement le choix des mesures de similarité, la technologie d'adaptation ainsi que la maintenance de la base de cas.

4. Domaines d'applications

Les systèmes de raisonnement à partir de cas peuvent être appliqués dans de nombreux domaines différents ainsi que sur les différents types de tâches à accomplir.

Parmi les domaines les plus développés nous pouvons citer la médecine, le commerce, les services de consultation, la maintenance, le contrôle, l'analyse financière.

Il existe différents types d'applications dont la classification, le diagnostic, la planification, le design, l'interprétation, la justification, l'aide à la décision, la recherche des documents, la configuration, la gestion des connaissances. Selon les domaines ou les types d'applications

étudiées les modèles et méthodes des systèmes de RàPC sont différents. Plusieurs auteurs ont présentés des typologies d'applications distinctes, en fonction de points de vue divers.

4.1. Typologie d'applications

Kolodner [1993] présente deux classes d'applications :

- ***les systèmes de résolution de problèmes*** incluant les tâches de planification, diagnostic, et design,
- ***les systèmes interprétatifs*** incluant les tâches de classification, évaluation, justification, interprétation.

Dans les systèmes de résolution de problèmes les solutions de nouveaux problèmes sont dérivées en utilisant les solutions déjà connues comme des guides. Dans les systèmes interprétatifs les nouvelles situations sont évaluées dans le contexte des situations déjà connues.

Althoff et al. [1995] proposent une autre décomposition dépendante du type de la tâche réalisée :

- les systèmes orientés vers ***la tâche de classification*** (prédiction, évaluation, help desk, diagnostic),
- les systèmes orientés vers ***les tâches de planification/synthèse*** (planification, configuration, design).

Une autre distinction présentée dans [Watson, 1994] porte sur ***les applications académiques et commerciales.***

Richter [2006] distingue d'une manière plus conventionnelle :

- ***les systèmes de résolution de problèmes*** (les tâches analytiques comme diagnostic, classification, aide à la décision, prévision),
- ***les systèmes de prise de décision*** (les tâches synthétiques comme design, planification, configuration).

Quant à Althoff [2001], il propose une typologie d'applications introduisant les différents systèmes de RàPC pour les applications de type ***classification, diagnostic, aide à la décision et gestion des connaissances.***

Le raisonnement à partir de cas est une méthodologie qui souvent implique l'utilisateur dans le développement de la solution. Les quatre types d'applications sont classés en fonction de l'engagement de l'utilisateur. Naturellement, plus l'utilisateur est engagé dans la résolution des problèmes plus les problèmes à résoudre peuvent être complexes tel que le montre la figure 3.7.

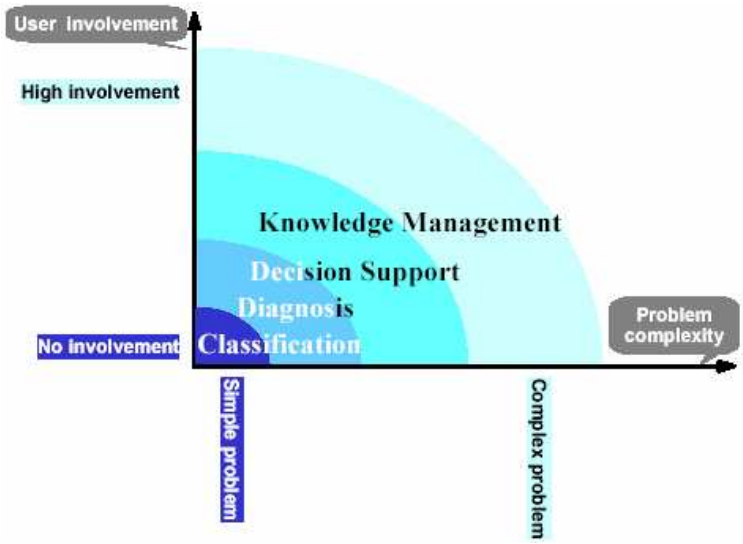


Figure 3.7. Quatre applications différentes de systèmes de RàPC [Althoff, 2001]

Le premier niveau de la hiérarchie est ***la classification*** (« classification ») où la solution du problème (donc le résultat de la classification) est issue de la sélection d'une ou plusieurs classes. Toutes les informations nécessaires doivent être disponibles pour le processus de tracement. Les exemples pour ce type d'application sont l'évaluation du risque, l'analyse des données ou encore l'estimation des coûts [Althoff, 2001].

Le deuxième niveau de la hiérarchie est ***le diagnostic*** (« diagnosis ») considéré comme une généralisation de la classification dans le sens où les informations nécessaires (les symptômes) ne sont pas nécessairement connues et doivent être recherchées dans un processus de diagnostic. Dans ce cas la connaissance générale du domaine est souvent nécessaire pour guider ce processus. Les systèmes de diagnostic technique ou médical peuvent servir d'exemples.

L'aide à la décision (« decision support ») représente le troisième niveau de la hiérarchie des systèmes de résolution de problèmes. Ce système est plus général que les deux précédents à cause de la distinction entre les symptômes et les solutions de problèmes qui n'est pas souvent directe et visible. Dans ce cas l'objectif de la classification est défini pendant le processus de résolution de problèmes. Des exemples sont la recherche d'une maison à acheter, d'un voyage de dernière minute, etc.

La gestion des connaissances (« knowledge management ») est une application beaucoup plus compliquée où aucune méthode de raisonnement basique ne peut être directement appliquée. Cette application demande l'intégration de différentes méthodes de raisonnement qui peut atteindre différents niveaux d'interactivité [Althoff, 2001].

4.2. Notre domaine de diagnostic

Nous nous intéressons aux systèmes d'aide au diagnostic et à la réparation des équipements industriels destiné aux opérateurs de maintenance. Dans ce paragraphe nous introduisons d'abord les notions de base liées au diagnostic technique.

Dans le cadre de la maintenance industrielle réalisée sur des systèmes de production, les biens à maintenir sont des équipements de production. Selon la norme [Afnor, 2001] on appelle ***un bien*** tout élément, composant, mécanisme, sous-système, unité fonctionnelle, équipement ou système qui peut être considéré individuellement.

Un composant est un élément discret d'un équipement. Il est interconnecté ou en interaction avec d'autres composants. C'est l'unité de description minimale de l'analyse systémique. Une observation est une information obtenue sur l'équipement.

Une panne est un état d'un bien inapte à accomplir une fonction requise, excluant l'inaptitude due à la maintenance préventive ou à d'autres actions programmées ou à un manque de ressources extérieures.

Lorsque nous distinguons les différents états de fonctionnement d'un équipement, nous parlons de ses ***modes de fonctionnement***.

Une défaillance est définie comme une cessation de l'aptitude d'un bien à accomplir une fonction requise. C'est donc un événement, à distinguer d'une panne qui est un état.

Il existe différentes définitions du ***diagnostic***.

[Dubuisson, 2001] le définit comme un problème de reconnaissance de forme. L'ensemble des états de fonctionnement d'un équipement est homologue à un ensemble de classes et le vecteur forme est le vecteur composant des paramètres observés sur le système.

Par contre [Piechowiak, 2003] intègre le diagnostic dans un processus plus global - la supervision qui comporte trois grandes fonctions : la détection, la localisation et la décision.

Selon [Zwingelstein, 1995] le diagnostic est l'identification de la cause probable de la (ou des) défaillance(s) à l'aide d'un raisonnement logique fondé sur un ensemble d'informations provenant d'une inspection, d'un contrôle ou d'un test.

Une définition générale et de référence du diagnostic de pannes est donnée par [Afnor, 2001] : ce sont les actions menées pour la détection de la panne, sa localisation et l'identification de la cause. Selon la même source, ***la localisation de panne*** représente les actions menées en vue d'identifier où se situe le fait générateur de la panne (la cause).

[Chandrasekaran, 1983] nous donne une définition plus opérationnelle : l'opération de diagnostic consiste à identifier une situation par un nœud spécifique dans une hiérarchie représentant le système diagnostiqué : les niveaux les plus élevés de cette hiérarchie correspondant aux concepts les plus généraux (niveau fonctionnel), les niveaux spécifiques (niveau des composants) se trouvant dans la partie inférieure.

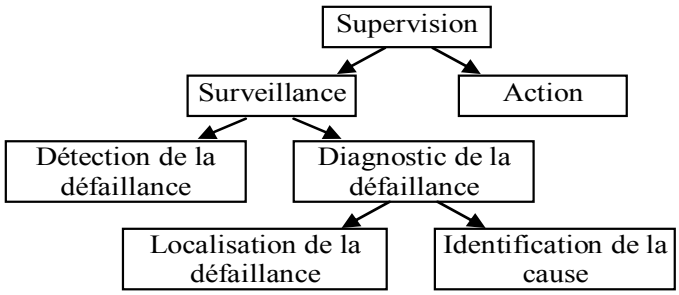


Figure 3.8. Diagnostic des pannes

Beaucoup de travaux ont été menés dans ce domaine en appliquant des méthodes très variées. Une classification de ces méthodes en trois groupes est faite dans [Toscano & Lyonnet, 2003] :

- La première classe de méthodes repose sur la connaissance d'un ***modèle mathématique du processus de diagnostic***. Les techniques utilisées se ramènent à un problème d'estimation paramétrique ou d'état.
- La deuxième classe de méthodes repose sur ***la connaissance des*** différents ***modes de fonctionnement du processus*** permettant de caractériser le ou les éléments défaillants. Les techniques utilisées se ramènent à un problème de classification.
- La troisième classe de méthodes repose sur ***les techniques d'intelligence artificielle***, elle implique l'utilisation d'un système expert.

Un bref résumé des méthodes d'intelligence artificielle utilisées dans le diagnostic technique est présenté dans [Piechowiak, 2003] concernant notamment les réseaux de neurones, les réseaux Bayésiens, les arbres de décision et le raisonnement à partir de cas. Cet aperçu n'est pas exhaustif, d'autres méthodes traitent du diagnostic technique telles que les systèmes à base de règles, les graphes causaux et contextuels, les modèles Markoviens, les réseaux neuro-floues.

Il existe plusieurs raisons désignant la méthode de raisonnement à partir de cas comme méthode bien adaptée au développement d'un système d'aide au diagnostic et à la réparation. La forme des cas présentés comme des expériences déjà rencontrées, et donc des exemples, facilite la compréhension de ces connaissances par les utilisateurs. Dans la phase d'élaboration, les algorithmes permettent de rechercher les cas même si le problème actuel n'est pas toujours complètement compris ou décrit. Le diagnostic fait aussi partie des domaines dont les problèmes se répètent et les solutions déjà trouvées et documentées peuvent être réutilisées.

Dans [Althoff et al., 1995] il existe deux types de systèmes de diagnostic basés sur le RàPC :

- Systèmes de type « **help desk** » caractérisés comme support de décision où l'accent est mis sur ce qu'il faut faire dans une situation donnée. Les applications de ce type sont utilisées dans les domaines où il existe un nombre important d'équipements techniques utilisés par des néophytes en maintenance.

- Systèmes de type **diagnostic général et réparation** caractérisés pour les domaines de diagnostic techniques des machines complexes ou de diagnostic médicale où l'accent est mis sur la recherche des causes précises des pannes et sur l'exploitation du problème. Les applications sont souvent utilisées par les experts du domaine en quête de plus amples informations.

5. Etude comparative des différents systèmes de diagnostic existants

Nous avons étudié sept systèmes de diagnostic technique.

- Il s'agit de trois *applications industrielles*, à savoir CHECKMATE qui étudie les imprimantes industrielles et CASSIOPEE et CASELINE qui s'occupent de dépannage de moteurs d'avion (le premier concernant les Boeing 737 et le deuxième touchant les Boeing 747-400).
- Ensuite quatre *applications universitaires* sont présentées : CREEK qui s'intéresse aux pannes de voiture, PATDEX qui traite de machines complexes, PAD'IM qui étudie la conception de systèmes de supervision dans le milieu industriel et MAIC la maintenance des équipements chimiques. Une présentation du système de développement JColibri est introduite dans la troisième partie.

Nous présentons dans la section suivante chacun de ces systèmes avec leur principe et les méthodes déployées. Afin de pouvoir comparer ces différents travaux, nous recenserons dans un tableau récapitulatif toutes les méthodes utilisées dans les différentes phases du RàPC. Ces tableaux permettent d'analyser ces travaux et d'en déduire les avantages et les inconvénients de chacun des systèmes présentés.

5.1. Systèmes développés à l'aide d'un système de développement

Trois systèmes ont été développés à l'aide d'un système de développement de raisonnement à partir de cas. Il s'agit des applications commerciales comme Cassiopée, CheckMate et CaseLine.

CASSIOPEE

Parmi les systèmes élaborés à l'aide d'un système de développement nous présentons l'application Cassiopée [Bergmann et al., 2003] implémentée dans le logiciel Kaidara (cf. fig. 3.9) et destinée au dépannage des moteurs d'avion Boeing 737 à CFM International. Dans la phase d'élaboration, les cas sont construits comme des ensembles d'objets décrits par les paires attribut-valeur. La structure d'objet est décrite par la classe d'objet identifiant l'ensemble des attributs et leur type (ensemble des valeurs possibles ou des sous-objets). Les classes sont rangées dans une hiérarchie représentant un arbre avec des relations d'héritage pour les sous-classes. Les arbres de

décision créés pendant la phase d'élaboration sont utilisés dans la recherche des cas similaires. Il s'agit d'un système structurel présentant quelques aspects du modèle conversationnel.

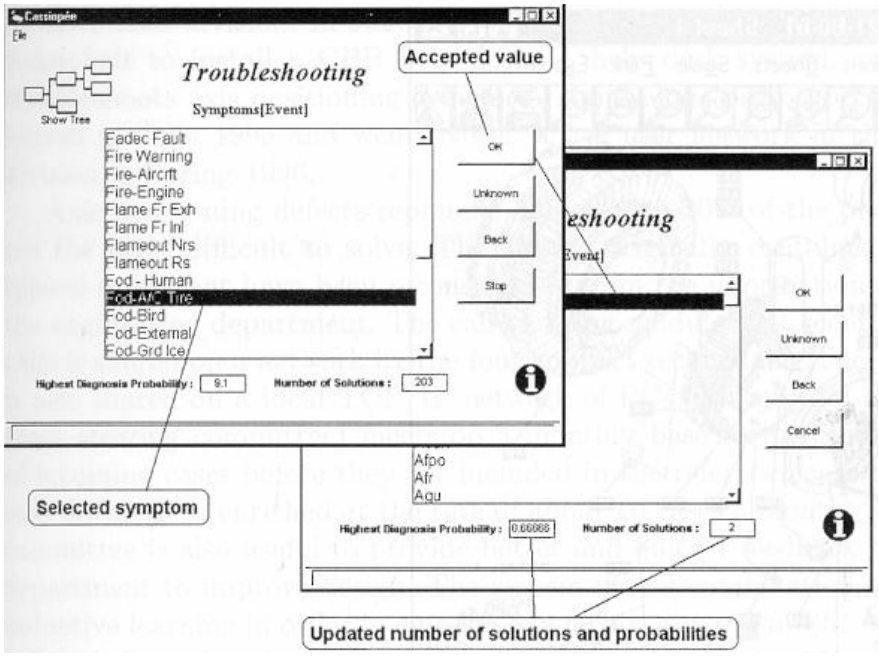


Figure 3.9. Le projet Cassiopée [Bergmann et al., 2003]

Cette approche dissocie la similarité locale pour chaque attribut et la similarité globale pour chaque classe d'objet définie par une fonction d'agrégation et par un modèle de poids. La similarité globale prend en compte les similarités inter et intra classe [Bergmann & Stahl, 1998]. Afin d'exécuter les phases de raisonnement, les connaissances générales nécessaires sont exprimées sous forme de trois types de règles différentes [Wilke & Bergmann, 1996]. Il s'agit des règles d'exclusion utilisées dans la phase d'élaboration du cas cible, des règles de complétion destinées à maintenir les cas dans la base de cas et les règles d'adaptation.

L'adaptation dans ce système est transformationnelle. Pour la maintenance de la base de cas, un nouveau cycle est introduit comprenant les phases de mémorisation et de affinement de la base. Elle est effectuée par l'administrateur du système et s'applique aux quatre containeurs de connaissances présentés dans [Althoff, 2001].

CHECKMATE

Le système CheckMate présenté dans [Grant et al., 1996] a été conçu pour le diagnostic de pannes et la recherche des documents relatifs (cf. fig. 3.10) aux imprimantes industrielles par Domino UK Ltd. C'est un système de RàPC de type help desk. Ce système est développé dans CasePoint qui est un module de l'outil de développement CBR Express destiné aux applications d'assistance technique et de diagnostic. Les fonctions principales du système sont la documentation ainsi que la distribution d'informations et d'expertise technique par un système automatique de diagnostic et de réparation.

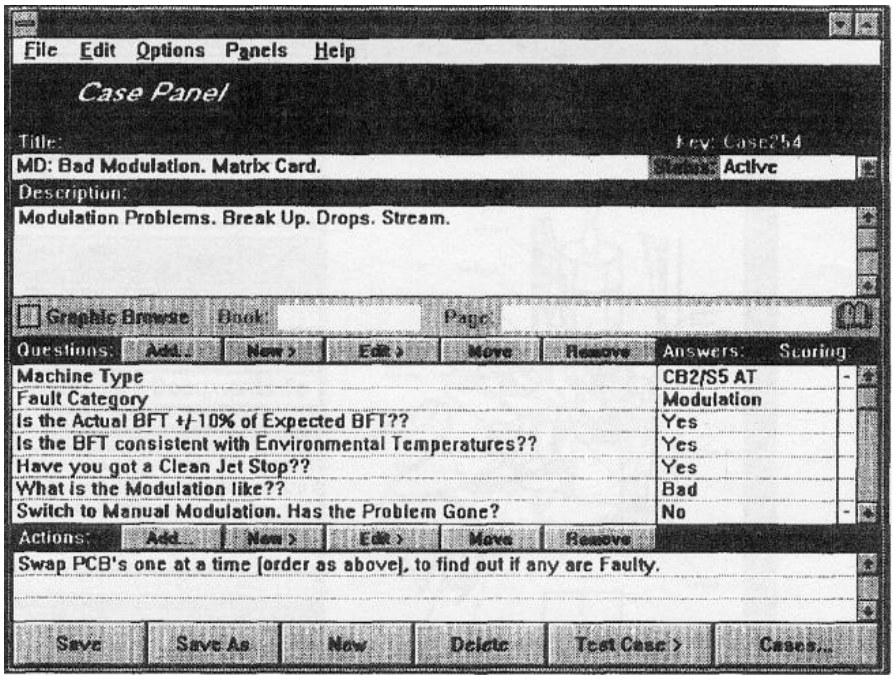


Figure 3.10. Le cas dans CBR Express [Grant et al., 1996]

C'est un système conversationnel où les cas sont composés d'une description textuelle du problème, de caractéristiques définies par l'utilisateur en répondant à des questions fermées, qui se traduiront en données symboliques, et de la solution du problème (donc l'action de réparation). L'indexation et la recherche des cas similaires sont basées sur les attributs sous forme de texte libre.

Le texte est coupé en petits ensembles (n-grammes) comparés avec les cas historiques coupés de la même façon. Cette recherche est ensuite assistée par des règles permettant de poser les questions dans la recherche avancée et dans la phase finale l'algorithme des K plus proches voisins est utilisé pour évaluer la similarité des cas. Après chaque question, le système recalcule pour chaque cas les similarités en fonction de la réponse à la question et les indique comme pourcentage de similarité.

Finalement le système propose comme solution le cas le plus similaire accompagné des documents techniques sont distribués pour l'action identifiée dans la solution de cas. La recherche des documents utilise la même base de cas. Le système ne propose pas de phase d'adaptation, ni de mémorisation ni de maintenance.

CASELINE

CaseLine [Watson & Marir, 1994] est un démonstrateur utilisé par British Airways pour le diagnostic (cf. fig. 3.11) des pannes et la réparation des moteurs de Boeing 747-400. Ce système a été implémenté à l'aide de l'outil de développement ReMind. C'est une application de type help desk. ReMind propose une représentation non structurée des cas soit sous forme d'une liste de couples attribut–valeur soit par représentation textuelle (CaseLine).

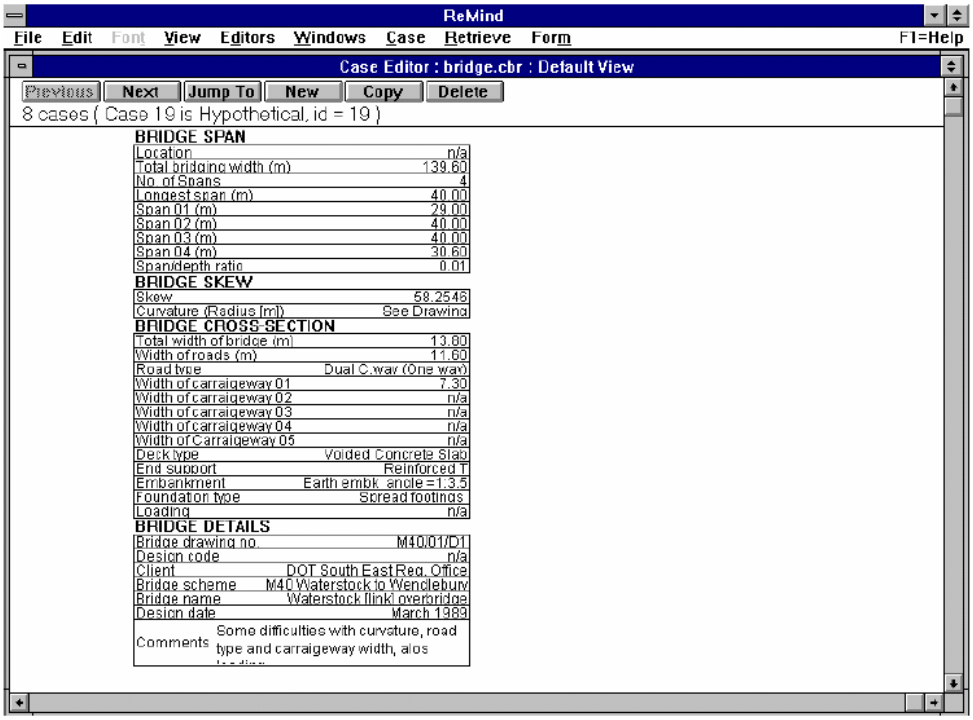


Figure 3.11. La représentation des cas dans ReMind [Watson & Marir, 1994]

Cette application utilise trois types de descriptions différentes pour la recherche de cas similaires basée sur la méthode des K plus proches voisins ou sur les arbres d'induction où chaque nœud est associé à un indice. Il s'agit soit d'un code sous forme de numéro ou de texte référant à une panne dans un manuel de maintenance, soit d'un texte décrivant la panne. Pour la recherche, une seule ou les trois descriptions peuvent être utilisées.

Le système ne supporte pas d'adaptation et aide seulement à identifier les procédures censées avoir le plus de succès. La base de cas contient environ 200 cas, par conséquent la maintenance s'avère peu importante. Les corrections des cas échouant se font manuellement.

5.2. Systèmes développés dans un système de représentation des connaissances

Trois autres systèmes ont été développés à l'aide des systèmes de représentation des connaissances. Ce sont les applications académiques comme CREEK, PATDEX, PAD'IM et MAIC mais souvent développées dans l'objectif de les faire commercialiser par la suite.

CREEK

Parmi les premiers systèmes de diagnostic technique développés en Europe au début des années 90 figure le système CREEK présenté dans [Aamodt, 2004]. L'auteur s'est inspiré des systèmes de diagnostic médical PROTOS et CASEY mais a appliqué son système sur le diagnostic des pannes de voiture. L'architecture de CREEK est basée sur un modèle – intégrateur de résolution de problèmes et d'apprentissage (cf. fig. 3.12). Un modèle général définit tous les

concepts – du diagnostic (symptômes), de l'apprentissage (généralisation des défaillances), du domaine (voiture) et du raisonnement (cas) à l'aide de la même structure de représentation. Les connaissances sont représentées à l'aide d'un système de représentation des connaissances basé sur des réseaux sémantiques et développé à l'occasion de la conception de CREEK.

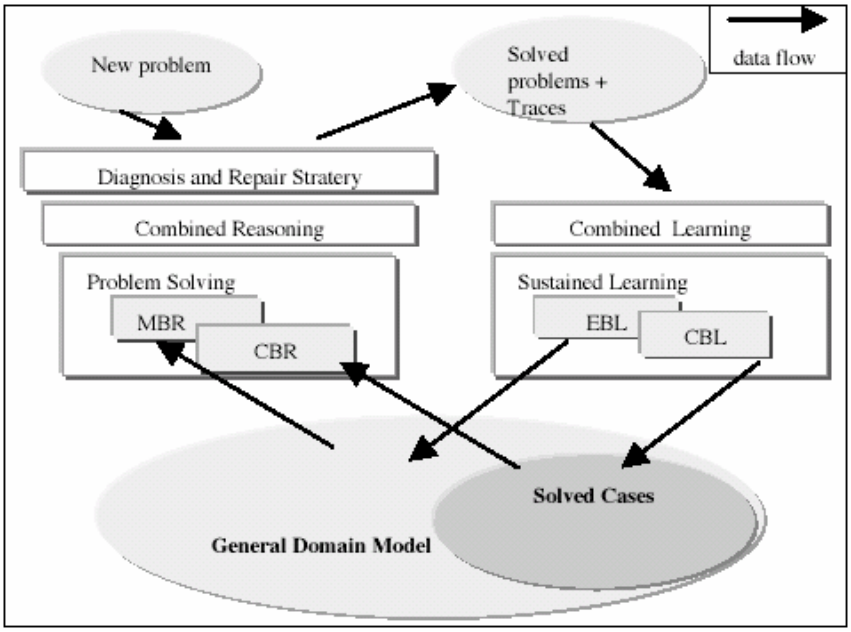


Figure 3.12. L'architecture fonctionnelle de CREEK [Aamodt, 2004]

Dans ce modèle structurel le cas est représenté par une liste de descripteurs (concepts), constitués par des triplets entité-relation-valeur ou des sous concepts. La recherche des cas similaires est basée sur la similarité sémantique qui prend en compte la connaissance générale du domaine et qui permet d'orienter la recherche vers les cas adaptables au nouveau problème. Il utilise des explications pour justifier la pertinence, la cohérence et la facilité de réutilisation des cas dans un nouveau contexte.

Ces explications sont ensuite exploitées par l'adaptation si des modifications s'avèrent nécessaires. Le système dans la phase d'adaptation se souvient de la trace du raisonnement et l'utilise pour l'apprentissage des nouveaux cas et pour la généralisation des attributs de cas.

PATDEX

Un autre système PATDEX dédié au diagnostic technique des machines complexes a été développé par Richter [Richter, 1991] à l'Université de Kaiserslautern (cf. fig. 3.15). Il a été intégré dans un système MOLTKE développé dans cette même université proposant des modèles de connaissances dans les systèmes experts pour les CNC (Computerized Numerical Control), machines à contrôle numérique. Le principe de PATDEX est basé sur l'intégration du raisonnement à partir de cas utilisant la méthode des K plus proches voisins et les arbres de décision.

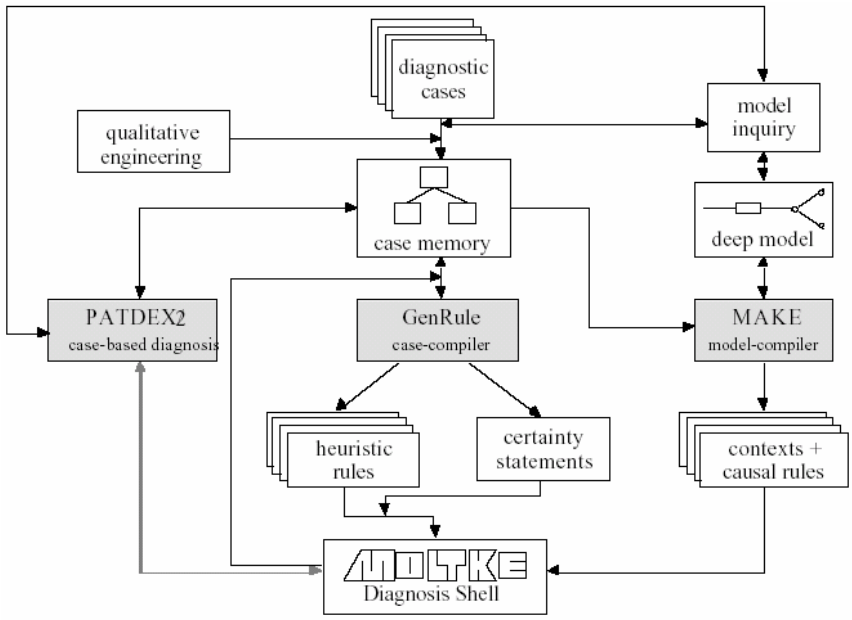


Figure 3.13. La méthodologie du projet INRECA [Richter, 1991]

C'est un système conversationnel parce que la description du cas cible se fait en excluant progressivement des cas après chaque nouvelle estimation de la similarité entre le cas cible et les cas sources. Les mesures de similarité utilisées tiennent compte de l'incertitude et de l'incomplétude de l'information dans la description du cas cible.

Dans la phase d'adaptation, le système utilise les graphes d'expériences et les compare avec les cas similaires trouvés à l'aide des mesures de similarité. Il s'agit d'apprentissage analogique où la connaissance du domaine est exploitée. La maintenance de la base de cas est assurée par le système MOLTKE. Ce système a été à la base des projets INRECA I et II qui ont apporté plusieurs contributions. Il s'agit du développement d'un langage orienté objet CASUEL pour la représentation des cas, d'une méthodologie de construction des systèmes qui a initié la création d'outils de développement des systèmes de RàPC (par exemple Kaidara) et l'implémentation de nombreuses applications industrielles décrites dans [Bergmann, 2001].

PAD'IM

Le système PAD'IM présenté dans [Mille, 1995] a eu pour objectif de faciliter la conception des systèmes de supervision dans le milieu industriel. Le système aide l'opérateur à choisir un nouvel environnement (différents tableaux de bord affichés sur les écrans de contrôle) après un changement de situation (cf. fig. 3.14). Nous comptons ce système parmi les modèles structurels. Les fonctionnalités de PAD'IM ont été implantées à l'aide de l'outil de développement Rocade [Fuchs, 1997].

Rocade est un système de représentation des connaissances par objet dont les fonctionnalités ont été spécialisées pour le raisonnement à partir de cas. La connaissance dans Rocade est décrite

sous forme de triplets – objet, attribut, facette - regroupés en classes. Les classes et les objets peuvent être structurés hiérarchiquement ce qui permet d'exploiter plusieurs mécanismes d'inférence, à savoir l'héritage multiple, l'inférence de valeur et la classification. Les similarités dans le système Rocade sont conceptuelles.

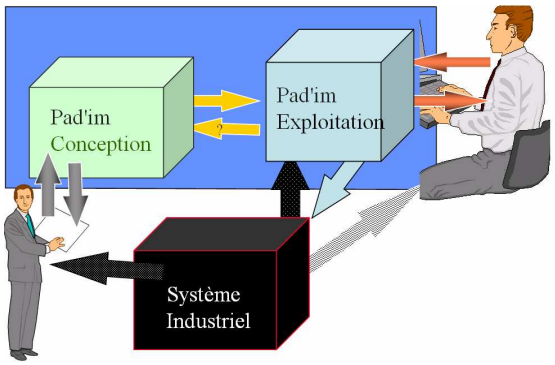


Figure 3.14. Deux facettes de système PAD'IM [Fuchs, 1997]

Dans le système PAD'IM, un cas est un épisode de supervision généralement composé d'un descriptif du problème et de la solution à proposer. Il est décrit en termes d'objets de supervision et de connaissances mobilisées. Pour les différentes phases de raisonnement, des connaissances générales sont nécessaires telles que les objets de supervision (composants, entités topologiques, produit industriel, etc.), les tableaux de bord, les actions de l'opérateur. Pendant l'élaboration du cas, un descriptif est créé en collectant les informations et l'opérateur désigne un objet focal de supervision.

Dans la phase de remémoration, une recherche indexée dans la mémoire de cas est faite en deux phases. La première permet de sélectionner un sous-ensemble de cas dont le contexte général est compatible avec celui de la situation courante. La deuxième évalue la ressemblance des épisodes discriminés à l'aide de deux mesures de similarité, une similarité conceptuelle et une dissimilarité événementielle. L'algorithme des K plus proches voisins est utilisé pour la recherche du cas le plus proche, qui est ensuite adapté en substituant les objets jouant des rôles semblables et en commentant la présence d'objets non similaires.

L'adaptation substitutionnelle est nécessaire pour chaque cas parce qu'une situation ne se reproduit jamais. Nous n'avons pas trouvé d'informations sur la maintenabilité de ce système et donc nous ne pouvons pas discuter l'aspect de la maintenance de la base de cas.

MAIC

MAIC [Pieri et al., 2001] est un système de maintenance traitant les aspects techniques et économiques sur l'étude et le choix de matériaux de construction (cf. fig. 3.15) pour le design et la maintenance d'équipements chimiques. Ce système utilise trois bases de données concernant les informations sur la corrosion, les matériaux et les coûts, et une base de cas concernant la

maintenance. Comme dans les autres systèmes structurels, un cas est représenté sous forme d'une liste d'attributs qui peuvent être de deux types : quantitatif et qualitatif.

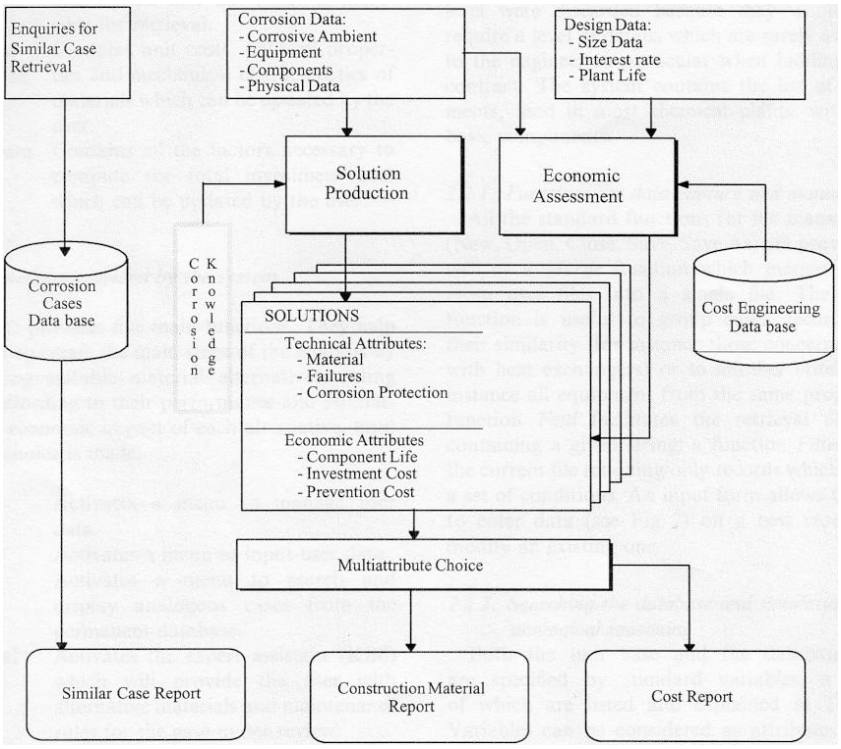


Figure 3.15. L'architecture de MAIC [Pieri et al., 2001]

Dans la phase de remémoration, les mesures de similarités sont basées sur le calcul de distance. Le système est développé dans OPTRANS – l'environnement de développement pour les systèmes d'aide à la décision basé sur un nombre limité de classes d'objet contenant les connaissances. Cet environnement couple le raisonnement à partir de cas, le raisonnement inductif et la simulation. Dans un premier temps, le système fait de la prévision de défaillances possibles causées par la corrosion. Dans un deuxième temps, il propose les mesures de protection afin d'éviter la corrosion et, finalement, il assure le support pour la maintenance.

Le raisonnement à partir de cas fonctionne comme le moteur de recherche des cas similaires dans la base de cas et ne supporte pas d'adaptation. Le système contient plusieurs fonctions basées sur la connaissance pour supporter l'évaluation économique de la solution techniquement acceptable.

5.3. JColibri

Il existe également un outil de développement des systèmes de RàPC nommé JColibri [Bello-Thomas et al., 2004] basé sur la représentation des connaissances par objet (cf. fig. 3.16). Cet outil n'a toutefois pas été utilisé pour des applications de diagnostic. Nous en faisons mention pour ses potentialités nombreuses comme par exemple le choix libre de la structure de cas.

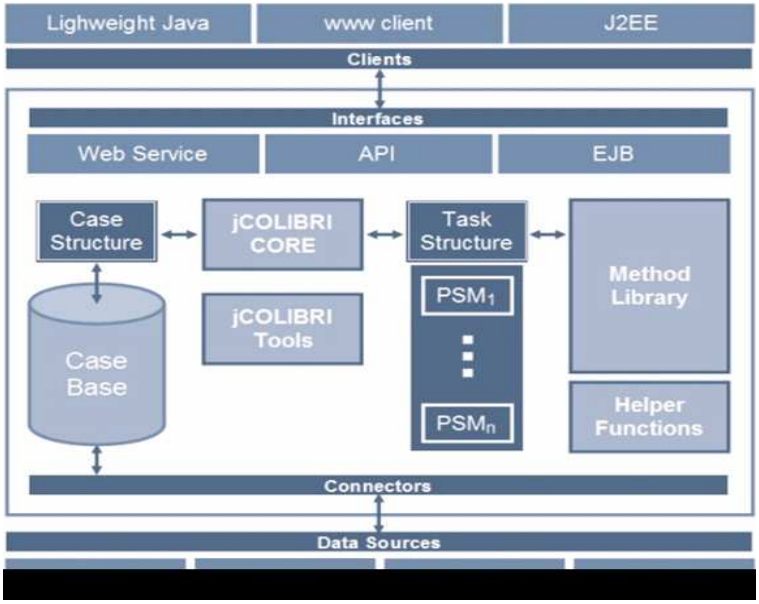


Figure 3.16. Architecture de JColibri [Bello-Thomas et al., 2004]

Le système JColibri est une application intégrée combinant la connaissance spécifique du cas avec les modèles terminologiques généraux du domaine. C'est une plateforme orientée objet développée en Java pour la création des applications du RàPC. Elle est basée sur une architecture Colibri (Cases and Ontology Libraries Integration for Building Reasoning Infrastructures) dont le principe de base est une acquisition des connaissances à partir des bibliothèques d'ontologies indépendantes des domaines d'application. Les bases de Colibri sont issues à partir des méthodes d'ingénierie de connaissances et notamment KADS qui dissocie la méthode de résolution des problèmes (raisonnement) du modèle de domaine décrivant la connaissance du domaine.

JColibri est basé sur la décomposition tâche-méthode du raisonnement. Les tâches de RàPC sont « mappées » en Java comme des classes abstraites ou les interfaces de l'architecture. La recherche des cas dans la base de cas se fait à l'aide des connecteurs. La gestion de la base de cas se fait par deux concepts séparés mais pourtant reliés : le mécanisme de persistance et l'organisation de « in-memory ».

Le mécanisme de persistance est basé sur les connecteurs - des objets sachant retrouver les cas et les rendre accessibles sous une forme unique.

Le deuxième principe de la gestion de la base de cas est la structure des données utilisée pour organiser les cas dans la mémoire. L'architecture laisse le choix libre de la structure des données utilisées.

Le cas est un individu qui peut avoir des relations avec d'autres individus (les attributs de cas) et qui peut utiliser les fichiers XML pouvant être échangés entre les utilisateurs. Il consiste à la description par les différents attributs, de la solution et du résultat (l'effet de l'application de cas

dans une situation réelle). Comme JColibri utilise une structure composite de cas, il y a deux types de fonctions de similarités : les fonctions de similarité locale qui calculent la similarité entre les attributs simples et les fonctions de similarité globale qui calculent la similarité entre les attributs composites. La composante qui maintient la configuration du système RàPC et l'exécution de l'application est le Core.

5.4. Tableaux récapitulatifs

Les sept systèmes de diagnostic sont résumés dans les deux tableaux ci-dessous. Dans le premier tableau 3.1, les caractéristiques générales de ces systèmes sont décrites, à savoir la famille du modèle (structurel, conversationnel), le type d'application (diagnostic général, help desk), l'outil de développement ou le système de représentation des connaissances (SRC) utilisé et la représentation des connaissances et du cas.

Le deuxième tableau 3.2 traite d'une manière plus détaillée les méthodes associées aux différentes phases du RàPC, la remémoration avec les mesures de similarité et les algorithmes de recherche, le principe d'adaptation et les autres méthodes utilisées dans les systèmes.

Système	**Modèle**	**Outil de dévelop.**	**Outil SRC**	**Type d'application**	**Représ. des connaiss.**	**Représentation des cas**
Cassiopée	Struct	Kate		Diagnostic des moteurs d'avions	Orientée objet	Liste des descripteurs hiérarchisés
CheckMate	Conv	CBR Express		Diagnostic des imprimantes – help desk	Textuelle Symbolique	Description de la panne, du contexte et l'action
CaseLine	Struct	ReMind		Diagnostic des moteurs d'avion – help desk	Textuelle	Numéro, nom de panne, description de la panne
CREEK	Struct		CREEK1	Diagnostic technique	Réseaux sémantiques	Ensemble de concepts (frame)
PATDEX	Conv		MOLTKE	Diagnostic technique	Orientée objet	Liste des descripteurs
PAD'IM	Struct		Rocade	Conception des systèmes de supervision	Orientée objet	Ensemble d'objets
MAIC	Struct		Optrans	Maintenance des équipements chimiques	Orientée objet	Liste des descripteurs qualitatifs et quantitatifs

Tableau 3.1. Tableau récapitulatif des systèmes en diagnostic

On peut constater que le modèle le plus répandu est le modèle structurel. Il y a cinq systèmes sur sept qui manipulent les connaissances sous forme d'objets ou concepts. Sur sept applications

décrites, trois (Cassiopée, CheckMate et CaseLine) sont développées à l'aide de systèmes de développement de RàPC commerciaux tels que Kate, CBR Express et ReMind. Parmi ces trois applications, deux d'entre elles sont de type help desk et ne présentent pas de phase d'adaptation et de maintenance de la base ce qui limite d'une manière conséquente le cycle de RàPC. Ces systèmes ont une représentation de cas type textuel, où les traitements sont plus orientés fouille de texte que RàPC.

Les autres systèmes académiques (CREEK, PATDEX, PAD'IM et MAIC) sont développés à l'aide des systèmes de représentation des connaissances (SCR). La représentation du cas le plus répandu est faite sous forme de liste d'attributs (descripteurs avec leurs valeurs relatives). Ces descripteurs peuvent être de différents types : symboliques, numériques, taxonomiques, type intervalles etc. Sur cinq applications, la représentation des cas est orientée objets basés sur des langages de programmation objets qui permettent de personnaliser les applications.

Dans le deuxième tableau, dans la phase de remémoration, on retrouve systématiquement l'algorithme des K plus proches voisins (KPPV) associé à des méthodes inductives exploitant la structure hiérarchique de la base de cas. Dans PATDEX et Cassiopée, les arbres de décision sont utilisés pour guider l'utilisateur dans la description du nouveau problème. Dans les systèmes à représentation textuelle de cas, se trouvant dans les systèmes help desk (CaseLine), les mesures de similarité sont basées sur les index et utilisent directement l'algorithme de recherche des K plus proches voisins.

Application	**Mesures de similarité**	**Recherche**	**Adaptation**	**Méthodes implémentée**	**Maintenance**
Cassiopée	Distance	KPPV Induction	Transformatio nnelle	Induction Arbre de décision	Raffinement de la base
CheckMate	Index	Recherche du texte Règles KPPV			
CaseLine	Index	KPPV			
CREEK	Index basés sur explication	Recherche indexée	Généralisation des attributs	Règles, modèles, explications	Connaissances dans la base
PATDEX	Mesures contenant l'incertitude et l'incomplétude	KPPV Induction	Graphe d'expériences	Induction Arbre de décision	Dans MOLTKE
PAD'IM	Conceptuel	KPPV	Substitution		
MAIC	Distance	« Easy retrieval »		Induction Simulation	

Tableau 3.2. Tableau récapitulatif des systèmes en diagnostic

La phase d'adaptation se révèle la plus délicate dans un système de raisonnement à partir de cas. Les applications développées par des systèmes commerciaux ne présentent pas d'adaptation, à l'exception de Cassiopée. Les autres systèmes proposent des règles d'adaptation simples, mais pas assez approfondies pour faire évoluer ou enchaîner les actions de réparation.

L'adaptation reste au niveau de la substitution des éléments (PAD'IM) ou de la généralisation des attributs basée sur les règles (CREEK). Dans les premiers systèmes développés, nous ne pouvons pas suivre l'évolution éventuelle de la situation étudiée. Les systèmes actuels essaient de remédier à ce manque en gardant la trace de raisonnement pour résoudre un problème à partir de sa description et effectuent ainsi différents types d'adaptations génératives (l'évolution de CREEK). D'autres systèmes utilisent la structure hiérarchique de la base de cas afin de réaliser une adaptation hiérarchique au niveau des descripteurs de cas (PAD'IM). Quand aux méthodes implémentées, nous trouvons alors les méthodes inductives basées sur les arbres de décision ou sur la hiérarchie des cas de la base de cas en phase de remémoration. Dans la phase d'adaptation, les systèmes appliquent les méthodes à base de modèles, de règles ou d'explications.

6. Nos choix et notre démarche

L'objectif de notre thèse est de proposer un système d'aide au diagnostic et à la réparation sur une plateforme d'e-maintenance. Au chapitre 2, nous avons choisi d'utiliser le raisonnement à partir de cas pour la résolution de notre problème que nous avons présentée dans ce chapitre. Afin de se situer par rapport aux travaux existants et de choisir les techniques les plus appropriées pour notre système, nous avons étudié sept systèmes de diagnostic industriel utilisant le raisonnement à partir de cas.

Pourquoi ne pas utiliser un système de développement ?

Parmi les applications de diagnostic étudiées trois d'entre elles ont été mis au point à l'aide de systèmes de développement. Ce sont des applications industrielles où une forte tendance de création des systèmes de type help desk peut être observée pour des raisons économiques. Ces applications n'exploitent pas toutes les phases de raisonnement. De plus, nous ne sommes pas dans l'optique de proposer à travers la plateforme, un interlocuteur à des néophytes.

L'implémentation industrielle des systèmes de diagnostic est limitée par les possibilités des systèmes de développement commercialisés. En effet, il existe actuellement plus d'une trentaine d'outils de développement des systèmes de RàPC existent sur le marché international, mais ceux-ci ne développent pas toutes les phases du raisonnement à partir de cas. Les utilisateurs sont souvent amenés à acheter plusieurs produits de la société distribuant l'outil de développement ; il

faut en effet agencer un ensemble de modules pour réaliser les différentes phases du raisonnement à partir de cas : module de création d'une base de cas, de sa mise à jour ainsi que sa maintenance, module destiné à la recherche des cas similaires et enfin module pour l'adaptation de cas s'il en existe. Certains systèmes comme Kate proposent un module d'adaptation, qui peut s'ajouter aux autres modules proposés. Mais cela peut poser problème quant à l'interactivité des phases de recherche de cas et d'adaptation de ces cas.

Nous avons choisi de développer notre propre système de RàPC afin de ne pas être limité par le manque d'interaction entre les différentes phases d'adaptation, d'élaboration de la connaissance etc. Nous nous intéressons à la partie représentation des connaissances en développant une démarche permettant d'élaborer un cas et d'y représenter l'expertise du domaine en utilisant des outils de sûreté de fonctionnement. Nous avons choisi la représentation orientée objet pour les avantages qu'elle amène. Les cas sont présentés par une liste d'attribut-valeur. L'approfondissement de cette étape de représentation des connaissances nous amène à avoir une démarche de gestion des connaissances.

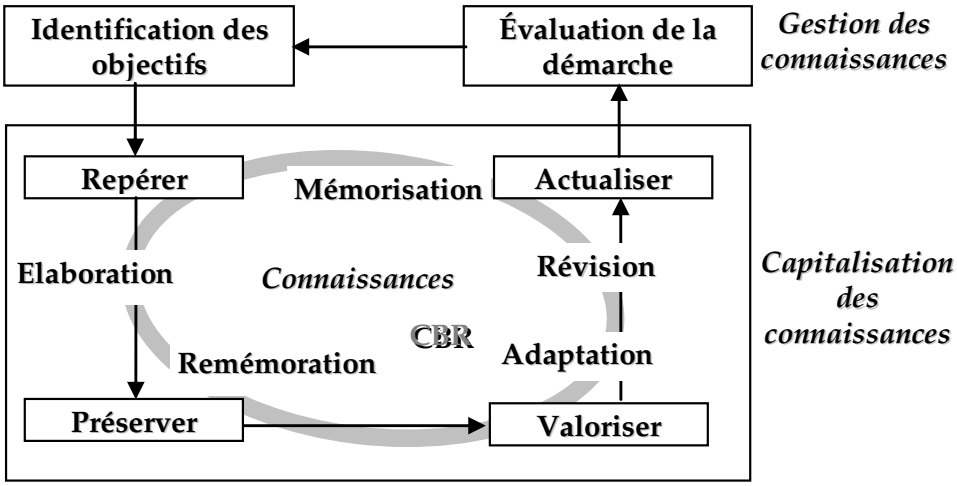


Figure 3.17. Le cycle de la gestion des connaissances et du RàPC

Un système de gestion des connaissances et de gestion d'une mémoire d'entreprise doit gérer conjointement des informations et des connaissances, les stocker sous forme persistante, les retrouver, les interroger en fonction de leur contenu, les visualiser et les réutiliser sous différents points de vues [Fuchs et al., 2006 réflexion]. Cela rejoint les tâches d'un système de raisonnement à partir de cas qui a pour objectif le stockage, le partage et l'utilisation des expériences individuelles [Chebel-Morello et al., 2005]. Le RàPC et la gestion des connaissances suivent donc le même objectif d'acquisition et de réutilisation d'expérience ou de connaissance (cf. fig. 3.17).

Nous tenons à faire remarquer que nous avons convergé vers la méthode présente après avoir suivi deux démarches différentes pour l'élaboration de notre système d'aide au diagnostic et à la réparation.

- La première démarche de capitalisation des connaissances nous a amené à proposer différentes connaissances dont une ontologie du domaine manipulée par les phases du RàPC.
- La deuxième démarche celle décrite dans ce chapitre en étudiant les différents systèmes existant du RàPC en diagnostic nous a amené à développer la phase représentation des connaissances comme dans [Fuchs, 1997] et de recouper ainsi la démarche de capitalisation.

Lorsque nous comparons ces deux approches, nous nous apercevons que le RàPC donne moins d'importance à la création de la connaissance (les cas sont supposés être déjà existants) contrairement à la gestion des connaissances qui s'appuie sur la création des modèles et des ontologies du domaine. Dans le RàPC, la représentation de la connaissance est simple mais l'énergie est portée sur les aspects de remémoration et d'adaptation. Alors qu'en gestion des connaissances l'importance est portée sur la description du domaine (connaissances du domaine). La représentation de cas complétée par des modèles de connaissances dans le raisonnement à partir de cas permet de considérer cette approche comme une approche de gestion des connaissances.

Due au fait que les ontologies n'ont pas leur propre mécanisme de raisonnement permettant d'utiliser les connaissances, les systèmes de gestion des connaissances emploient des méthodes d'inférence basées sur la logique. Ces méthodes trouvent les résultats corrects et prouvables basés sur les ontologies et les méta données. Les deux approches peuvent être complémentaires au sens que le RàPC permet de réutiliser et d'exploiter les bases de connaissances et notamment la mémoire d'entreprise faisant partie des systèmes de gestion des connaissances.

Les systèmes de RàPC sont isolés et fermés sur la coopération avec d'autres systèmes. Dans le passé, l'aspect de standardisation a été négligé et donc les systèmes ne sont pas réutilisables que dans la même application et ne permettent pas un échange facile de la connaissance. De leur côté, les ontologies sont souvent très spécifiques au domaine d'application même si elles utilisent les langages de représentation standardisés.

7. Conclusion

Le raisonnement à partir de cas a été choisi comme méthode de manipulation des connaissances dans la mémoire d'entreprise proposée au deuxième chapitre et comme méthode de résolution de problèmes dans notre système d'aide au diagnostic et à la réparation.

Nous avons introduit des principes fondamentaux d'un système de raisonnement à partir de cas et un état de l'art des méthodes utilisées dans le cycle de RàPC. Cela nous a permis de poser les jalons pour comparer les différents systèmes de RàPC appliqués au diagnostic technique. Le diagnostic fait partie des diverses applications supportées par le RàPC, que nous avons présenté par le biais de différentes typologies.

Le diagnostic est un domaine où de nombreux système de RàPC ont été expérimentés Nous avons fait leur comparaison à partir de différentes phases de raisonnement. Deux types de diagnostic ont émergé, le premier développé à l'aide de système de développement de RàPC et le deuxième à l'aide de système de représentation des connaissances. Un système à part, JColibri, présente les deux caractéristiques, c'est à la fois un système de développement des systèmes RàPC en lien avec un système de représentation des connaissances.

Nous avons déduit de cette étude plusieurs conclusions importantes pour la suite de notre travail. Nous ne nous orientons pas sur l'achat d'un système de développement, afin de ne pas être limité par le manque d'interaction entre les différentes phases d'adaptation, d'élaboration de la connaissance, etc.

Nous avons confirmé notre démarche de capitalisation des connaissances présentée au deuxième chapitre par une mise en parallèle entre le cycle du RàPC et celui de la capitalisation des connaissances. Nous allons alors créer une mémoire d'entreprise manipulée par le raisonnement à partir de cas.

Contrairement à la plupart des systèmes de diagnostic industriel, nous mettons un soin particulier à développer la phase de d'élaboration des cas. La construction de cas dans la base de cas se fait par l'utilisation des outils de sûreté de fonctionnement et tient également d'une démarche de capitalisation des connaissances. Le développement de notre système d'aide au diagnostic et à la réparation et son application dans le cadre de la plateforme générique d'e-maintenance est présenté au quatrième chapitre.

Chapitre 4

Développement et intégration d'un système d'aide au diagnostic et à la réparation dans une plateforme d'e-maintenance

1. Introduction 110

2. Cadre d'application 110

2.1. Architecture de la plateforme Proteus 112

2.2. Intégration d'un outil sur la plateforme 116

2.3. Plateforme de démonstration 119

3. Développement du système d'aide au diagnostic et à la réparation 122

3.1. Méthodologie proposée pour élaboration du système d'aide au diagnostic 123

3.2. Représentation et acquisition d'un cas 127

3.3. Remémoration des cas similaires 134

3.4. Adaptation 138

3.5. Révision et mémorisation 139

4. Conclusion 140

1. Introduction

Ce chapitre est consacré à l'implantation de notre système d'aide au diagnostic dans une plateforme d'e-maintenance. Nous développons un outil de gestion des connaissances, et plus particulièrement une mémoire d'entreprise ayant comme objectif la capitalisation des connaissances en diagnostic et en réparation.

Dans le paragraphe 2 de ce chapitre, nous présentons d'abord le cadre général de notre démarche, le projet PROTEUS avec ses différents objectifs et les groupes de travail. Ensuite nous présentons l'architecture de PROTEUS avec les différents composants et les outils nécessaires au bon fonctionnement de la plateforme d'intégration est présentée. Les fonctions des systèmes intégrés dans la plateforme sont proposées comme des Web services et donc l'intégration d'un tel service est introduite au niveau générique ainsi qu'au niveau spécifique pour notre outil d'aide au diagnostic basé sur le raisonnement à partir de cas. Une plateforme de démonstration (le système de transfert de palettes SORMEL) est présentée dans la dernière partie du paragraphe 2.

Dans le paragraphe 3, nous développons notre système de capitalisation des connaissances en diagnostic et en réparation qui fournira un service d'aide aux opérateurs de maintenance. Nous proposons une méthodologie d'élaboration d'un tel système basée sur l'ontologie des connaissances générales du domaine exploitées par le raisonnement à partir de cas. Ces connaissances font partie des deux premiers niveaux de la mémoire d'entreprise telle que nous l'avons présentée au chapitre 2. Un soin particulier dans le développement du système a été apporté à l'élaboration de la base de cas par des techniques empruntées à la sûreté de fonctionnement. Nous avons proposé un algorithme de recherche de cas similaires et les mesures de similarité associées à la fois à la structure de cas et à la structure hiérarchique de la base de cas. La recherche des cas similaires est orientée vers l'adaptabilité de ces cas. Nous avons proposé un algorithme d'adaptation basé sur la classe générale du composant défaillant en suivant leur typologie. Nous avons intégré cet outil dans une plateforme d'e-maintenance pour en montrer la faisabilité ; les différentes phases d'intégration sont illustrées par les diagrammes de séquence introduits en annexe D.

2. Cadre d'application

Dans le projet PROTEUS, 16 partenaires de trois pays Européens (France, Allemagne et Belgique) ont participés, à savoir : AKN, Arc informatique, Bikkit, Cegelec France, Cegelec Allemagne, Fraunhofer IML, IFAK, LAB et LIFC, LORIA, Pertinence DI, Schneider Electric, TIL, Université Paris 6, Université technique de Munich et Vartec. Un échange fructueux entre les

universités et les industriels a contribué aux résultats intéressants à la fin de ce projet en février 2005. Plusieurs démonstrations ont été faites tout au long du projet. En ce qui concerne les partenaires français, une démonstration a été faite sur un système de transfert des palettes Sormel au Laboratoire d'Automatique de Besançon et sur les machines-outils BAPS sur le site de Cegelec à Belfort.

Le projet a eu pour but le développement d'une plateforme distribuée, coopérative d'e-maintenance fournissant un ensemble d'outils d'aide à la décision associés aux différentes activités de maintenance, et concernant les différents acteurs allant du réparateur d'équipement au manager de contrat de maintenance. En apportant l'expertise, via Internet, directement sur le site d'utilisateur, le projet a contribué à la croissance de l'efficacité du processus de maintenance. Les autres buts économiques et organisationnels de ce projet se résument à :

- augmenter la productivité de l'entreprise par la mise en place d'un système permettant de fournir à un individu l'information utile au moment où il en a besoin, dans les meilleurs délais, et de façon exploitable,
- améliorer l'organisation en diminuant le gaspillage par une réutilisation des savoirs, des savoirs faire et des compétences développées au cours du temps,
- réduire les coûts de maintenance (le temps de diagnostic et la durée d'intervention),
- éviter les pannes et les défaillances par la surveillance des équipements en appliquant la maintenance prédictive.

En ce qui concerne la conception et la réalisation de la plateforme, 7 groupes de travail (workpackages) se sont penchés sur les différents aspects devant être réalisés :

WP1 : Dépôt et gestion des données techniques & e-documentation dont l'objectif est le management des données techniques et de la documentation associée dans un serveur d'e-documentation.

WP2 : Base de connaissances & aide à la décision dont l'objectif est de définir le management de la connaissance et les outils d'aide à la décision en maintenance.

WP3 : Acquisition de données & télécommande d'équipements dont l'objectif est la surveillance systématique des équipements et l'acquisition des données de capteurs.

WP4 : Gestion de contrat & portail Internet dont l'objectif est le management des contrats de maintenance ainsi que de définir un portail Web permettant l'accès à la plateforme aussi bien par les différents acteurs que par les différents systèmes connectés à la plate forme.

WP5 : Aide à l'intervenant & logistique dont l'objectif est la planification des opérations de maintenance et les problèmes de logistique.

WP6 : Architecture de la plateforme d'intégration d'outils & interfaces dont l'objectif est l'architecture de la plateforme et l'intégration d'applications par le développement des interfaces adaptées pour les différents outils, sous forme de web services.

WP7 : ***Démonstrateur*** dont l'objectif est de développer un démonstrateur pour des machines outils dans les ateliers mécaniques (en France) et pour des machines sur rail (en Allemagne).

En développant un système d'aide au diagnostic et à la réparation, nous nous plaçons dans le workpackage 2.

2.1. Architecture de la plateforme Proteus

La plateforme générique Proteus fournit les méthodes et les interfaces applicatives nécessaires à l'intégration des systèmes et des applications de maintenance existants ou nouveaux. Cette plateforme montre les avantages d'intégrer tous les aspects du processus de maintenance dans un même environnement multiutilisateurs basé sur le Web et accessible via un portail unique. Le projet propose une forme de standardisation par l'utilisation étendue des nouvelles technologies de structuration de données, des techniques d'intégration des applications ainsi que des technologies de l'Internet.

2.1.1. Composants de la plateforme

La plateforme illustrée à la figure 4.1 doit faciliter l'intégration de nombreux systèmes informatiques d'acquisition de données, de contrôle – commande, de gestion de la maintenance, d'aide au diagnostic, de gestion de la documentation, des bases de données et de connaissances, etc. Ces systèmes peuvent être classés dans trois principaux groupes tels que :

- ***des systèmes de synthèse et de présentation des données*** destinés aux utilisateurs finaux : les ordinateurs des utilisateurs, les téléphones portables, les PDAs, etc.,

- ***des systèmes assurant le bon fonctionnement de l'équipement à maintenir ainsi que la prise des mesures et l'acquisition des données*** provenant de différents capteurs installés sur cet équipement : le SCADA (Supervisory Control and Data Acquisition), les caméras infrarouges, les contrôles-commandes numériques des équipements, etc.,

- ***des systèmes de gestion des réparations et de la maintenance*** : la GMAO (Gestion de la Maintenance Assistée par Ordinateur), l'ERP (Enterprise Resource Planning), le système de gestion des données techniques et de la e-documentation, les systèmes d'aide à la décision en diagnostic, pronostic et réparation des équipements, etc.

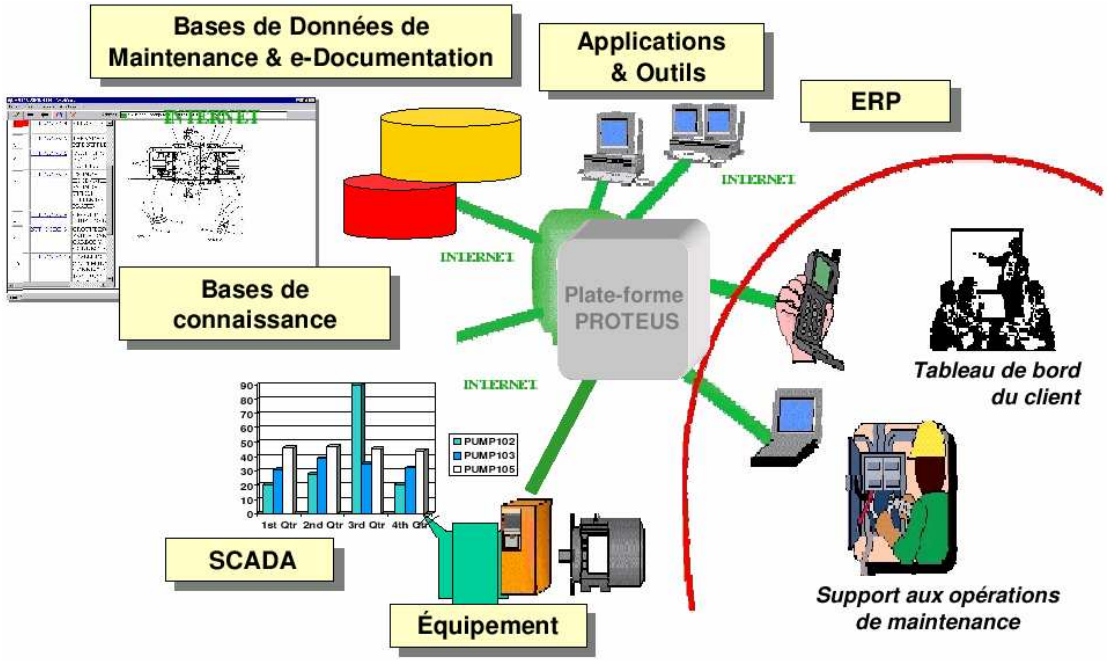


Figure 4.1. Plateforme d'e-maintenance Proteus

Ces différents composants de la plateforme sont proposés comme des services génériques sous forme de services Web. Chacun de ces systèmes s'appuie sur un certain modèle de l'entreprise, du système physique ou de l'équipement à maintenir. Ces modèles peuvent être différents parce que leurs objectifs sont différents, ils peuvent être incohérents car définis indépendamment les uns des autres. Les logiciels sont parfois redondants et souvent non-interopérables à cause de leurs représentations hétérogènes des informations et à cause de leurs interfaces incompatibles entre elles.

2.1.2. Plateforme d'intégration

Nous définissons l'architecture de la plateforme d'intégration sur la figure 4.2. Cette intégration relève d'une démarche plus générale dans l'industrie, intégration qui se veut la plus dynamique possible. La plateforme doit offrir un service global et homogène pour les utilisateurs. Pour cela, elle doit :

- garantir la coopération entre des applications de différentes natures (temps réel, transactionnelles, interactives),
- assurer l'échange d'informations entre sites distants,
- comprendre la variété des formats de données.

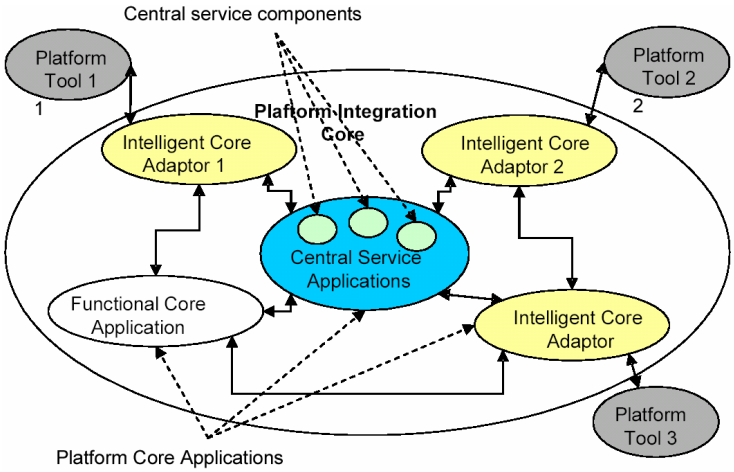


Figure 4.2. Architecture simplifiée de la plateforme Proteus [www.proteus-iteaproject.com]

Les différents outils sont présentés sur la figure 4.2. comme « Plateforme d'outils » et représente ainsi des services Web génériques définis pour l'ensemble des systèmes de même type. Ils offrent rarement la même interface. Pour chaque type de système une interface générique standard (***Application Programming Interface*** - API) est définie. Elle présente ainsi une vision « idéale » de l'outil et permet d'uniformiser l'accès à ses fonctionnalités depuis la plateforme.

L'implémentation de cette interface pour un outil existant est assurée par un type de composant appelé « ***Intelligent Core Adapter*** » (ICA). Le rôle de l'ICA est d'implémenter les différents Web Services dont l'ensemble standard est défini pour chaque type d'applications. La plateforme est basée sur une standardisation des communications. Les interactions doivent donc suivre des schémas de flots de contrôle. Le composant ICA permet aux services proposés par chaque application de correspondre à ces schémas.

Enfin, d'autres outils sont nécessaires pour le bon fonctionnement de la plateforme. Par exemple, la plateforme nécessite des serveurs d'authentification pour les utilisateurs et un annuaire des services disponibles. Ces différents outils sont intégrés au sein d'un composant appelé « ***Central Service Application*** ».

Afin d'assurer l'interopérabilité des échanges et le travail en coopération, nous proposons d'utiliser les Web services et le protocole SOAP (Simple Object Access Protocol). En effet, selon [Berners-Lee et al., 2001], ***le Web sémantique*** est « une extension du Web dans laquelle les informations sont fournies avec une signification bien définie, permettant aux ordinateurs et aux personnes de mieux travailler en coopération ». Les définitions plus approfondies du Web sémantique ainsi que des Web services sont présentées en annexe C.

2.1.3. Implémentation des scénarii de maintenance

Un des intérêts de la plateforme PROTEUS réside dans la possibilité d'automatiser tout ou partie des scénarii de maintenance. Chaque étape correspond à l'invocation d'un ou plusieurs services. Les résultats issus d'une étape conditionnent le choix de l'étape suivante. Cet ensemble de scénarii peut donc être vu comme un workflow définissant un processus générique. Il devient alors possible d'automatiser le traitement d'une alarme (par exemple) conformément aux scénarii de maintenance préalablement définis. Bien entendu, certaines étapes sont à effectuer par des humains/personnes.

Les scénarii de maintenance doivent pouvoir évoluer selon les besoins. Pour garantir une implémentation flexible de ces scénarii, la notion d'orchestration de Web Services a été utilisée. Cette approche présente plusieurs avantages :

- De nombreuses implémentations d'orchestration proposent un éditeur graphique. Cet aspect est très important car il permet à un spécialiste de maintenance de valider le codage des scénarii. L'utilisateur (non-spécialiste des techniques informatiques) peut les consulter et les adapter.
- Les interpréteurs d'orchestration exécutant automatiquement ces scénarii.
- Des formats de codage commun existent et permettent de passer d'un interpréteur à un autre.

Cependant, pour des raisons d'efficacité, il est toujours possible de définir des services génériques « figés » (i.e. qui n'évoluent pas) en codant directement l'enchaînement dans un langage de programmation.

Du scénario au workflow

Nous présentons sur la figure 4.3 un scénario simplifié de maintenance et nous en déduisons le scénario et les différents services Web génériques correspondants. Chaque étape a été détaillée de façon à correspondre à un service élémentaire. Le scénario de départ est le suivant :

- Une alarme est générée par le SCADA.
- L'utilisateur effectue une demande d'intervention.
- Il déclenche l'outil de diagnostic.
- Ce dernier récupère les données pertinentes auprès du SCADA.
- Il propose un diagnostic.
- L'utilisateur donne son Ordre de travail et réserve les pièces de rechange auprès de la GMAO.
- Il réserve les ressources humaines nécessaires auprès de l'ERP.
- Finalement, il demande à la documentation les gammes correspondantes (i.e. les procédures d'intervention).

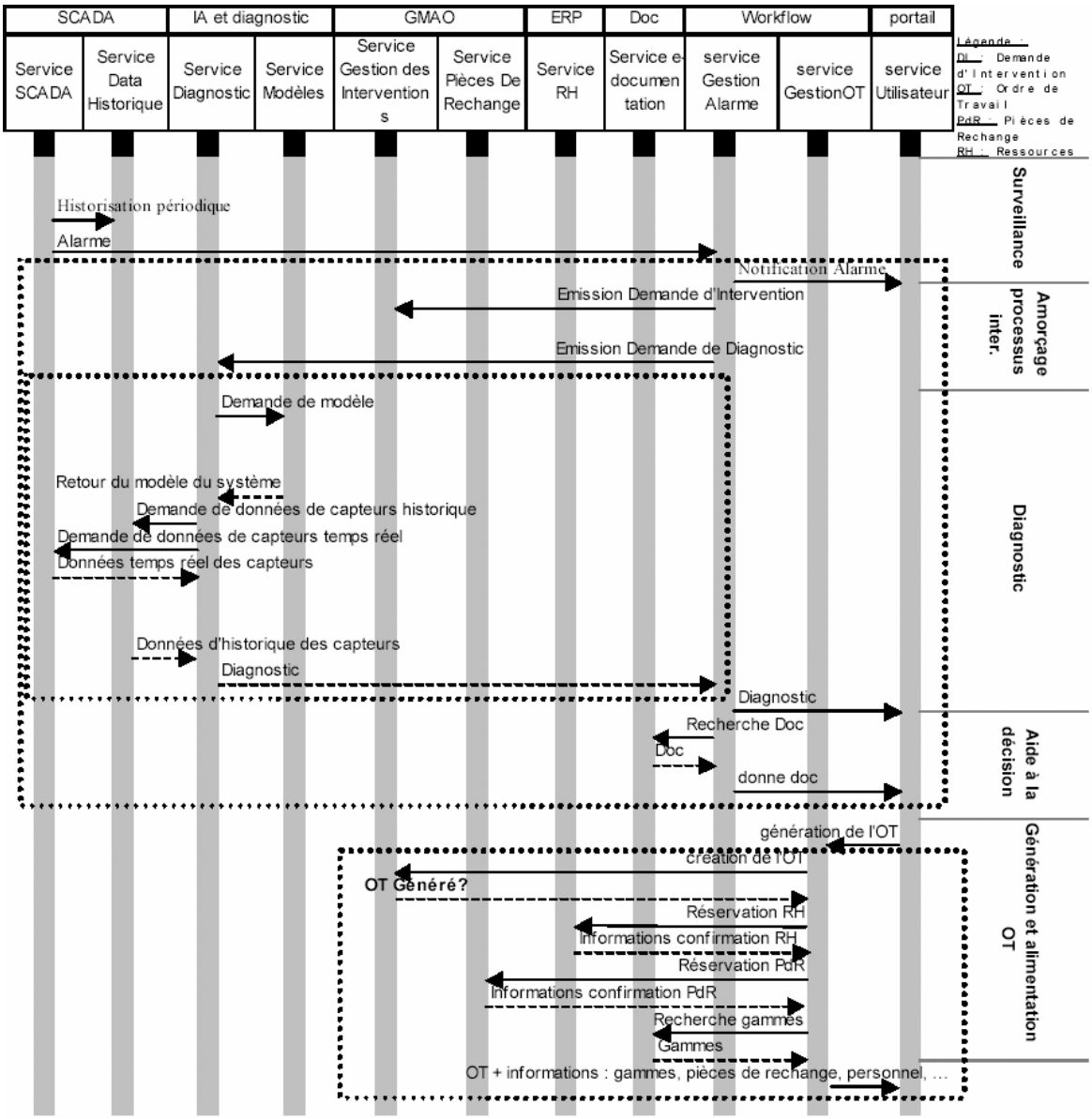


Figure 4.3. Identification d'un workflow partant d'un scénario

2.2. Intégration d'un outil sur la plateforme

Dans cette section, nous décrivons les services des différents composants de la plateforme - les services Web. Ils peuvent échanger, interpréter et intégrer des informations/données de différentes sources à l'aide d'une architecture du Web sémantique.

Un service Web générique : le diagnostic

Nous avons travaillé dans le workpackage 2 portant sur le développement d'un service Web générique qui propose les services de différents outils d'intelligence artificielle développés dans le projet [Proteus-WP2 Team, 2004]. Nous allons décrire ce service Web générique « Diagnostic ». Le diagnostic a pour objectif de déterminer l'état d'un équipement ou d'un procédé à partir

d'observations. Il s'agit d'évaluer si le fonctionnement est correct, dégradé, défaillant et de déterminer les composants qui sont en panne ou qui nécessitent une action de maintenance.

L'intégration de l'outil de diagnostic est facilitée par une architecture générique constituée d'un modèle général de données (cf. fig. 4.4), des workflows pour les différentes phases d'intégration et d'interfaces utilisateur (interface homme machine – IHM). Le modèle de données permet de formaliser l'outil de diagnostic regardé comme un outil « idéal » communiquant avec d'autres outils de la plateforme.

Les différentes phases d'intégration comprennent : la configuration, l'initialisation, le diagnostic et la mise à jour du modèle .

La configuration correspond à la création du modèle et à son application sur un équipement donné. Cela permet à l'utilisateur de relier (associer) les objets suivants :

- l'équipement,
- la technologie pour effectuer le diagnostic associé,
- la structure de données requise pour effectuer le diagnostic.

L'initialisation permet à l'utilisateur d' « apprendre » le modèle de décision utilisé dans le processus de décision et de le valider/raffiner manuellement.

Le diagnostic est principalement une présentation des résultats de décision à l'utilisateur. L'aide à la décision pour assister l'opérateur consiste à analyser une situation donnée et à établir la liste des entités ayant un rôle à jouer (ou ayant contribué) dans le contexte de la situation en cours. La liste des éléments pertinents trouvés est présentée à l'opérateur par l'intermédiaire d'un ensemble d'outils tels que des tableaux de bord.

La mise à jour du modèle met à jour le modèle de décision en utilisant les dernières données disponibles. Ce peut être une étape automatisée comme une étape manuelle. Une éventuelle phase de mise à jour de la configuration du modèle permet à l'outil de s'adapter à une nouvelle configuration (comme l'ajout d'un capteur) sans devoir réapprendre le modèle depuis le début en perdant les expériences passées.

Les différentes phases d'intégration peuvent être décrites par les digrammes de séquence. Nous présentons l'intégration de notre outil d'aide au diagnostic et à la réparation basé sur le raisonnement à partir de cas en annexe D. Nous avons choisi le langage de programmation Java pour implémenter le service de diagnostic basé sur le RàPC. L'interface homme machine est une interface Web Standard et est également présenté an annexe D.

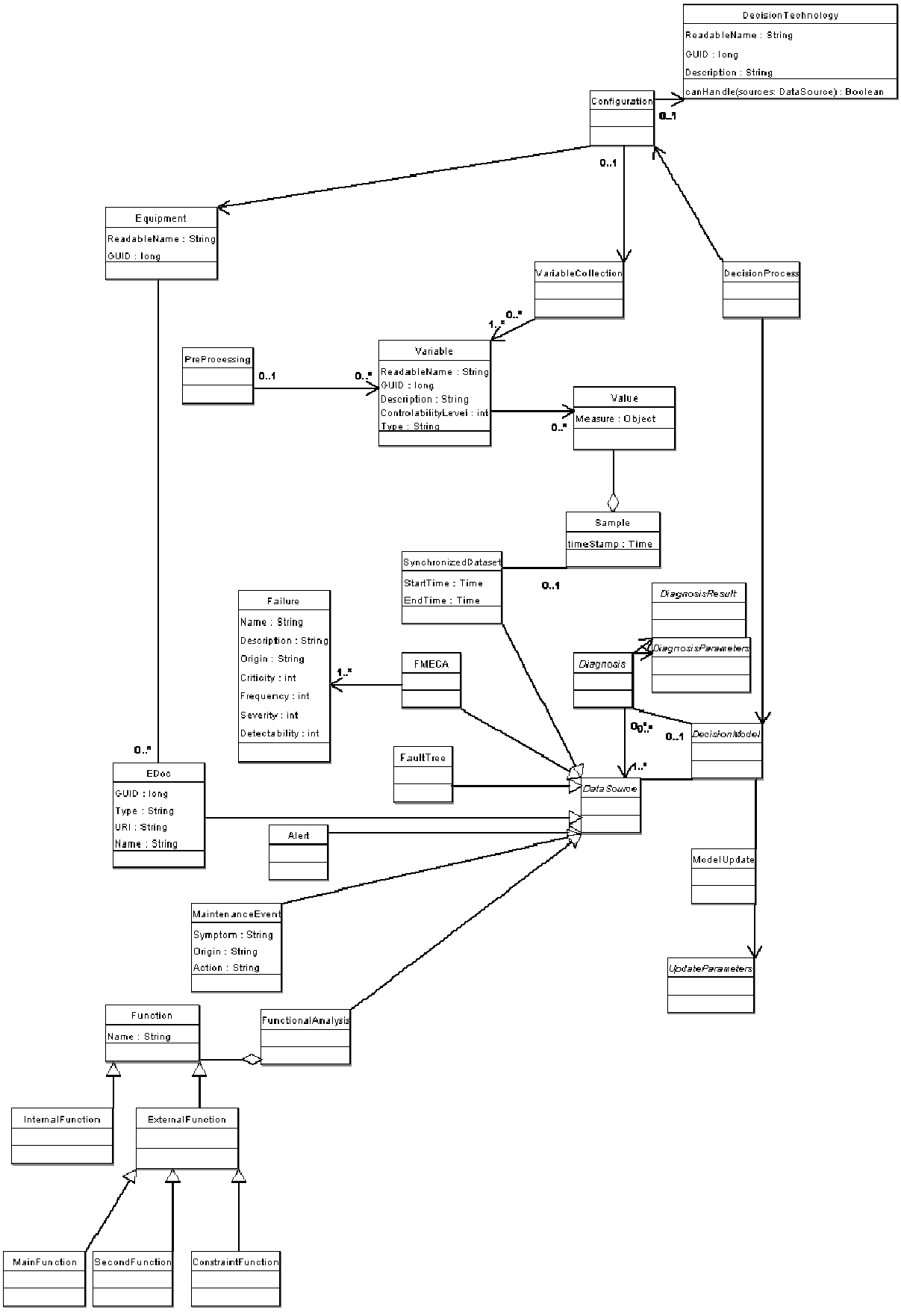


Figure 4.4. Le modèle général de données pour WP2

2.3. Plateforme de démonstration

Le transfert SORMEL, montré à la figure 4.5, est un îlot flexible d'assemblage organisé en double anneau, constitué d'un système de transfert de palettes et de 5 stations de travail. Ce transfert flexible est une installation automatisée, équipée d'automates de la marque Télémécanique (Groupe Schneider). Il permet de déplacer des palettes qui peuvent recevoir des pions ou des Legos. Les palettes sont véhiculées sur l'anneau intérieur permettant ainsi le transit entre les différentes cellules. Lorsque l'une des palettes doit subir une opération de la part d'un robot (information lue sur l'étiquette de la palette), cette dernière est déviée sur l'anneau extérieur où se trouve le poste de travail concerné.

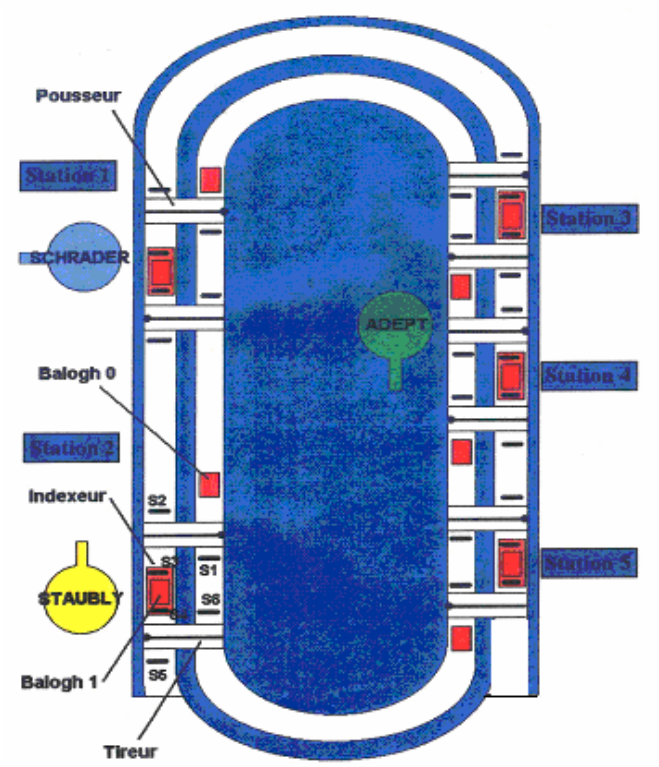


Figure 4.5. Ilot flexible d'assemblage SORMEL

La plateforme est divisée en cinq stations (cf. fig. 4.6) qui disposent chacune de leur automate. Elles fonctionnent donc indépendamment les unes des autres. Le déplacement des palettes est assuré par frottement sur des courroies, elles-mêmes entraînées par des moteurs électriques. Les palettes sont munies d'une étiquette magnétique qui leur sert de "mémoire embarquée". Ces mémoires peuvent être lues par chaque station grâce à des plots magnétiques de lecture / écriture (BALOGH) et mémorisent ainsi une gamme d'assemblage de produits. Ces étiquettes permettent donc de déterminer le cheminement des palettes à travers le système.

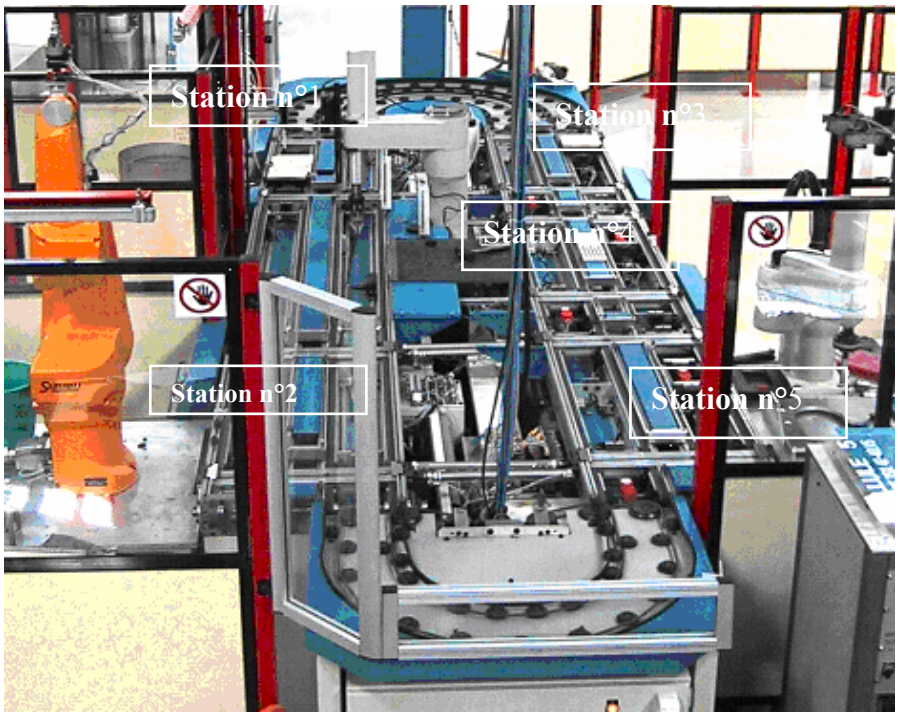


Figure 4.6. Vue d'ensemble de la plateforme flexible

Le système de transfert est constitué de 5 modules identiques. La figure 4.7 décrit la composition d'un de ces 5 modules. Chaque module comprend :

- ***Des actionneurs pneumatiques*** :
    - *Un pousseur* en entrée du poste qui dévie les palettes de l'anneau intérieur sur l'anneau extérieur en vue d'un traitement sur les postes.
    - *Un tireur* en sortie de poste qui ramène les palettes sur l'anneau intérieur si elles ne sont pas à travailler sur le prochain poste ou en limite de convoyeur.
    - *Un indexeur* permettant de bloquer les palettes lors de l'opération et d'assurer un positionnement précis de l'étiquette magnétique par rapport au plot de lecture/écriture Balogh.

- ***Des actionneurs électriques*** : 6 *stoppeurs* qui permettent de bloquer ou de laisser passer les palettes en différents endroits en fonction du routage de celles-ci, de leur environnement, de l'état des actionneurs, de l'état des postes.
- ***Un opérateur pour chaque station :*** station 1 - manipulateur Schrader, station 2 - robot Stäubli, station 3 - poste manuel et poste de chargement/déchargement des produits, station 4 - robot Adept One, station 5 - robot Adept Cobra.

- ***Des plots de lecture/écriture BALOGH*** qui permettent de lire et d'écrire des informations sur les étiquettes magnétiques des palettes. Le premier module de lecture se trouve à l'entrée de chaque cellule (sur l'anneau intérieur) et sert à déterminer si la palette doit être déviée ou non.

Il ne fait que la lecture à la volée, c'est-à-dire que la palette n'est pas arrêtée. Le second module se trouve au poste de travail (anneau extérieur) pour effectuer les différentes lectures/écritures des paramètres relatifs aux opérations. Ceci nécessite le blocage de la palette sur la tête magnétique.

- ***Des détecteurs de position*** : Chaque cellule contient 9 capteurs inductifs permettant de détecter la présence ou le bon positionnement des palettes (voir emplacement fig. 4.7 – points bleus). En effet, les palettes sont munies de deux éléments métalliques. La barre permet de détecter le passage d'une palette, le pion permet de s'assurer de sa bonne position (au niveau des pousseurs, tireurs et indexeurs).

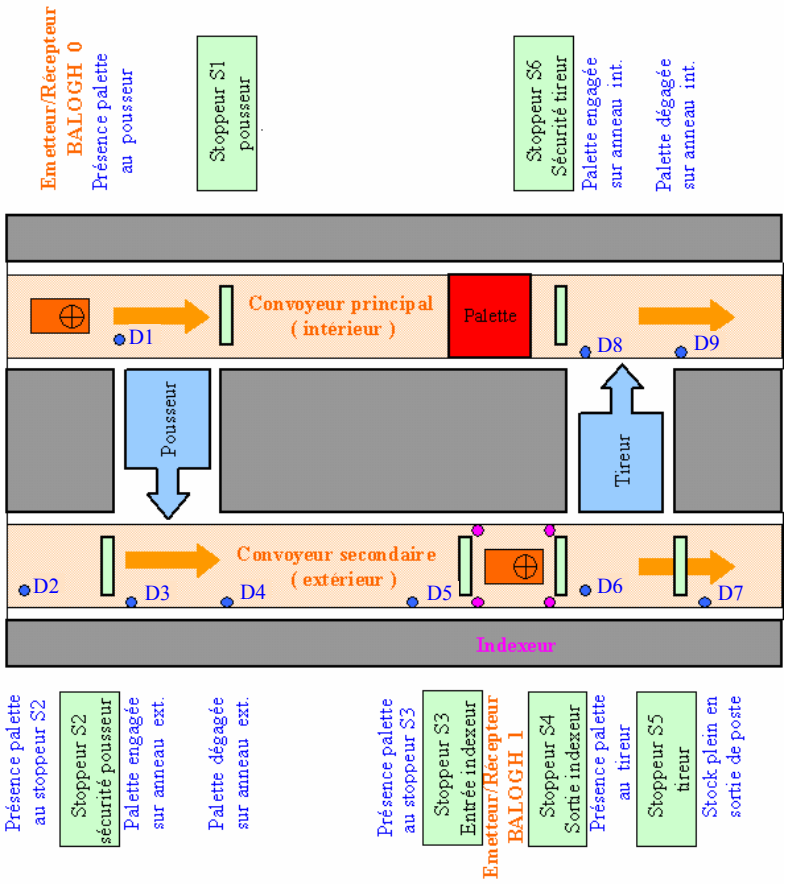


Figure 4.7. Composition d'un module du système de transfert SORMEL

Fonctionnement d'une station

Lorsqu'une palette entre dans une station, le plot de lecture à la volée (Balogh 0) lit au passage, sur l'étiquette de la palette, si le poste doit réaliser la prochaine opération à effectuer sur la palette. Si ce n'est pas le cas ou si la lecture a mal fonctionné, alors la palette continue son chemin sur l'anneau intérieur. Dans le cas contraire, la palette est envoyée sur l'anneau extérieur grâce au pousseur. Ensuite, la palette avance jusqu'à la station où elle est indexée. Le second plot de lecture/écriture (Balogh 1) lit alors sur l'étiquette les paramètres de l'opération à réaliser, c'est-à-

dire le numéro de la tâche en cours, le numéro de la palette, les paramètres de position… Puis, quand l'automate est informé que l'opération est finie, la palette est désindexée et le résultat et la durée de l'opération sont écrits sur l'étiquette. La palette quitte alors la station soit par l'anneau extérieur (si possible) si la prochaine opération est sur la station suivante, soit par l'anneau intérieur.

3. Développement du système d'aide au diagnostic et à la réparation

Nous proposons un outil permettant de capitaliser les connaissances dans le processus de maintenance. Ce processus peut être tracé par les différentes étapes d'intervention de maintenance décrite dans un document (papier ou informatique) contenant les informations sur cette intervention. Ce document change de forme en cours d'exécution de l'intervention et s'enrichit au fur et à mesure par d'autres informations. Nous montrons ses différentes formes dans les phases de processus de maintenance sur la figure 4.8.

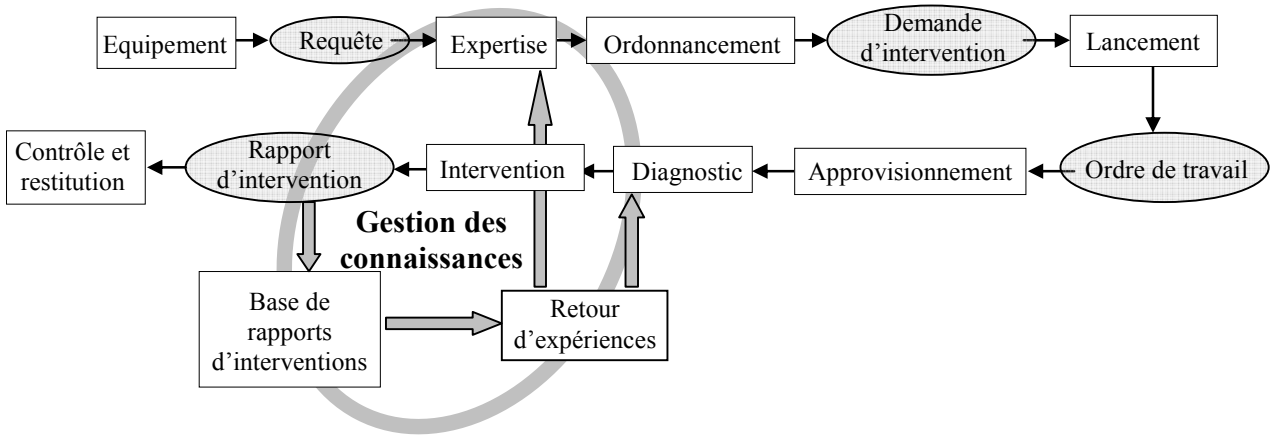


Figure 4.8. Le processus d'exécution d'une intervention

Tout d'abord, le besoin d'intervention sur un équipement donné doit être exprimé ce qui se fait sous forme ***d'une requête***. Ce besoin est manifesté soit par une émission d'alarme du système de surveillance qui envoie la requête de capteur, soit par le client (le service de production) qui envoie une requête externe, soit par le prestataire (le service de maintenance) qui envoie une requête interne. Cette requête, une fois arrivée au service de maintenance, se transforme en ***une demande d'intervention*** après avoir été vérifiée et ordonnancée, éventuellement enrichie par l'expertise sur des situations similaires déjà rencontrées et enregistrées dans une base. Lorsque le service de maintenance planifie cette intervention, elle est lancée sous forme ***d'un ordre de travail*** à l'opérateur concerné. A ce stade l'opérateur de maintenance effectue l'intervention et fait ***un rapport d'intervention.*** L'équipement est ensuite contrôlé et restitué. Les rapports d'intervention

peuvent être stockés dans une base permettant de mettre en place le retour d'expérience. Nous avons pour objectif de mettre au point à partir de cette base de rapports d'interventions une base de connaissances, plus précisément une base de cas. Cela nous permet de mettre en place un système d'aide au diagnostic et à la réparation. Cette base de cas participera ensuite à la création de la mémoire d'entreprise concernant le processus de maintenance en général.

3.1. Méthodologie proposée pour élaboration du système d'aide au diagnostic

Un des buts recherchés lors de la conception du système d'aide au diagnostic et à la réparation est de capitaliser la connaissance experte d'un opérateur lors d'un dépannage et de la partager entre opérateurs afin d'améliorer la performance et l'efficacité des interventions du service de maintenance. On a pu constater à partir d'une étude comparative sur différents systèmes existants qu'il n'y a pas de méthodologie d'élaboration d'un cas typique du diagnostic. Dans la plupart des applications, un cas reflète et traduit l'expérience de diagnostic sans pour autant faire référence à un modèle.

En ce qui nous concerne, nous proposons une méthodologie pour pallier au manque d'expertise en la créant par des outils de sûreté de fonctionnement ce qui permet d'élaborer un cas de diagnostic. Nous modélisons l'expertise issue de ces outils d'analyse par des méthodes de représentation et de modélisation des connaissances issues du domaine d'ingénierie des connaissances. Nous utilisons plus particulièrement la technique d'ontologies qui permet à la fois de modéliser les connaissances du domaine et les concepts liés à la méthode de raisonnement qui manipule les connaissances du domaine. Ces ontologies participent à la mémoire d'entreprise crée pour le processus de maintenance.

Le but du système d'aide est d'établir un diagnostic en déterminant le composant défaillant et de proposer une action de réparation adéquate avec des informations supplémentaires pouvant être recherchées dans la mémoire d'entreprise. Il s'agit de pièces de rechange, d'outils nécessaires, de compétence requise de l'opérateur et de documentation technique appropriée.

Le système est centré sur l'équipement à maintenir, par un premier niveau de surveillance lié à l'équipement (la supervision des données mesurées à l'aide de capteurs) et un deuxième niveau élaborant le diagnostic de pannes et leur réparation. Ces deux niveaux serviront ensuite de support à deux autres niveaux concernant la gestion des ressources aussi bien humaines que techniques, et la gestion des politiques de maintenance [Rasovska et al., 2004].

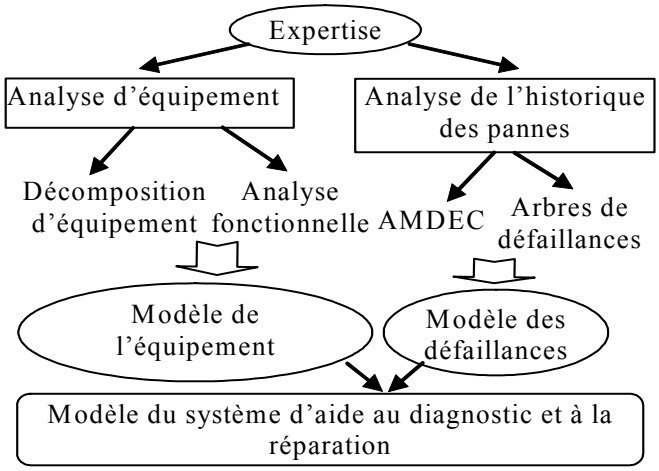


Figure 4.9. Méthodologie proposée – expertise de l'équipement

La conception de notre système nécessite la modélisation cognitive des processus et des connaissances utilisées par l'opérateur lors de son activité. La modélisation des connaissances du domaine de maintenance suit la démarche employée par les experts du domaine en appliquant des méthodes d'analyse relatives à la sûreté de fonctionnement (cf. fig. 4.9). Le modèle se base sur l'analyse de l'équipement et l'analyse de l'historique des pannes.

L'analyse de l'équipement, grâce à un système de décomposition, aboutit à une modélisation hiérarchique de l'équipement complétée par l'analyse fonctionnelle qui détermine les fonctions assurées par l'équipement et ses composants.

L'analyse de l'historique des pannes se base sur la méthode AMDEC (Analyse des Modes de Défaillance, de leur Effets et de leur Criticité) et sur les arbres de défaillances qui identifient les défaillances possibles de l'équipement et ses composants. Ces méthodes font actuellement objet de la recherche et de développement des outils automatiques [Kimura et al. 2002], [Barkai, 1999] ce qui nous permet d'envisager la récupération automatique de l'expertise issue de ces outils sous forme directement exploitable par le système d'aide.

Modèle de données du système d'aide au diagnostic basé sur le RàPC

Les analyses nous permettent donc de créer un modèle du domaine qui est ensuite associé à un modèle de raisonnement. La méthode de raisonnement choisie pour notre système est le raisonnement à partir de cas. Cela signifie qu'une vue spécifique de modèle d'expertise est une base de cas exploitable par les différentes fonctions du raisonnement à partir de cas. Nous y identifions les descripteurs pertinents pour la représentation de cas, une structure appropriée de cas associée aux connaissances générales et exploitées dans les phases de remémoration et d'adaptation.

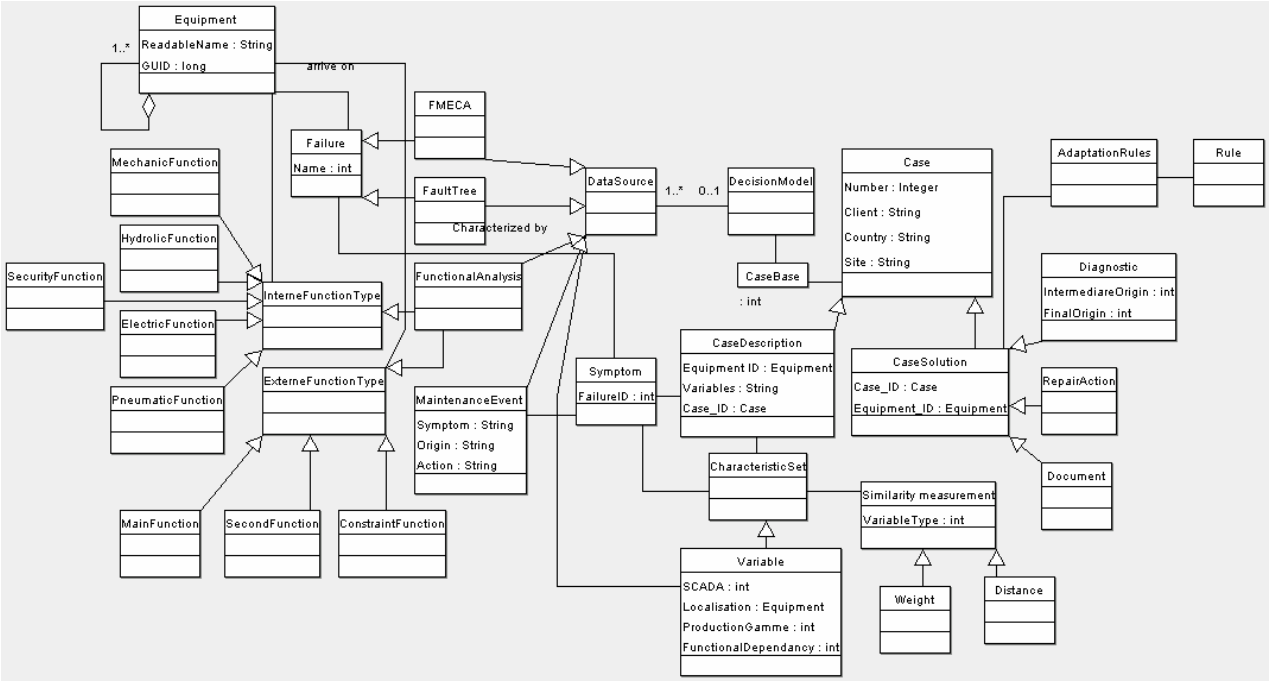


Figure 4.10. Modèle de données du système d'aide au diagnostic et à la réparation

Sur la figure 4.10 nous présentons le diagramme de classes UML décrivant, d'une part, l'équipement, sa décomposition et son fonctionnement et d'autre part, la structure du système d'aide au diagnostic et à la réparation basé sur le RàPC.

L'équipement se décompose en sous-équipements sous une forme arborescente [Rasovska et al., 2004]. A chaque équipement sont associées deux classes *InterneFunctionType* et *ExterneFunctionType* spécifiant le fonctionnement de celui-ci. La première classe représente le type de l'équipement (*MechanicFunction, HydrolicFunction* etc.) et concerne le fonctionnement interne. La deuxième classe est caractérisée par la hiérarchie des fonctions -fonction principale (*MainFunction*), secondaires (*SecondFunction*) et les contraintes (*ConstraintFunction*). De plus, une défaillance donnée (*Failure*) est associée à un équipement. Dans les arbres de défaillances, une panne est caractérisée et décrite par un symptôme causé par une autre défaillance appelée origine.

Le modèle du système de RàPC décrit par la classe *DecisionModel*. Le raisonnement à partir de cas stocke les cas (*Case*) dans la base de cas (*CaseBase*). Le cas est composé de deux parties différentes, à savoir la description du cas (*CaseDescription*) et sa solution (*CaseSolution*). La description est associée à la classe *CharacteristicSet* comportant les variables (*Variable*) caractérisant le symptôme de la défaillance (*Symptom*). Pour retrouver les cas similaires dans la base de cas nous associons à l'ensemble des variables les mesures de similarité (*SimilarityMeasurement*) qui sont basées sur la distance et du poids associé (*Weight*). Le diagnostic consiste à identifier les variables décrivant le symptôme permettant ainsi d'identifier l'origine de la panne et de déterminer l'action à entreprendre (*RepairAction*) et la documentation appropriée pour l'intervention (*Document*). Pour adapter les solutions des cas précédents aux

nouveaux cas nous introduisons la classe des règles d'adaptation. Chacune des classes de ces diagrammes peut être instanciée afin de créer des objets et par conséquent des cas. Un cas est représenté par une instanciation du diagramme de classes [Ruet, 2002].

L'intégration de notre système dans la plateforme générique permet une acquisition des données relativement facile et efficace. Le système récupère le maximum d'informations et de données automatiquement en renvoyant les requêtes automatisées à d'autres systèmes et bases de connaissances.

L'algorithme du système d'aide au diagnostic basé sur le RàPC

Les systèmes de RàPC modélisent et adoptent le raisonnement d'un être humain basé sur la remémoration des expériences précédentes. Cette manipulation des connaissances a été traduite dans un cycle de RàPC et a été adaptée à des systèmes de gestion des connaissances [Watson, 2001]. Nous avons traduit ce cycle en formalisme UML dans un diagramme de séquence décrivant les différentes étapes du système d'aide au diagnostic et à la réparation (cf. fig. 4.11). Ce diagramme permet de modéliser l'exploitation des connaissances par un système informatique [Rasovska et al., 2004b]. Après avoir reçu une demande d'intervention de la part de l'opérateur de production, le technicien de maintenance sollicite l'aide du système. Il envoie l'ordre de travail (WorkOrder) avec le premier symptôme identifié par l'opérateur de production. A partir de ce premier diagnostic, le processus du raisonnement à partir de cas crée le questionnaire (ensemble de questions fermées rempli par sélection de la bonne réponse par l'opérateur de maintenance). Les réponses serviront à la description d'un nouveau problème qui représente la première phase du cycle du système de RàPC (l'élaboration).

La deuxième étape, la recherche de cas similaires, prendra appui sur les mesures de similarité. Ces similarités sont calculées pour les origines du symptôme décrit dans la phase précédente et seront utilisées pour retrouver les cas similaires dans la base de cas. La réutilisation de la solution du cas similaire se fera soit en adoptant la solution du cas retrouvé soit en adaptant la solution du cas au nouveau cas ce qui correspond à la phase d'adaptation. Ensuite, l'opérateur de maintenance effectue la procédure donnée dans la solution. Une fois la procédure complétée, l'opérateur rend le rapport d'intervention qui indique la réussite ou non de la procédure. Si la procédure est satisfaisante, ce nouveau cas doit être introduit dans la base de cas. Dans le cas contraire, une autre solution basée sur un cas moins similaire que le précédent est proposée jusqu'au dernier cas dans la liste des cas similaires. L'opérateur face à une solution fausse proposée par le système peut appliquer sa propre solution. Dans ce cas, la solution doit être décrite dans le rapport d'intervention et ensuite enregistrée dans la base de cas. Les modifications dans la base de cas peuvent être effectuées seulement sous le contrôle d'un expert.

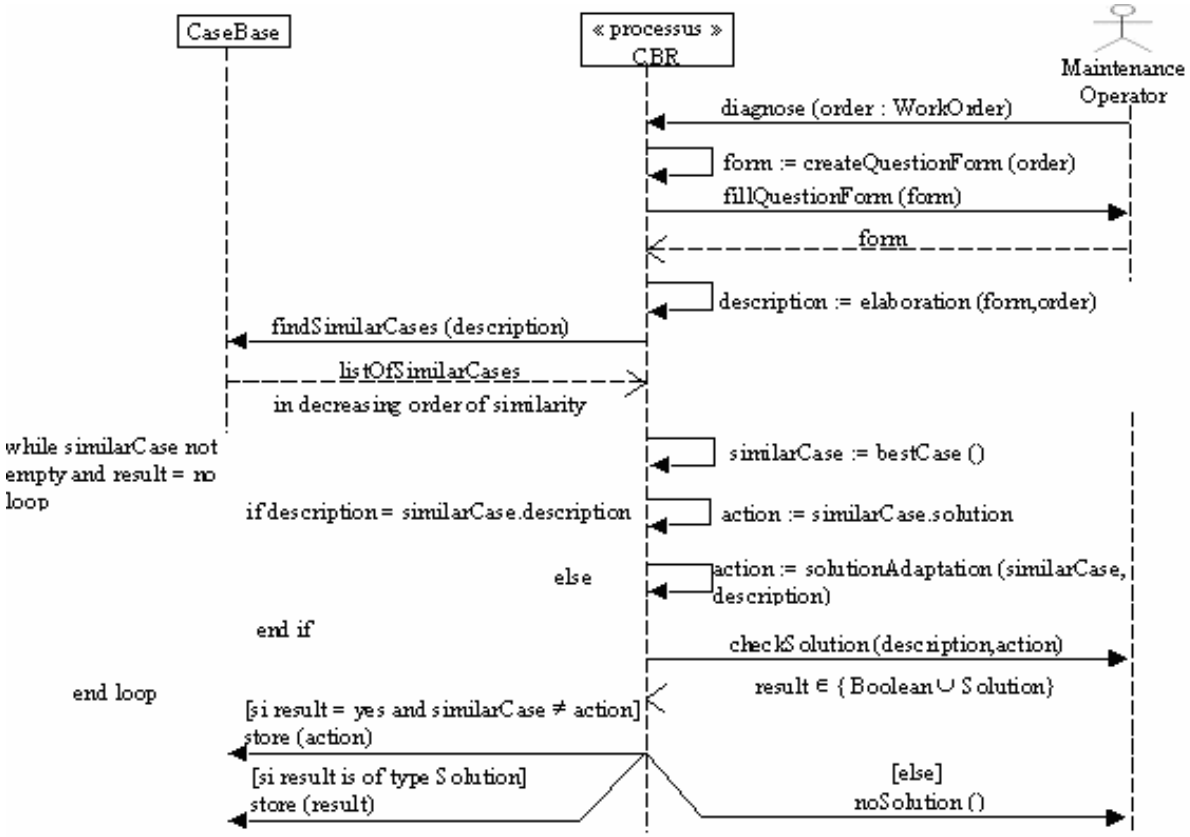


Figure 4.11. Le diagnostic

	Operation	Argument	Type	Argument	Type	Result
<<processus>> CBR	**diagnose**	WorkOrder	In			void
	createQuestionForm	WorkOrder	In			QuestionForm
	elaboration	QuestionForm	In	WorkOrder	In	Description
	bestCase					Case
	adaptSolution	Case	In	Description	In	Action
BaseCase	**findSimilarCases**	Description	In			List Of Case
	store	Case	In			
MaintenanceOperator	**fillQuestionForm**	QuestionForm	In/Out			
	checkSolution	Description	In	Action	In	Case Or Boolean
	noSolution					

Tableau 4.1. Liste des opérations

3.2. Représentation et acquisition d'un cas

La représentation d'un cas consiste à faciliter la description du problème pour permettre la recherche d'un cas dont la solution sera facilement adaptable. La représentation la plus communément reprise est la représentation structurée en liste de descripteurs qui peuvent être des objets complexes. La représentation orientée objet permet de manipuler les connaissances à caractère complexe à l'aide de différentes formes de mise en relation des classes représentant les concepts du monde [Ruet, 2002].

On peut trouver un parallélisme entre la composition d'un cas et la définition d'un objet. L'objet possède une identité, un état et un comportement parallèlement au cas qui est constitué de champs représentant les indexes (cela peut être un symptôme final), le contexte et l'action ou thérapie. Le cas est à la représentation des connaissances ce qu'est l'objet aux langages de programmation. La représentation objet permet d'élaborer aisément une structure hiérarchique de descripteurs que l'on pourra exploiter dans les phases de remémoration et d'adaptation.

3.2.1. Ontologies des connaissances générales

Synthèse des connaissances générales (modélisation triviale)

La première synthèse des connaissances du domaine concerne le modèle d'équipement et le modèle des défaillances. A partir de ces deux modèles et de la composition du système Sormel, nous avons développé une ontologie des descripteurs du cas montrée sur la figure 4.12. Elle nous a permis d'identifier les descripteurs nécessaires et incontournables dans la représentation d'un cas correspondant aux composants du système de transfert. Ce système a été découpé premièrement selon l'analyse fonctionnelle en composants et deuxièmement selon la décomposition géographique en ensembles, zones, sous-zones, etc.

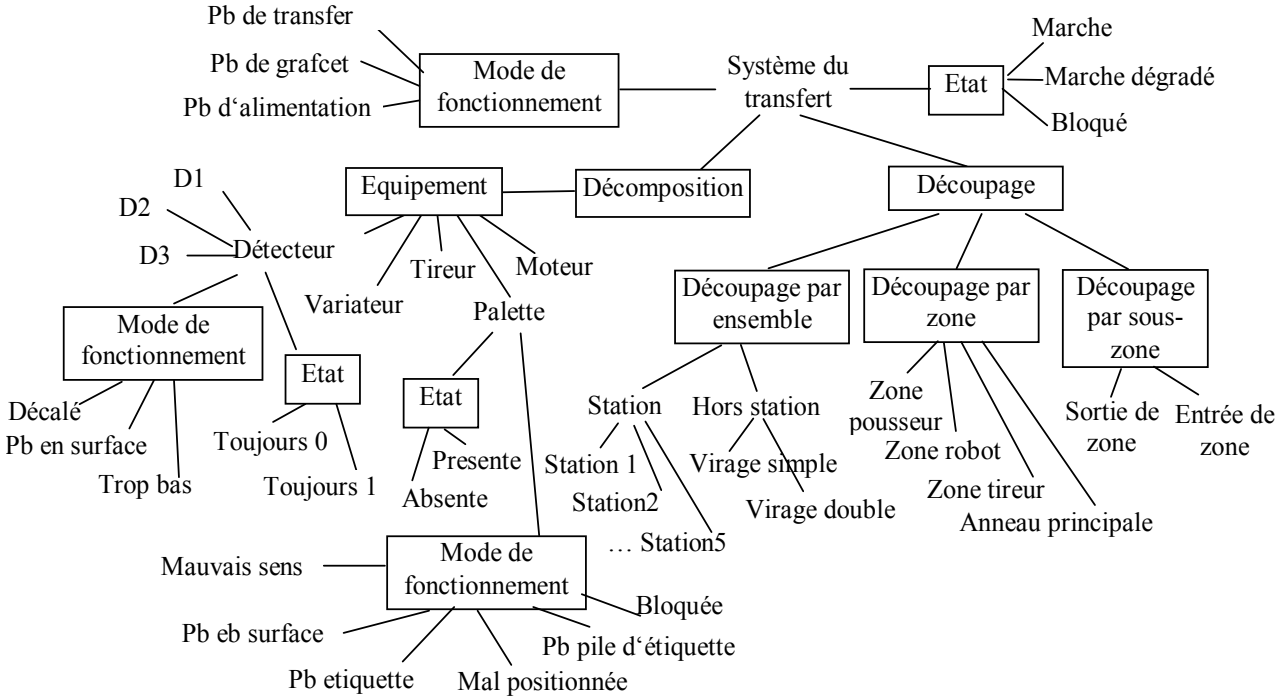


Figure 4.12. Ontologie des descripteurs

Chaque composant peut être caractérisé par son état et les modes de fonctionnement (défaillances) ainsi que par l'état du contexte de ce composant grâce à l'AMDEC, l'historique des pannes et les arbres de défaillances. Dans le cas relatif au transfert de palettes, les composants sont

de différentes natures, à savoir capteurs, contrôleurs, unités de commande, etc. L'exemple pour le détecteur est montré au tableau 4.2.

Composant	**Symbole**	**Contexte**	**Etat de composant**	**Mode de défaillance de l'équipement : *action***
Détecteur	D1,D2 D3,D4 D5,D6	Palette : Présente	Tjrs 0	Pb détecteur - trop bas : *lever détecteur* Pb détecteur – décalé : *pousser détecteur* Pb détecteur – défectueux : *changer détecteur*
	D7,D8 D9	Palette : Absente	Tjrs 1	Pb détecteur - pb en surface (élément métallique, poussière de fer) : *nettoyer détecteur*

Tableau 4.2. Exemple des modes de fonctionnement de Détecteur

Nous identifions d'abord le contexte dans lequel le composant donné se trouve. Pour un détecteur de présence de la palette, le contexte comprend la présence ou l'absence de la palette. Ensuite nous évaluons l'état du détecteur (il peut prendre la valeur 1 s'il détecte la palette ou la valeur 0 s'il ne la détecte pas). Le mode de fonctionnement du détecteur résulte de son état et de l'état des composants du contexte. A titre d'exemple, si la palette est présente et dans le bon sens mais le détecteur ne la détecte pas (égal à 0), le mode de fonctionnement est dégradé (le détecteur peut être mal positionné, décalé ou défectueux). Nous déterminons aussi l'action à faire pour corriger le problème.

Analyse des connaissances générales (modélisation approfondie)

La modélisation approfondie des connaissances générales, à savoir la typologie des composants et le découpage spatial du système de transfert, servira également aux autres phases de raisonnement à partir de cas telles que la remémoration, l'adaptation et la maintenance de la base de cas.

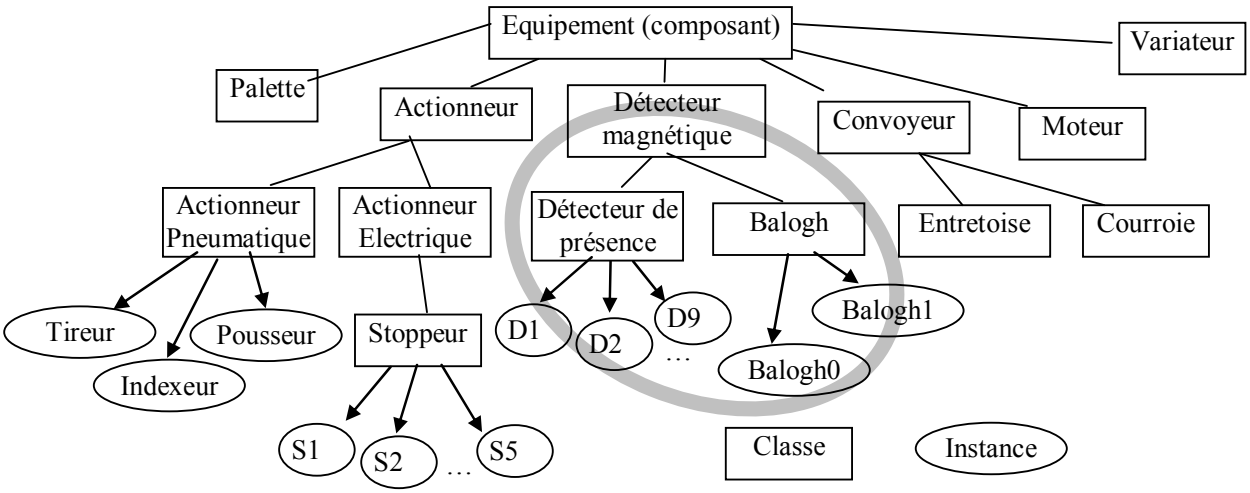


Figure 4.13. Typologie des composants du système

A partir de l'analyse fonctionnelle d'équipements, nous avons construit une typologie des composants du système qui correspond à un modèle fonctionnel (cf. fig. 4.13). Nous avons déterminé les ensembles de composants dont le fonctionnement est similaire ou assuré par les mêmes fonctions (électrique, pneumatique, mécanique, magnétique).

Un exemple est illustré par la partie entourée en gris sur la figure 4.13 pour la classe des détecteurs fonctionnant en mode magnétique. Deux équipements assurent la même fonction de détection de la présence de palette, à savoir le détecteur magnétique et le plot magnétique Balogh. Le Balogh assure en plus une fonction supplémentaire de lecture et écriture sur l'étiquette de la palette. Ensuite chacune de ces classes a des instances correspondant à des composants concrets tels que les détecteurs D1, D2,…, D7 et les Balogh0, Balogh1.

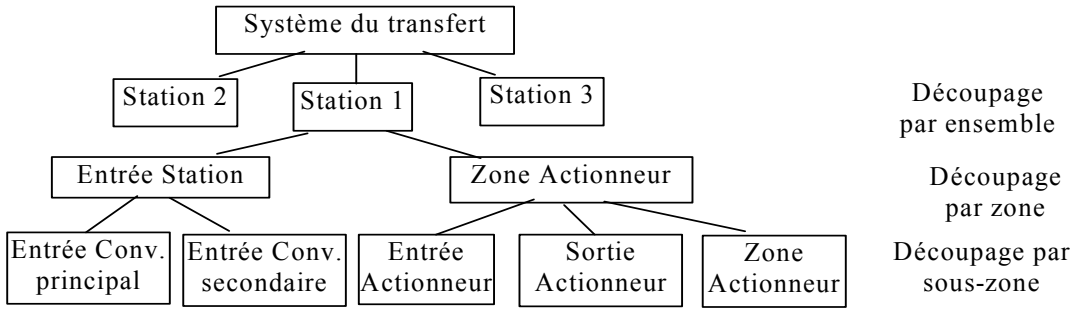


Figure 4.14. Aperçu de découpage spatial du modèle

Le découpage spatial correspond à la décomposition géographique du système de transfert et se justifie pour le suivi de la palette qui est transférée d'un poste vers un autre et qui parcourt toutes les zones spécifiées sur la fig. 4.14. La plateforme de démonstration est décomposée en ensembles, zones et sous zones.

3.2.2. Représentation du cas

A partir des ontologies nous avons développé la représentation de cas pour notre système d'aide au diagnostic. Un exemple de cas est donné sur la figure 4.15. Sa structure dans la base de cas (cf. fig. 4.16) a été choisie de façon à ce qu'elle montre facilement les différences dans les informations sur les défaillances. La première partie représente la description de la situation diagnostiquée et contient :

- ***Le contexte*** constitué de premier symptôme et de localisation du problème (de la défaillance) dans le découpage spatial (ensemble, zone, sous zone appelés des modules dans la base de cas). C'est la partie discriminatoire de la description du cas pour la recherche des cas similaires.
- ***La liste des attributs*** (descripteurs) correspondant à l'ensemble des composants (associés à chaque zone ou sous zone dans le contexte) et leurs valeurs indiquées par les états de ces composants. Cette partie permet d'évaluer la similarité du cas cible avec les cas sources.

La deuxième partie représente l'évaluation et la solution de la situation diagnostiquée et contient :

- ***L'évaluation*** qui correspond à l'identification du composant défaillant caractérisé par une défaillance détaillée. Ce composant est associé à sa classe générale dans l'ontologie des descripteurs. Cette partie permettra d'adapter la solution du cas similaire source au cas cible.
- ***La solution*** consiste à proposer une action de réparation associée à la documentation technique appropriée, la proposition des pièces de rechange, les outils nécessaires, la compétence requise, le temps d'intervention.

Contexte	Cas	Attribut : valeur
	Symptôme	Pb de transfert [pb d'alimentation, pb de grafcet, pb de transfert]
	Identification du contexte : localisation	Localisation_ensemble : station [entretoise, virage-simple, virage_double] Localisation_zone: actionneur [poste, convoyer] Localisation_sous-zone : entrée [sortie]
Liste des attributs	Attributs du contexte : état	Détecteur D6 : 1 [0 (palette absente), 1 (palette présente)] Balogh 1 : 1 [0 (palette n'entre pas dans la station), 1 (palette entre)] Stoppeur S5 : 0 [0 (stoppeur en haut), 1 (stoppeur en bas)] Pousseur: 0 ne revient pas [tire, ne revient pas, ne tire pas]
Evaluation	Partie discriminatoire	Symptôme pas de transfert indique : palette bloquée Localisation de la palette bloquée : station.actionneur.entrée
	Composant défaillant Mode de fonctionnement	Pousseur Défaillant - vérin hors service
	Classe	Actionneur défaillant
Solution	Action, vérification	Changer vérin de l'actionneur
	Compétence Pièce de rechange Outils Document Temps	Mécanicien Vérin de l'actionneur Tournevis Schéma N13002 30 minutes

Figure 4.15. Exemple d'un cas

3.2.3. Base de cas

Un aperçu de cas dans la base de cas est illustré sur la figure 4.16. Le développement de la base de cas a été présenté dans [Rasovska et al., 2005b]. La base de cas est structurée selon la représentation de cas en quatre parties correspondant au contexte, à la liste des attributs valeurs, l'évaluation et la solution.

	Contexte		Liste des attributs–valeurs				Evaluation		Solution
	Symptôme	Localisation					Composant défaillant : mode de fonct.°	Classe générale	
Cas	Pb de transfert	Station,Actionneur.Entrée	D6 :1	B1 :1	S5 :0	P :0	Pousseur : Vérin hors service	Actionneur	Changer vérin

Figure 4.16. Le cas dans la base de cas

La hiérarchisation de l'information pour l'élaboration du cas dans la base de cas est définie sur la figure 4.17 en considérant différents niveaux dépendant de la complexité du système de transfert. Néanmoins, le résultat de diagnostic peut être plus ou moins précis ce qui dépend des parties qualitatives retrouvés (notamment de la classe générale dans l'ontologie des descripteurs).

Le dernier niveau de la hiérarchie correspond au mode de fonctionnement du composant défaillant. Le niveau au-dessus est défini comme le niveau des composants, i.e. ici le système sera capable de diagnostiquer quel composant est défaillant mais ne sait pas encore de quel type de défaillance il s'agit. Il est possible que certaines défaillances puissent être localisées seulement au niveau de la hiérarchie spatiale, i.e. dans les ensembles, les zones et les sous zones que nous appelons ainsi des modules. Le nombre de niveaux des modules dépend de la complexité de système (ici 3 niveaux). Afin de créer une hiérarchie de la base de cas, nous sommes amenés à utiliser les connaissances du domaine (des ontologies).

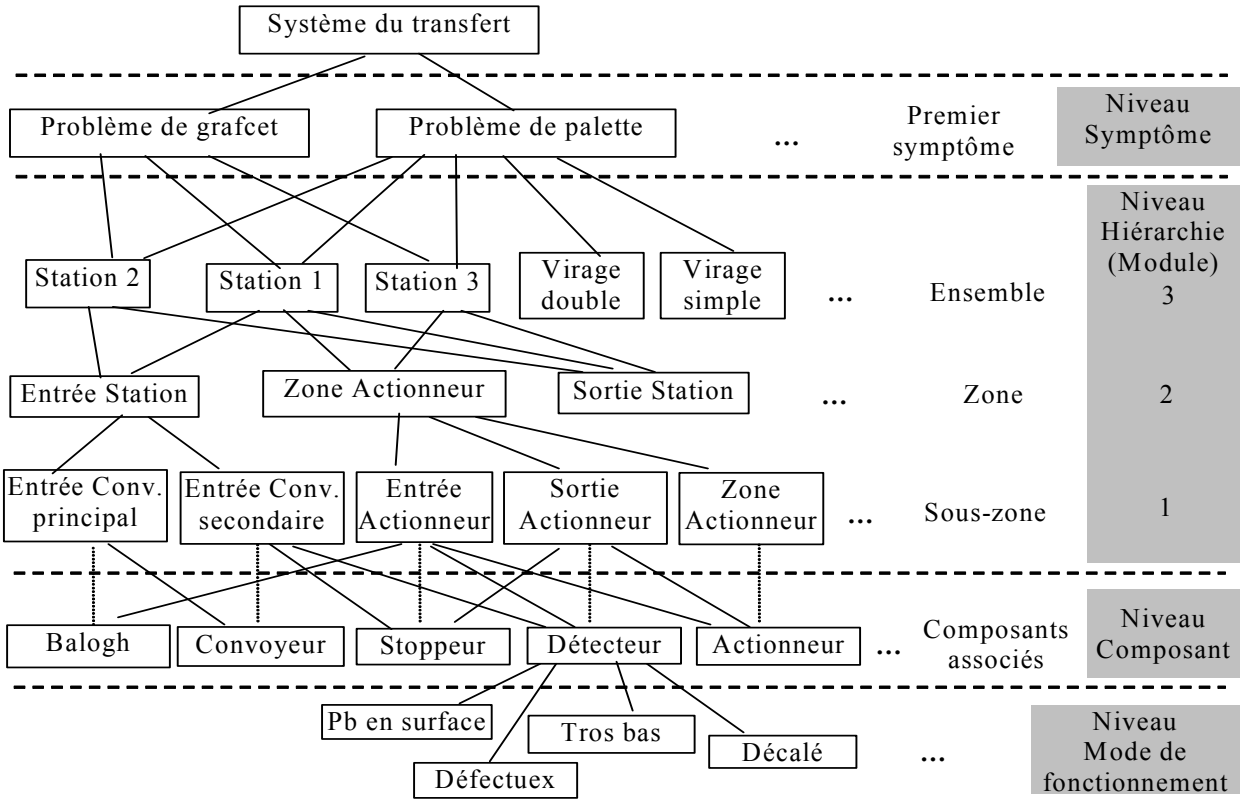


Figure 4.17. Hiérarchisation de l'information pour la base de cas

Une première phase de création de base de cas nous a amené à définir une base de cas brute où les descripteurs pertinents associés à chaque cas sont recensés. La démarche utilisée consiste à compléter ou filtrer la description d'un problème en se fondant sur la connaissance du domaine. La structure particulière de cette base comprend une liste de descripteurs dépendant du contexte. Cela nous a amené à hiérarchiser la base de cas en deux niveaux. Un niveau dépendant du contexte qui

dissociera la base en un ensemble de sous bases sur lesquelles nous appliquerons un algorithme de remémoration spécifique. Nous montrons une partie de cette base de cas à l'état brute sur la figure 4.18.

Ensemble	Zone	Sous-zone	Situer le contexte		Recensement de descripteurs pertinents
			Zone précédente	Zone suivante	
Station	Anneau secondaire	Entrée	Sortie anneau sec prec	Sortie pousseur	D2, S2, Tapis ext
		Convoyeur extérieur	Sortie pousseur	Entrée poste	D3, D4, Tapis ext
		Sortie	Entrée tireur	Entrée anneau sec suiv	S5, D7, Tapis ext
	Zone pousseur	Entrée	Entrée anneau prin	Pousseur	D1, S1, BLGH0, Pousseur
		Pousseur	Entrée pousseur	Sortie pousseur	Pousseur
		Sortie	Entrée anneau sec, Pousseur	Entrée conv externe	Pousseur, D3, D2, S2
	Zone poste	Entrée	Convoyeur exterieur	Indexeur	D5, S3, Tapis ext
		Poste	Indexeur	Entrée tireur	Balogh1, S4, Indexeur, Robot

Figure 4.18. Base de cas à l'état brute

Dans la deuxième phase de création de base de cas, nous avons déduit les descriptions des cas pertinentes (éventuellement incomplètes) considérées nécessaires et suffisantes pour la solution du problème. Nous pouvons également pondérer les descripteurs en fonction de leurs dépendances identifiées. Nous montrons un aperçu de cette base sur la figure 4.19.

Cas	BASE DE CAS							
	Ensemble	Zone	Sous-zone	Composant	Détecteur	Stoppeur	Var du contexte	Composant défaillant
1	Station	Anneau prin	Entrée	Tapis int	Bal0=0 (presence)			Pb Balogh
3	Station	Anneau prin	Convoyeur intérieur	Tapis int		S6=0 haut	D8=0	Pb stopp S6
5	Station	Anneau prin	Sortie	Tapis int	D9=0			Pb détect D9
7	Station	Anneau prin	Sortie	Tapis int	D9=1		D8=1	Pb détect D8
8	Station	Anneau sec	Entrée	Tapis ext	D2=1	S2=0 haut	D3=0	Pb stopp S2
9	Station	Zone poste	Indexeur	Indexeur	I=0 bas	S4=1 bas		Pb stopp S4
11	Station	Zone pouss	sortie	Pousseur	D3=1		D4=1	Pb D4
12	Station	Zone pouss	sortie	Pousseur	D3=1		D4=0	Pb pouss vérin
13	Station	Zone pouss	entrée	Pousseur	D1=1	S1=1 bas	Bal0=0,1	Pb S1

Figure 4.19. Base de cas structurée

Elaboration d'un nouveau cas cible

Dans la phase d'élaboration, un formulaire composé de questions fermées issues d'une structure arborescente est proposé aux acteurs (cf. fig. 4.20). Ce formulaire permet de cerner le contexte en localisant la panne qui se détermine par ensembles, zones et sous zones et d'identifier les composants de ce contexte et leurs états (des équipements et leurs valeurs). Notre base de cas est ainsi constituée de trois parties discriminées par la nature du premier symptôme (problème de grafcet, problème d'alimentation, problème de transfert). Dans chaque partie de la base, les cas associés au contexte donné sont enregistrés dans une hiérarchie suivant la décomposition spatiale (dans le cas de problème de transfert) ou la décomposition fonctionnelle (dans le cas de problème d'alimentation, ou problème de grafcet).

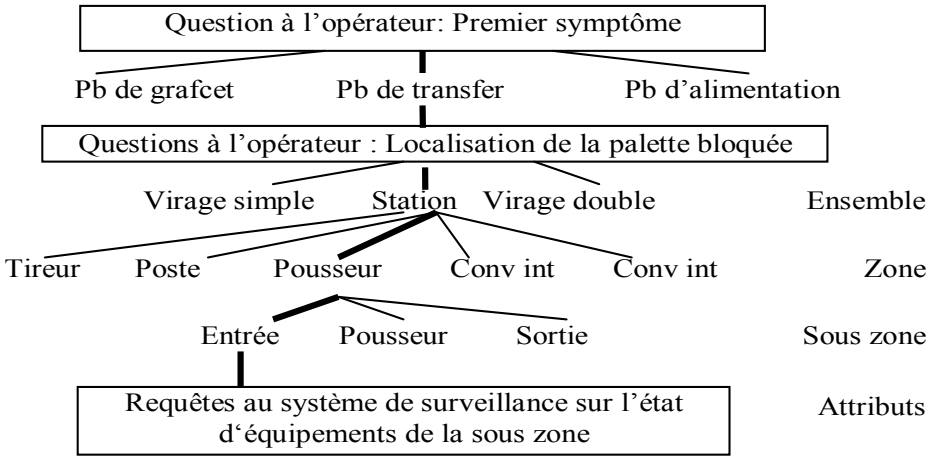


Figure 4.20. Le formulaire des questions

3.3. Remémoration des cas similaires

La phase de remémoration de cas similaires dépend essentiellement de la représentation des cas, de la structure de base de cas, des mesures de similarité et de l'exactitude de la réponse attendue. Pour permettre de comparer les cas les uns avec les autres, il faut pouvoir comparer leurs valeurs d'attributs de façon à établir à quels points ces valeurs sont proches. Chaque attribut doit donc être typé. C'est la connaissance du type qui permet de connaître les opérations de comparaison licites et par là d'établir des similarités. Notre système adopte la méthode du réseau de recherche de cas (**C**ase **R**etrieval **N**ets - **CRN**) présenté dans [Lenz, 1999].

3.3.1. Algorithme de recherche des cas similaires

La recherche des cas similaires se fait à l'aide de l'algorithme de réseau de recherche des cas [Rasovska et al., 2005a]. La recherche des cas similaires se fait par propagation des activations sur les variables du contexte déterminées par formulaire.

Formalisme associé au réseau de recherche de cas [Lenz, 1999]

$[E, C, \sigma, \rho]$ est un réseau de base où chaque nœud d'entité d'information représente un symptôme spécifique dans l'application avec

E est l'ensemble fini des nœuds d'entité d'information,

C est l'ensemble fini des nœuds de cas,

σ est la fonction de similarité décrivant la similarité entre les entités d'information,

ρ est la fonction de pertinence entre une entité d'information et un cas.

Algorithme pour la recherche des cas similaires

La recherche de cas similaires se fait par une activation des nœuds propagés dans le réseau comportant trois phases de CRN :

1. Activation initiale : est-ce que les caractéristiques sont présentes ?
2. Propagation des similarités : comparaison des valeurs de caractéristiques identifiées.
3. Propagation des pertinences : la pertinence de la caractéristique au nouveau problème posé.

Soit $q = \{q1,...,qn\}$ description d'un nouveau problème (une requête) :

1. Activation des nœuds d'information pour q :
   $\alpha 0 = 1$ pour les nœuds décrits dans la requête *q*
   $\alpha 0 = 0$ sinon
2. Propagation des similarités entre les nœuds d'information :
   α0 est propagées à tous les nœuds d'information du réseau :
   $\alpha_1(e) = \pi_e(\sigma(e_1, e) \bullet \alpha_0(e_1), ..., \sigma(e_s, e) \bullet \alpha_0(e_s))$
3. Propagation des pertinences pour les nœuds de cas:
   Résultat de la propagation des similarités est propagé à tous les nœuds de cas du réseau :
   $\alpha_2(e) = \pi_c(\rho(e_1, c) \bullet \alpha_1(e_1), ..., \rho(e_s, c) \bullet \alpha_1(e_s))$

Les connaissances de base sont les entités d'information représentées comme les termes dans la structure ontologique auxquels nous donnons des valeurs admissibles. Un cas est un ensemble de ces entités d'information qui sont connectés par des arcs de similarité et les cas sont accessibles à partir de ces entités à travers des arcs de pertinence

3.3.2. Mesures de similarité

Les mesures de similarité dans notre système sont adaptées à la représentation orientée objet et à la représentation de cas dans la base de cas. Dans notre base de cas on peut constater que les cas n'ont pas toujours le même nombre de descripteurs. Nous partons du principe que dans le cas source ne sont présents que les descripteurs pertinents. Nous en tenons compte dans la similarité de présence $sim_{Présence}$. Nous évaluons également la similarité de la classe de deux descripteurs

correspondants sim_{Classe} faisant partie de l'ontologie des descripteurs. Finalement nous comparons les valeurs de ces descripteurs par une similarité $sim(c_i,s_i)$. Nous agrégeons ces trois similarités dites locales dans une similarité σ entre deux cas. La similarité $\sigma = Sim(Cible, Source)$ représente la similarité globale entre deux descriptions du cas source et du cas cible :

$$\sigma = Sim(Cible, Source) = \frac{\sum_{i=1}^{p} \omega_i sim_{\text{Présence}} sim_{Classe} sim(c_i, s_i)}{\sum_{i=1}^{p} \omega_i}, \text{ où}$$

ωi est le poids de descripteur i ;

p et le nombre d'attributs ;

$sim_{Présence}$ est la similarité de présence du descripteur i qui prend la valeur :

1 lorsque le descripteur est présent à la fois dans la description du cas source et du cas cible,

0 lorsque le descripteur n'est pas présent dans la description de deux cas ;

sim_{Classe} est la similarité de deux descripteurs c_i du cas cible et s_i du cas source dans la typologie des descripteurs calculée pour le descripteur i ;

$sim(c_i,s_i)$ est la similarité locale des valeurs de deux descripteurs c_i du cas cible et s_i du cas source calculée pour le descripteur i. La similarité locale $sim(c_i,s_i)$ pour des valeurs symboliques peu être, par exemple :

$$sim(c,s) = \begin{cases} 1 & pour\ c = s \\ 0 & pour\ c \neq s \end{cases}$$

L'expert définit l'ensemble des similarités de classes pertinentes sim_{Classe} où *Classe* est une classe de descripteur définie dans l'ontologie (cf. fig. 4.13). La similarité sim_{Classe} est une fonction $sim_C : C \times C \rightarrow [0...1]$ où C est l'ensemble des classes.

1. L'expert peut ne définir que les similarités spécifiques et laisser l'héritage objet propager les définitions.
2. Les similarités peuvent être définies en utilisant des fonctions génériques comme celles de [http://www.diagnosticstrategies.com/similarity.htm] ou définir des similarités particulières plus pertinentes pour certaines classes.

Un exemple d'une telle similarité peut être la mesure pour les taxonomies (hiérarchie) :

$$sim(a,b) = \frac{h(commonnode(a,b))}{h}, \text{ où}$$

h est le nombre de niveaux de l'arbre taxonomique (dans l'hiérarchie).

Un exemple de recherche des cas similaires est donné à la figure 4.21. Dans la phase d'élaboration nous avons identifié une nouvelle description : problème de transfert, palette bloquée

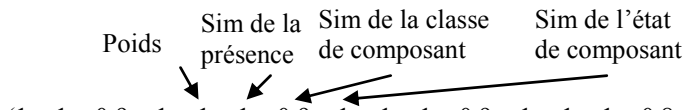
à station.actionneur.entrée. Le système envoie des requêtes pour connaître les états et les modes de fonctionnement des composants de ce contexte.

Dans la phase de remémoration, une fonction liste tous les cas de la partie « *problème de transfert* » et compare les attributs et leurs valeurs avec ceux de la nouvelle description. Nous avons considéré que tous les attributs sont de la même importance donc les ωi sont égales à 0. Nous avons évalué la similarité du cas cible avec les cas sources de la base de cas et illustré le calcul sur la figure 4.22. Quatre éléments de la similarité sont pris en compte pour chaque descripteur : le poids (importance), la présence du descripteur, la similarité de la classe des descripteurs et la similarité de leurs valeurs. Pour deux descripteurs *c* et *s* la similarité de la classe commune est illustré à la figure 4.23 et se calcule en remontant au premier concept commun de ces objets et en comparant les slots/attributs commun à ce niveau (la similarité est directement défini pour cette classe).

	Contexte				Liste des attributs–valeurs				Sim %	Compo défaillant	Classe
	Symptôme	Localisation									
Cib	**Pb de transfert**	**Station**	**Actionneur**	**Entrée**	**D1 :1**	**B0 :1**	**S1 :0**	**T :0**		**?**	**?**
C1	**Pb de transfert**	**Station**	**Actionneur**	**Entrée**	**D6 :1**	**B1 :1**	**S5 :0**	**P :0**	**87,5**	**Pousseur.P**	**Actionneur**
C2	Pb de transfert	Station	Actionneur	Entrée	D1 :1	B0 :1	S1 :1	T :0	75	Tireur.T	Actionneur
C3	Pb de transfert	Station	Actionneur	Entrée	D1 :1	B0 :1	S1 :0	T :1	75	Stoppeur.S1	Stoppeur
…	…		…		…	…	…	…	…		…
Cas	Pb de transfert	Station	Actionneur	Entrée	D1 :0					D1	Détecteur
Cas	Pb de transfert	Station	Actionneur	Sortie	D3 :0					D3	Détecteur
Cas	Pb de transfert	Station	Poste	Indexeur		B1 :1	S4 :1	I :0		Stoppeur.S4	Stoppeur
Cas	Pb de transfert	Station	Poste	Indexeur		B1 :1	S4 :0	I :0		Indexeur.I	Actionneur
Cas	Pb de transfert	Station	ConvPrinc	Sortie	D9 :0					D9	Détecteur

Figure 4.21. Résultat de la recherche des cas similaire dans la base de cas

Nous avons trouvé trois cas les plus similaires dans la base de cas (cf. fig. 4.21) et nous prenons celui qui est le plus similaire (à 87,5 %) avec le cas cible. L'adaptation se fera à l'aide de la typologie hiérarchique des composants du système (classe générale du composant défaillant).



(1 x 1 x 0.9 x 1 + 1 x 1 x 0.9 x 1 + 1 x 1 x 0.9 x 1 + 1 x 1 x 0,8 x 1) / 4 = 87.5 % pour (cible, cas 1)

(1 x 1 x 1 x 1 + 1 x 1 x 1 x 1+ 1 x 1 x 1 x 0 +1 x 1 x 1 x 1) / 4 = 75 % pour (cible, cas 2)

(1 x 1 x 1 x 1 + 1 x 1 x 1 x 1+ 1 x 1 x 1 x 1 +1 x 1 x 1 x 0) / 4 = 75 % pour (cible, cas 3)

Figure 4.22. Similarité du cas cible avec les cas sources

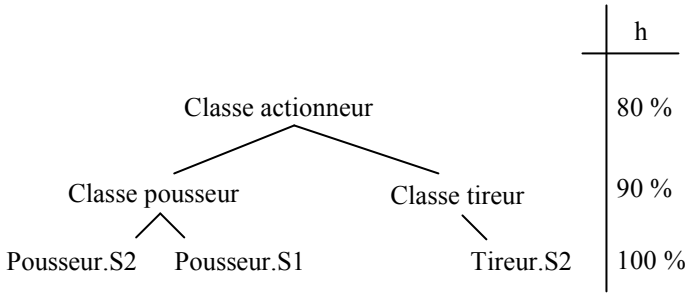


Figure 4.23. Exemple de la similarité des classes générales

3.4. Adaptation

Dans la phase d'adaptation, nous exploitons la hiérarchie de composants. Le système de transfert est constitué de 5 stations identiques ce qui nous permet de construire des classes générales pour adapter les solutions à chaque station. Nous dissocions deux niveaux d'adaptation générique utilisée dans notre système :

- ***l'adaptation substitutionnelle*** pour les ensembles des équipements proches (5 stations identiques),
- ***l'adaptation basée sur l'héritage*** dans l'ontologie des descripteurs (classes générales dans la base de cas).

Les modèles des connaissances générales (le découpage spatial, la typologie des composants du système) nous permettent d'analyser les cas constituant la base de cas actuelle selon les fonctions et la localisation. Ceci nous amène à extraire des « prototypes » des cas qui seront les représentants des fonctionnalités, à savoir la détection, l'action pneumatique, etc. Ces prototypes correspondent aux classes générales identifiées dans chaque cas pour le composant défaillant et seront à la base de l'adaptation pour les nouveaux problèmes à résoudre. Nous exploitons cette hiérarchie (typologie) de descripteurs pour substituer un descripteur donné par une classe générale et pour utiliser ses méthodes génériques associées. Cela revient à se baser sur les actions de réparation (la solution) de la classe générale qui sont adaptées en fonction des différences entre la classe générique et les cas particuliers.

Algorithme d'adaptation

L'adaptation se fait de la manière suivante :

Soit *s* le composant défaillant désigné par le cas source similaire et *c* le composant défaillant désigné par le cas cible. Le cas source indique que la solution consiste à appeler la méthode *« m »*. Deux possibilités se présentent :

1. Si *« m »* est une méthode virtuelle de la classe commune à *s* et *c*, alors la solution est adaptable, c'est *c.m( ).* La résolution de la méthode appelée est automatique par héritage.
2. Si *« m »* n'est pas une méthode n'appartenant pas à la classe commune, alors la solution n'est pas adaptable.

Dans notre exemple, le cas source le plus similaire est le cas source 1 ou le composant défaillant est le pousseur et donc la classe générale qui lui appartient est la classe actionneur.

- *Première phase de l'adaptation* : identification du composant défaillant est alors l'adaptation substitutionnelle (on utilise la classe générale du composant défaillant et on remplace l'instance de cette classe du cas source par l'instance du cas cible). Donc la solution du cas cible sera la classe générale - actionneur, le composant défaillant – tireur.

- *Deuxième phase de l'adaptation* : association de la méthode (action de réparation) au composant défaillant se fait par héritage à partir de la classe commune au composant définit source et au composant défini cible. L'action de réparation – autrement appelé la méthode est adapté au niveau de la classe générale – changer le vérin de l'actionneur (tireur). Par conséquent, nous identifions les informations complémentaires à cette action telles que la compétence requise, les outils et pièces de rechange nécessaires, la documentation disponible et la durée de l'intervention.

3.5. Révision et mémorisation

La révision de la solution proposée se fait manuellement par l'opérateur. Lorsque la solution n'est pas correcte, l'opérateur est censé proposer et décrire l'action qui à mené à la réparation de l'équipement. Ensuite l'expert doit corriger l'action dans le cas générique stocké dans la base de cas ou créer un nouveau cas concernant ce nouveau problème.

La base de cas est notablement réduite de 50 % (une vingtaine de cas) en utilisant les cas généraux (« prototypes ») et fait appel à des modèles et règles associées. L'évaluation de cette base de cas permet de vérifier la couverture des différents cas à partir des éléments prototypes, des modèles du contexte et des règles d'adaptation et fera partie de la maintenance de la base de cas.

Actuellement la base de cas contient à peu près une vingtaine de cas ce qui a permis de faire les premiers tests de recherche de cas similaires. Les limites et les contraintes de la mise en

application de la méthodologie proposée sont : la modélisation des données du domaine d'application nécessite une étude poussée et gourmande en temps mais elle est toutefois inévitable pour la mise en place de l'outil d'exploitation de ces données.

4. Conclusion

Nous avons présenté l'architecture de la plateforme d'intégration Proteus ainsi que le principe d'intégration d'un outil spécifique présenté comme un service Web. La plateforme de démonstration (le système de transfert de palette Sormel) a été présentée. Nous avons proposé une méthodologie pour l'élaboration du système d'aide au diagnostic et à la réparation basé sur le raisonnement à partir de cas. Un soin particulier a été apporté à l'élaboration de la base de cas par des techniques empruntées à la sûreté de fonctionnement ainsi qu'à l'étape d'acquisition de cas. La représentation de cas choisie relève de l'ontologie des connaissances du domaine qui emploie le formalisme de représentation orienté objet. Le modèle du domaine est ensuite associé au modèle des connaissances du raisonnement à partir de cas. Nous avons proposé une base de cas hiérarchique contenant plusieurs niveaux associés au découpage spatial du système de transfert, aux composants et leurs états ainsi qu'au niveau des modes de fonctionnement.

Nous avons proposé un algorithme de recherche des cas similaires ainsi que des mesures de similarité adapté à la structure de cas ainsi qu'à la structure hiérarchique de la base de cas et orienté vers l'adaptabilité des cas similaires. L'algorithme de recherche des cas similaires est composé d'une phase de recherche des cas similaires à l'aide de réseau de recherche des cas présenté par [Lenz, 1999], qui sélectionne un ensemble de cas, et d'un calcul de similarité permettant la phase de sélection de cas.

Nous avons proposé un algorithme d'adaptation composé de deux phases : identification du composant défaillant et association de la méthode de réparation à ce composant défaillant. L'algorithme est donc basé sur la typologie des composants, notamment sur la classe générale du composant défaillant identifié comme la solution de cas.

Nous avons montré la faisabilité de notre démarche sur une plateforme de démonstration d'e-maintenance assurant la maintenance d'un système de transfert de palettes Sormel.

Conclusion générale

Les travaux de recherche présentés dans ce mémoire ont porté sur la capitalisation des connaissances en maintenance industrielle à partir d'une démarche de capitalisation des connaissances articulée autour d'une méthodologie de raisonnement à partir de cas. Nous avons suivi l'objectif de proposer un service d'aide qui utilise et capitalise les connaissances et est intégré dans une plateforme distribuée d'e-maintenance.

Dans le premier chapitre, nous avons défini ce que sont une fonction et un service de maintenance, afin de pouvoir proposer un ensemble d'e-services de maintenance via une plateforme distribuée communiquant sur le web. Nous avons analysé le processus de maintenance afin de déterminer quatre niveaux d'applications imbriqués les uns dans les autres. A chaque niveau est associé un ensemble d'outils d'aide à la décision qui peut être agencé en modules intelligents concourant à la maintenance d'équipements et à la gestion des stratégies de maintenance.

Les nouvelles technologies de l'information permettent d'intégrer ces différents systèmes d'aide via des plateformes. Nous avons proposé une classification des différentes architectures des systèmes de maintenance en fonction de l'intensité des relations entre les systèmes (autonomie, communication, coopération, collaboration) dans une architecture particulière. Nous avons présenté un type d'architecture, la s-maintenance, que nous appelons la maintenance sémantique et qui prend appui sur la technologie émergeante du Web sémantique et sur des ontologies communes aux différents systèmes. Ces ontologies sont les témoins d'une démarche de gestion des connaissances.

Dans le deuxième chapitre, nous avons étudié la connaissance experte, mise en jeu lors de toute activité de maintenance par les acteurs du domaine, afin de proposer un e-service intelligent. Nous avons analysé les différentes approches de gestion et de capitalisation des connaissances ce qui nous a permis d'adopter une démarche de capitalisation des connaissances et d'élaboration d'une mémoire d'entreprise en maintenance. Notre démarche consiste à préserver les connaissances essentielles destinées à l'ensemble des systèmes d'aide à la décision dans la mémoire d'entreprise. Nous avons proposé une mémoire d'entreprise multi-niveaux correspondant aux quatre niveaux d'applications du processus de maintenance. Chaque niveau de la mémoire est spécifique en fonction des caractéristiques des systèmes d'aide.

Nous avons employé les méthodes de représentation et de modélisation des connaissances de l'ingénierie des connaissances pour structurer et modéliser les concepts identifiés dans l'analyse du système de maintenance. Ces concepts sous forme « d'expertise capitalisée » sont à la base de la mémoire d'entreprise et de notre système d'aide à la décision. La conception de ce système passe par la modélisation des connaissances qui se décline en un modèle de représentation associé à un modèle de résolution de problèmes. Nous avons adopté comme modèle de représentation une ontologie du domaine de maintenance. Le modèle de résolution de problèmes choisi correspond à une méthode de manipulation des connaissances, à savoir le raisonnement à partir de cas.

Dans le troisième chapitre, nous avons étudié le raisonnement à partir de cas (RàPC) dans le but d'élaborer un système d'aide au diagnostic et à la réparation. Nous avons présenté différentes typologies d'application ainsi que notre domaine d'application - le diagnostic et la réparation. Ensuite nous avons réalisé une étude comparative des différents systèmes de raisonnement à partir de cas existants appliqués au domaine du diagnostic technique. Nous les avons décrits puis comparés selon les différents critères correspondants aux phases de RàPC. Nous avons identifié deux types d'élaboration de système dans ce domaine, le premier type mis au point à l'aide d'un système de développement de RàPC et le deuxième associé à un système de représentation de connaissances.

Nous avons identifié certains verrous associés aux outils existants. Pour l'élaboration de notre système d'aide au diagnostic et à la réparation, nous avons proposé notre propre démarche de capitalisation des connaissances articulée autour du raisonnement à partir de cas guidé par les connaissances. Pour cela nous avons appréhendé le raisonnement à partir de cas à la fois comme une méthode nous permettant de manipuler les connaissances dans la mémoire d'entreprise et comme une méthode de résolution de problèmes pour notre système d'aide. Cela nous a permis de mettre en évidence l'objectif commun entre le raisonnement à partir de cas et la gestion des connaissances. Il s'agit de l'acquisition et de la réutilisation d'expériences ou des connaissances.

La première démarche de capitalisation des connaissances, nous a amené à proposer différentes connaissances dont une ontologie du domaine manipulée par le raisonnement à partir de cas. La deuxième démarche de raisonnement à partir de cas, nous a amené à développer un système d'aide au diagnostic où la phase de représentation de cas et de la base de cas est renforcée par la démarche de capitalisation. Nous avons alors associé ces démarches et nous les avons appliquées sur les deux premiers niveaux de la mémoire d'entreprise (l'analyse d'équipements puis le diagnostic et l'expertise) en élaborant notre système d'aide au diagnostic et à la réparation.

Dans le quatrième chapitre, nous avons présenté l'architecture de la plateforme Proteus distribuée générique d'e-maintenance intégrant les différents systèmes et applications existants en maintenance. Le concept de cette plateforme est défini par la description unique et cohérente de l'équipement à maintenir (une ontologie), par l'architecture générique basée sur les services Web

et en proposant des modèles et des solutions technologiques d'intégration. Le projet Proteus permet de faire bénéficier les industriels de bases de données techniques s'appuyant sur le retour d'expérience. La plateforme s'appuie sur un système de connaissances, créé à partir d'informations issues des différents niveaux de maintenance. Le portail Web développé dans la plateforme joue le rôle d'intégrateur de services capables de se connecter aux autres systèmes, propres aux entreprises. Afin d'intégrer notre propre système d'aide, nous avons présenté le principe d'intégration d'un outil spécifique présenté comme un service Web. Nous avons développé ce système dans la deuxième partie du quatrième chapitre.

Nous avons présenté notre méthodologie d'élaboration du système d'aide au diagnostic et à la réparation basée sur l'expertise capitalisée. Un soin particulier a été apporté à l'élaboration du cas et de la base de cas par des techniques empruntées à la sûreté de fonctionnement. Nous avons choisi des mesures de similarité et un algorithme de recherche des cas similaires tenant compte de la structure de cas et de la base de cas et orientés vers l'adaptabilité des cas similaires. Nous avons intégré notre système dans la plateforme d'e-maintenance Proteus et nous avons montré la faisabilité de notre démarche sur une plateforme de démonstration assurant la maintenance d'un système de transfert de palettes.

Enfin, nous avons proposé, dans cette thèse, le concept de mémoire d'entreprise basée sur une ontologie commune du domaine d'expertise associée à l'ontologie de l'entreprise. Cette proposition s'articule autour du cycle général de capitalisation qui participe à la création de la mémoire d'entreprise. Plus précisément, elle s'articule autour du cycle de raisonnement à partir de cas qui permet d'enrichir et d'actualiser continuellement cette mémoire d'entreprise. Nous avons suivi une démarche de capitalisation des connaissances pour l'élaboration d'un service de diagnostic.

Dans notre système d'aide au diagnostic et à la réparation, nous avons pallié le manque d'expertise dans la phase d'élaboration de cas et de la base de cas par l'utilisation des techniques de sûreté de fonctionnement (AMDEC, arbres de défaillances…). Cela permet de construire les connaissances sur le diagnostic et la réparation et de les modéliser sous forme de cas. Ensuite, elles sont liées aux connaissances du contexte et manipulées grâce au cycle de raisonnement à partir de cas.

De nos travaux se dégagent plusieurs perspectives.

Concernant notre outil d'aide au diagnostic et à la réparation dédié à un équipement précis, dans un premier temps, nous envisageons de formaliser les modèles proposés pour les adapter à d'autres équipements industriels suivant les typologies à déterminer. Cela consiste à fournir une base de connaissances commune avec une ontologie classifiant tout type de composants des équipements industriels. Un tel système pourra être exploité dans différentes industries.

Premièrement, nous envisageons de prendre en compte les composantes de fiabilité, criticité et temps dans la recherche des cas similaires. Cela consiste à évaluer ces variations au niveau de la description des cas et à les intégrer dans le calcul de similarité. Nous pouvons ainsi évoquer un outil mathématique basé sur les méthodes statistiques correspondantes.

Deuxièmement, l'évolution du système dans le temps doit être étudiée. La phase de révision et de mémorisation, donc l'apprentissage, doit être développée d'une manière automatique.

Une autre piste consisterait à traiter plusieurs palettes à la fois dans le système de transfert. Dans cette optique nous seront amenés à traiter les systèmes de séquences où les relations de dépendances entre les problèmes cibles s'avèrent primordiales pour la résolution du problème. Il s'agit, plus particulièrement, de cibler quel problème résoudre en premier parmi un ensemble de problèmes simultanés. Pour cela nous envisageons de tenir compte des travaux récents traitant des séquences temporelles en RàPC.

En ce qui concerne notre méthodologie d'élaboration d'un système d'aide, basée sur la capitalisation des connaissances, articulée autour du cycle de raisonnement à partir de cas, certaines pistes de perspectives se dégagent.

Nous envisageons de formaliser notre démarche afin de la proposer à d'autres systèmes d'aide à la maintenance. Plus particulièrement, nous envisageons de développer deux autres champs d'application pour les outils d'aide, comme nous l'avons définit dans l'analyse du processus de maintenance. Il s'agit de la gestion des ressources et de la gestion des stratégies de maintenance.

Les connaissances à ces deux niveaux seront issues de l'expérience (retour d'expérience) du système d'aide au diagnostic et vont compléter la mémoire d'entreprise. Les indicateurs techniques et économiques issus des rapports d'intervention vont servir à la création des connaissances pour ces systèmes. Ce retour d'expérience participera également à l'optimisation du processus de maintenance aussi bien au niveau du préventif et prédictif qu'au niveau du correctif. Cette mémoire d'entreprise, basée sur l'ontologie commune et les modèles génériques des données associées aux systèmes intégrés vont garantir l'interopérabilité de ces systèmes dans le cadre d'une plateforme d'e-maintenance.

En tenant compte de la technologie émergeante du Web sémantique et des techniques d'ontologie, nous avons pu faire évoluer le concept d'e-maintenance vers un nouveau concept de s-maintenance (maintenance sémantique) et proposer notre service d'aide dans cette optique. Notre perspective, dans cet axe, est de développer une plateforme de s-maintenance, en précisant finement son architecture. Ceci, en développant tous les systèmes d'aide de maintenance par une ontologie commune, en ce qui concerne l'équipement et des modèles de connaissance spécifiques à chaque domaine d'application. Une des perspectives possibles de la plateforme d'e-maintenance

de PROTEUS est la reconfiguration automatique de la plateforme d'e-maintenance basée sur des ontologies et sur les applications intégrées à la plateforme.

Annexe A

Politiques de maintenance

Nous présentons dans cette annexe les différentes stratégies de maintenance brièvement mentionnées dans le chapitre 1.

La maintenance productive totale (TPM) (Total Productive Maintenance) est une adaptation japonaise qui date de 1971 de la PM (Productive Maintenance) américaine développée aux Etats-Unis dans les années 50. La TPM est un système global et transversal qui implique la participation de toutes les divisions, notamment l'ingénierie, l'exploitation et la maintenance, et qui concerne tous les niveaux hiérarchiques.

Cette politique comprend l'automaintenance, c'est-à-dire la participation des exploitants à certaines tâches de maintenance et a pour objectif de réaliser le rendement maximal des équipements (indicateur de performance d'un équipement). Elle a été implantée par exemple chez Toyota au Japon, en France chez Renault comme la méthode MEG (maintenance exploitation globale) ou chez Sollac comme la « topomaintenance » [Monchy, 2000].

La maintenance basée sur la fiabilité (MBF) (RCM – Reliability Centered Maintenance) a été introduite aux Etats-Unis dans les années 60. La MBF est une méthode destinée à établir un programme de maintenance préventive utilisant une méthode d'analyse structurée permettant d'obtenir des niveaux très élevés de fiabilité des équipements critiques. L'objectif principal est d'améliorer la disponibilité des équipements sélectionnés comme critiques par leur influence sur la sécurité, sur la qualité et par leur impact sur les flux de production.

La méthode s'appuie sur une démarche de type AMDEC (Analyse des Modes de Défaillance, et de leurs Effets Critiques) et des matrices de criticité pour hiérarchiser les équipements, puis les causes de défaillances [Monchy, 2000]. Cette politique a été utilisée en aéronautique aux Etats-Unis, chez EDF en France et la démarche a été simplifiée et adaptée à l'industrie par l'ADEPA (Association pour le développement de la productique et de l'automatisation) en France.

Maintenance basée sur le risque (MBR) (RBM – Risk-Based Maintenance) utilisent les mêmes méthodes et analyses que la MBF mais en diffère par le calcul de la criticité des équipements. Il consiste à comprimer les dépenses de maintenance au plus proche du seuil de survenance des risques indésirés. La MBR suppose une parfaite maîtrise préalable de la MBF.

Les principes permettent de fonder des décisions liées à la maintenance sur le risque que ces décisions représentent, et sur l'impact de ces décisions sur la fiabilité des équipements et sur les coûts. Cette politique a été implantée par exemple chez Toshiba (les turbines à gaz). Elle est récemment développée dans le projet Européen RIMAP (Risk-Based Inspection and Maintenance Procedures for European Industry) lancé en 2001.

La politique d'intégrer la maintenance à la conception cherche à atteindre la situation idéale où les équipements devraient être conçus de façon à minimiser les opérations de maintenance voir à les rendre inutiles. L'idée de base est de développer simultanément, à toutes les étapes, la conception fonctionnelle du produit « équipement » et les éléments de soutien correspondants par la sûreté de fonctionnement et le soutien logistique intégré dans un plan de maintenance. Le développement de ce plan dés la conception d'un équipement n'est possible que par une exploitation d'un retour d'expérience.

Pour l'enjeu économique, le concept du LCC (Life Cycle Cost) ou coût du cycle de vie en français dont la définition selon CEN est « ensemble des coûts engendrés pendant le cycle de vie d'un bien » est au cœur de cette politique. La réflexion économique repose sur l'idée que le coût consenti à l'acquisition d'un équipement doit être moins élevé que le gain espéré sur le cycle de son exploitation [Monchy, 2000].

Le benchmarking est plutôt un outil d'amélioration. C'est une démarche continue et systématique permettant à une entreprise d'évaluer et de comparer ses propres performances en termes de qualité, de productivité et de pratique aux meilleures performances démontrées dans l'industrie en général (benchmarking généralisé) ou dans des entreprises qui ont des processus de production similaires (benchmarking sectoriel ou compétitif).

Cette politique est née aux Etats-Unis. Une expérience de benchmarking a été menée au niveau européen dans le projet TOMAS (Total Maintenance System for Foundry Operation) [Monchy, 2000]. La démarche a consisté en une analyse qualitative comparée des services internes de maintenance et en une analyse quantitative de ratios prédéterminés évaluant les performances de la maintenance et a permis de mettre en évidence les points faibles.

Récemment la politique du ***Maintien en Conditions Opérationnelles*** des systèmes de production s'est mise en place en garantissant tout au long de la phase d'exploitation la finalité attendue du système par une maîtrise de ses performances [Muller, 2005]. Cette politique met en œuvre un ensemble d'activités et de moyens logistiques (ravitaillement, acquisition et gestion des

rechanges, opérations de maintenance, outillages, documentation, formation…) en respectant trois principes fondamentaux : conserver une vision globale de la performance du processus de production, assurer une collaboration efficace entre les acteurs concernés, adopter une démarche de pilotage proactive afin de respecter la dynamique du processus de production. Comme le terme MCO est issu du domaine militaire un exemple de la société qui applique cette politique est la DCN (Direction des Chantiers Navals).

Annexe B

Modélisation UML

Nous présentons dans cette annexe la modélisation UML (Unified Modeling Language) ce qui est un langage pour la représentation des constructions des systèmes complexes. Ce langage a été proposé suite à une requête lancée par l'OMG (Object Management Group) concernant une méthode orientée objet standard. Cette annexe s'inspire de la présentation des principaux diagrammes d'UML dans [Guichet, 2003] et présente plus en détail le formalisme de représentation statique – le diagramme de classes et le diagramme d'objets.

Les diagrammes des cas d'utilisation décrivent la façon dont le système sera utilisé. Ils montrent les relations entre les acteurs et les cas d'utilisation du système. [1992] définit les acteurs comme les entités interagissant avec le système (les acteurs sont en général des utilisateurs ou des dispositifs extérieurs), et un cas d'utilisation comme une manière spécifique d'utiliser le système.

Les diagrammes de séquence décrivent une interaction entre plusieurs objets organisée dans le temps. Cette interaction est un ensemble de messages échangés entre les objets pour effectuer une opération ou obtenir un résultat.

Les diagrammes de collaboration présentent l'interaction organisée autour des objets et de leurs liaisons. A la différence des diagrammes de séquence, ils ne montrent pas le temps comme une dimension séparée. Ces diagrammes, ainsi que les diagrammes de séquence, sont appelés diagrammes d'interaction.

Les diagrammes de classes expriment de manière générale la structure statique d'un système, en termes de classes et de relations entre ces classes.

Les diagrammes d'objets sont des instances des diagrammes de classes, utilisés pour présenter un contexte particulier du problème.

Les digrammes d'états montrent le comportement des classes. Ils sont basés sur les « statecharts » définis par Harel [1987]. Ce sont des automates hiérarchiques, permettant une certaine représentation du parallélisme.

Les diagrammes d'activités sont des variantes des diagrammes d'états, organisés par rapport aux actions (ou opérations). Ils sont utilisés pour représenter le comportement interne d'une méthode ou d'un cas d'utilisation.

Les diagrammes de composants sont utilisés pour modéliser les différents composants du système et leurs relations. Ces composants peuvent être des modules, des sous-programmes, des tâches et des sous-systèmes.

Les digrammes de déploiement montrent l'organisation des composants matériels et le rattachement du logiciel aux dispositifs matériels (ou nœuds).

Dans la deuxième partie de cette annexe, nous présentons le formalisme de représentation statique des différents concepts (classe, instance, héritage et association) permettant de modéliser (représenter) les connaissances du domaine. Deux diagrammes permettent de représenter les connaissances, à savoir le diagramme de classe et le diagramme d'objets.

Le premier représente une structure de données générales et permet de modéliser les connaissances génériques du domaine. Cela rejoint une ontologie du domaine qui est une représentation de concepts partagée par une communauté d'experts pour exprimer des informations sémantiques consensuelles.

Le diagramme d'objets correspond au modèle de connaissances. Il s'agit d'une instanciation du diagramme de classes. Pour ce qui suit, nous nous sommes inspirés des travaux de M. Ruet [2002] pour présenter les éléments de ces diagrammes.

Une classe est une structure représentant une information générique et statique. Une classe définit des structures et des comportements communs. La structure est caractérisée par une liste d'attributs et le comportement par une liste d'opérations (ou méthodes) que peut réaliser la classe.

Un attribut est une propriété commune à tous les objets d'une classe. Il est attaché à une classe et possède un nom, un type et, éventuellement, une méthode d'évaluation. Le type d'attribut peut être un nombre, une chaîne de caractères ou une classe.

Une opération est une fonction ou une transformation qui peut être appliquée aux objets d'une classe.

Une méthode est l'implémentation d'une opération sur une classe et elle peut avoir des paramètres. Un exemple général d'une classe est présenté à la figure B.1.

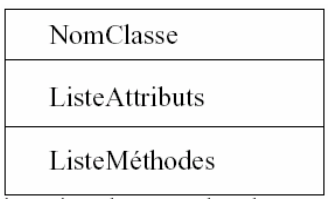


Figure B.1. Exemple général d'une classe

La notion d'***héritage*** permet de spécialiser une description selon différentes classes. Le terme de généralisation est également utilisé. Ainsi, des classes générales peuvent être reliées à des classes spécifiques. Ces dernières héritent de toutes les propriétés contenues dans la classe générale. La relation de généralisation peut être qualifiée de relation « sorte-de » ou « est-un ». Plusieurs niveaux d'abstraction peuvent ainsi être représentés. Dans l'exemple de la figure B.2 la classe « Equipement » généralise les classes « *Convoyeur* », « *Pousseur* » et « *Station* ».

L'héritage multiple correspond à la situation où une classe hérite de plusieurs classes. Cela peut poser des problèmes pour la définition d'un attribut de même nom hérité de deux classes différentes.

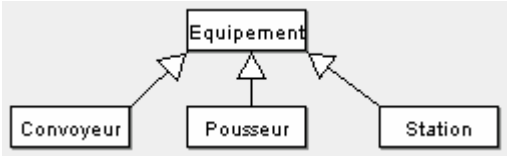


Figure B.2. Exemple d'héritage

Une association est une relation structurelle qui relie plusieurs classes. Une association peut avoir un nom ainsi que des rôles : il s'agit des rôles joués par chacune des classes dans l'association. Dans l'exemple de la figure B.3. « *est alimentée par* » est une association de la classe « *Station* » avec la classe « *Alimentation* ».

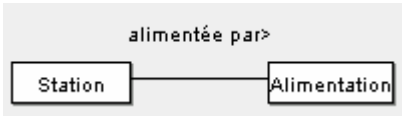


Figure B.3. Exemple d'une association

Une agrégation est une relation « composé-composant » ou « partie-de » dans laquelle les objets représentant les composants d'une chose sont associés à un objet représentant l'assemblage (ou l'agrégation) entier. L'exemple dans la figure B.4. exprime l'idée que le transfert « *Sormel* » est composé de « S*tation* ».

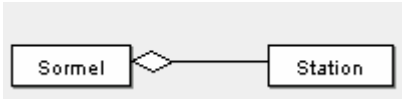


Figure B.4. Exemple d'une agrégation

Une composition est une variante de l'agrégation. La composition est une agrégation réalisée par valeur. Le principe pour les classes composantes est le même que si les classes composantes étaient des attributs de la classe composée. Un exemple est montré sur la figure B.5. La classe « *Etiquette* » est la composante de la classe « *Palette* ».



Figure B.5. Exemple d'une composition

Annexe C

Web sémantique et Web service

Nous présentons dans cette annexe les définitions de base concernant la technologie Web et plus particulièrement la technologie du Web sémantique.

Le Web peut être considéré comme un réseau de documents multimédias reliés entre eux par des liens hypertextes.

Il repose principalement sur le protocole de communication HTTP (HyperText Transfer Protocol) permettant d'accéder à des documents identifiés et localisés grâce à leurs URL (Unified Resource Locator).

HTML (HyperText Markup Language) est le langage de présentation généralement utilisé, permettant une mise en forme graphique des documents diffusés sur le Web.

Le W3C (World Wide Web Consortium) est l'organisme chargé d'établir les standards concernant les technologies relatives au Web.

Le Web sémantique associe les ressources du Web à une sémantique formelle, c'est-à-dire standard et conçue pour les machines, de façon à ce que leur contenu devienne manipulable par des programmes informatiques, dotés de mécanismes de raisonnements automatiques. Dans le but de permettre aux programmes informatiques de comprendre et de manipuler les informations selon la sémantique qu'elles transportent, il est nécessaire qu'ils disposent de connaissances, définissant de façon formelle les concepts mis en œuvre et les relations entre ces concepts. C'est le rôle des ontologies au sein du Web sémantique.

Le Web sémantique est typiquement décrit comme une architecture pour la propagation et la distribution de données/informations, de structures des données/informations et des données/informations sur l'articulation entre les ontologies.

Les trois composants du Web sémantique sont caractérisés comme suit :

- ***Les informations/données en HTML ou XML format***. HTML est un langage de balisage qui permet de décrire l'affichage des pages Web. Le langage de balisage XML (eXtensible Markup Language) est actuellement le plus utilisé et permet d'échanger des données structurées et de

décrire quelles structures sont valides. Mais XML ne s'intéresse qu'à la structure des informations et ne permet pas d'en exploiter la sémantique.

- ***Les structures des informations/données en format Web sémantique comme RDF*** (Resource Description Framework). Le RDF est un langage qui permet de définir et lier des ressources réseaux. Il définit les méta-données sur les documents du Web en déclarant des triplets (objet, prédicat, sujet) où les objets et les sujets sont des ressources et les prédicats des relations entre ces ressources. Le RDF permet non seulement de décrire les données mais également les règles pour l'interprétation de ces données.
- ***Les informations/données sur les articulations entre les ontologies***. Il s'agit d'une extension du RDF comme RDF schéma, Web Ontology Language (OWL) ou encore Darpa Agent Markup Language (DAML) qui permettent de comparer et lier des concepts. Ils représentent des ontologies de termes et de concepts dans une hiérarchie exprimant les relations entre ces termes et ces concepts.

Définition des services Web

D'après le W3C, ***un service Web*** est un logiciel conçu pour permettre des interactions de machine à machine sur un réseau. Un service Web a une interface qui est décrite d'une manière compréhensible pour un ordinateur comme, par exemple, WSDL (Web Service Description Language). Les autres systèmes interagissent avec ce service Web de la manière décrite par l'interface en échangeant des messages. Ces messages sont généralement transportés en utilisant http, leur contenu est décrit dans le format XML conjointement avec d'autres standards Web. Les logiciels écrits dans différents langages, fonctionnant sous différents systèmes d'exploitation peuvent coopérer en utilisant leurs services Web. OASIS et le W3C sont les principaux organismes responsables de l'architecture et des normes pour les services Web.

Deux technologies standard, toutes deux fondées sur XML, sont utilisées pour la mise en œuvre des services Web : Simple Objects Acess Protocol (SOAP) et Web Service Description Language (WSDL).

Le langage WSDL permet la description de services Web qui contient les informations essentielles à un client pour faire appel au service décrit, comme par exemple, son adresse Web, les méthodes qu'il offre et les paramètres de ces méthodes.

Le protocole SOAP permet d'utiliser les méthodes offertes par un service Web en suivant, par exemple, la description fournie dans un fichier WSDL mentionné ci-dessus.

Annexe D

Implémentation de l'outil sur la plateforme

Les différentes phases de l'intégration peuvent être décrites par les digrammes de séquence. Nous présentons ici l'intégration de notre outil d'aide au diagnostic et à la réparation basé sur le raisonnement à partir de cas. Ces digrammes ont fait l'objet de deliverable D2.3. dans le projet Proteus [Deliverable D2.3, 2004].

Phase de configuration

L'étape de base est la représentation et l'élaboration du cas basée sur l'analyse d'équipement qui nécessite la connexion du système de RàPC à d'autres sources de données comme SCADA, AMDEC, arbres de défaillance, etc. La base de connaissance est considérée comme un service de stockage pour l'outil de RàPC (la base de cas). Les différentes phases de l'étape de configuration sont décrites par les diagrammes de séquences. La phase de configuration est composée des parties suivantes :

- La configuration d'un nouveau modèle du cas contenant la configuration du modèle de cas, la configuration des mesures de similarité et des règles d'adaptation.
- La création d'un nouveau cas contenant la création de la description d'un nouveau cas et la création de la solution du nouveau cas.

Proposed name of the function	createNewModel
Inputs	Name: The name of the model to create
Outputs	A Boolean that indicates the model creation was successful.
Description	This service creates new model for an equipment.

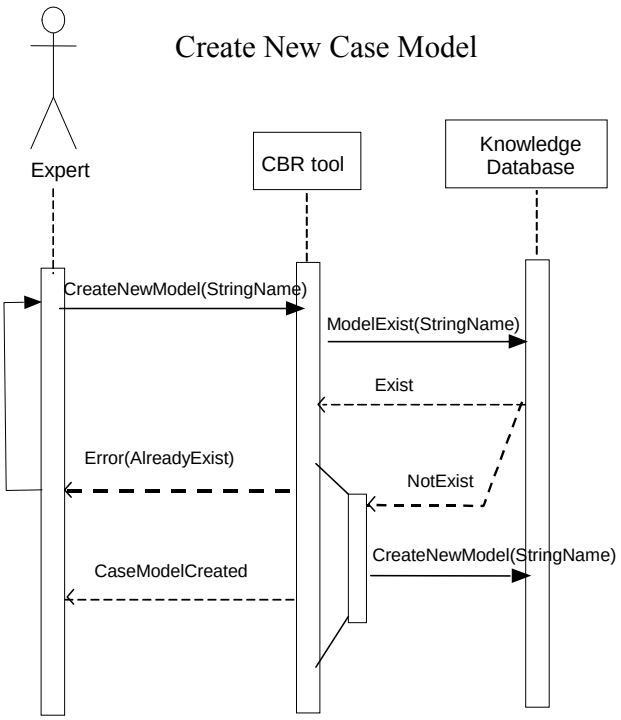


Figure D.1. Configuration d'un nouveau modèle du cas

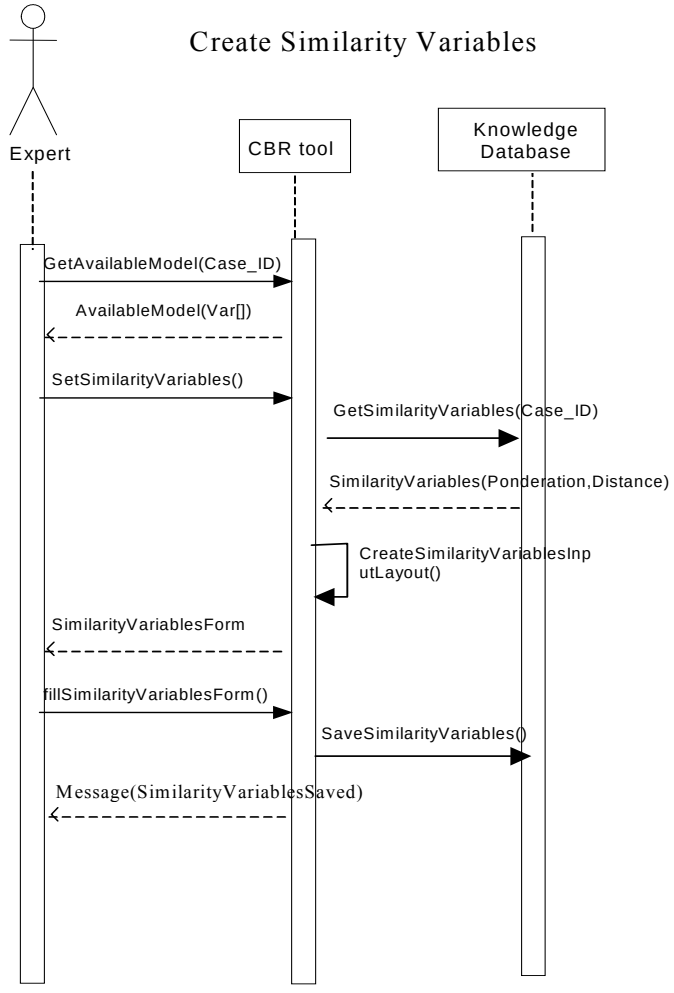


Figure D.2. Configuration des variables de similarité

Proposed name of the function	getAvailableModel
Inputs	Equipment_ID
Outputs	Case model: the requested model if existing
Description	Returns from the storage service a model.

Proposed name of the function	setSimilarityVariables
Inputs	Model_ID
Outputs	/
Description	This method selects the similarity variables that will be used in the recent case.

Proposed name of the function	getSimilarityVariables
Inputs	Model_ID
Outputs	Variables[]: ponderation calculus
Description	Returns from the storage service similarity variables like ponderation equations and measurement equation.

Proposed name of the function	saveSimilarityVariables
Inputs	Model_ID, Variables[]
Outputs	/
Description	Store the similarity variables in the storage service (Knowledge base).

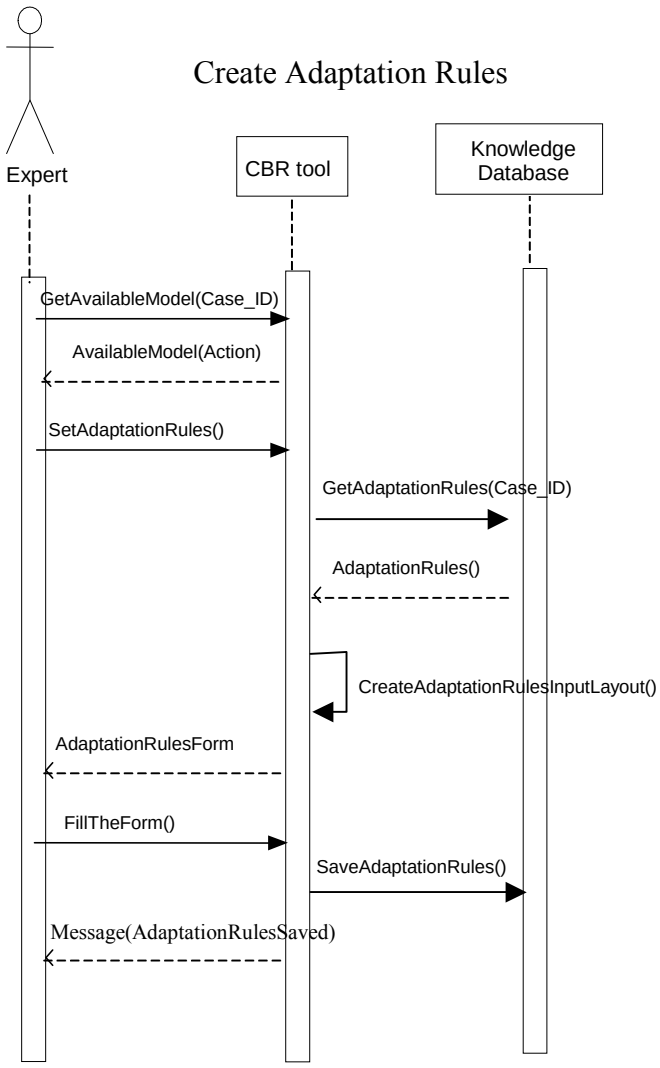


Figure D.3. Configuration des règles d'adaptation

Proposed name of the function	setAdaptationRules
Inputs	Model_ID
Outputs	/
Description	This method selects the adaptation rules that will be used in the recent case.

Proposed name of the function	getAdaptationRules
Inputs	Model_ID
Outputs	String[]: list of the adaptation rules.
Description	Returns from the storage service a model.

Proposed name of the function	saveAdaptationRules
Inputs	Model_ID, AdaptationRules[]
Outputs	/

Description	Store the adaptation rules in the storage service (Knowledge base).

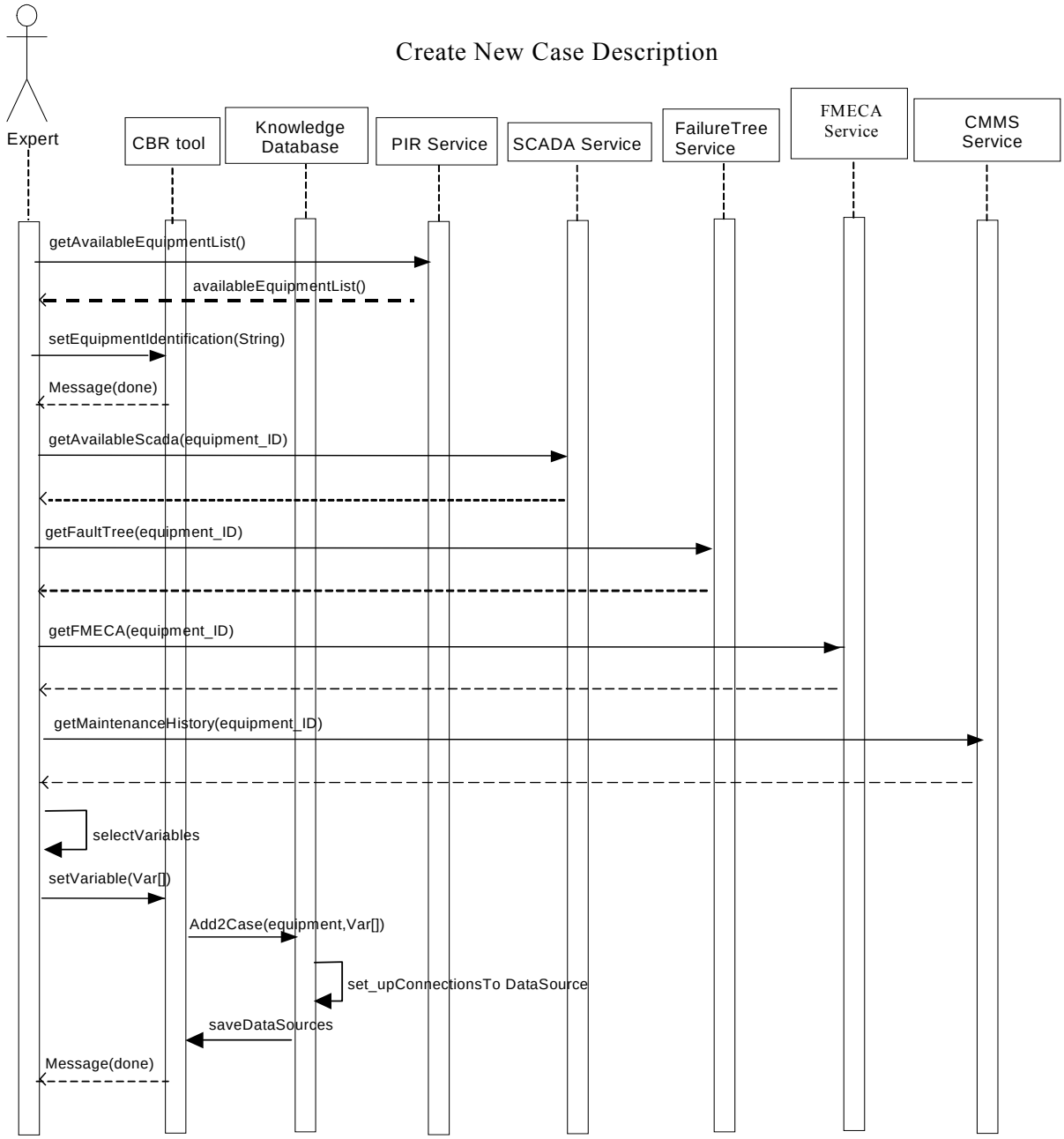


Figure D.4. Création d'un nouveau cas – partie description

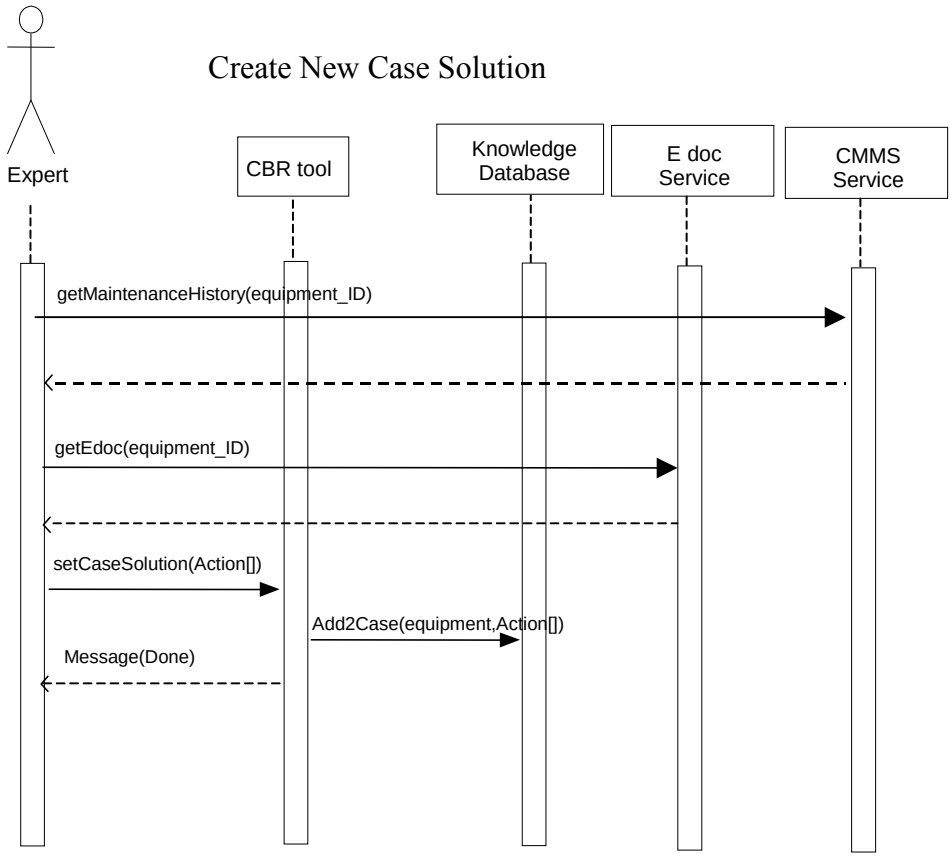


Figure D.5. Création d'un nouveau cas – partie solution

Proposed name of the function	setCaseSolution
Inputs	Model_ID, equipment_ID, case_ID
Outputs	Action[] Edocs[]: reference to e-documents like schemas etc.
Description	Store the action to do in the form of rule (change the detector "X") and send the reference to the e-document stored in the e-document server.

Phase d'initialisation

Pour l'outil de RàPC, cela consiste à choisir le modèle de cas selon l'équipement donné et en précisant les valeurs et les paramètres de la description du cas.

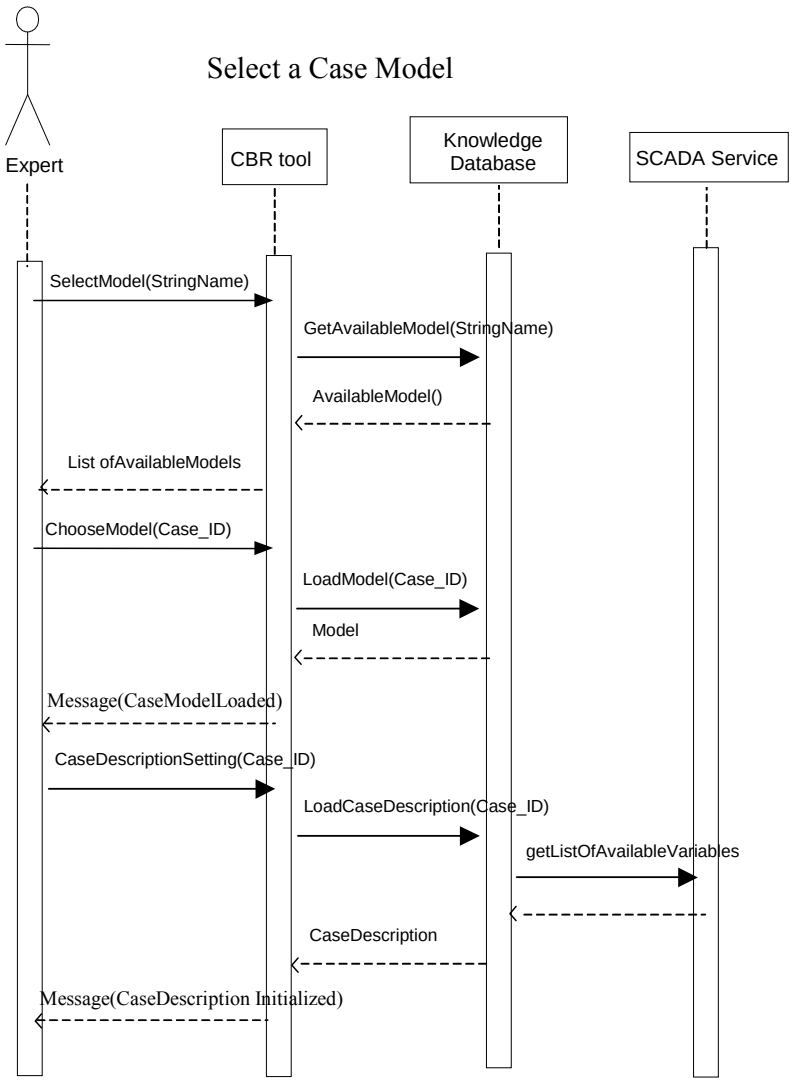


Figure D.6. Sélection du modèle de cas

Proposed name of the function	selectModel
Inputs	Equipment_ID
Outputs	/
Description	Choose the case model.

Proposed name of the function	caseDescriptionSetting
Inputs	Model_ID, equipment_ID
Outputs	/
Description	Identify the new case description.

Proposed name of the function	loadCaseDescription
Inputs	Case_ID
Outputs	DescriptionVariables[]
Description	Returns description variables from the storage service.

Phase de diagnostic

Cette phase a été déjà décrite dans le quatrième chapitre, elle correspond à l'algorithme de raisonnement à partir de cas.

Phase de mise à jour du modèle

Pour l'outil de RàPC le, modèle peut être mis à jour de trois manières différentes :

- mise à jour du modèle de cas,
- mise à jour des variables de similarité,
- mise à jour des règles d'adaptation.

Un cas de la base de cas peut également être révisé par un expert et une nouvelle solution est enregistrée comme un nouveau cas dans la base de cas lorsque la solution actuelle de cas ne mène pas à remédier la défaillance de l'équipement.

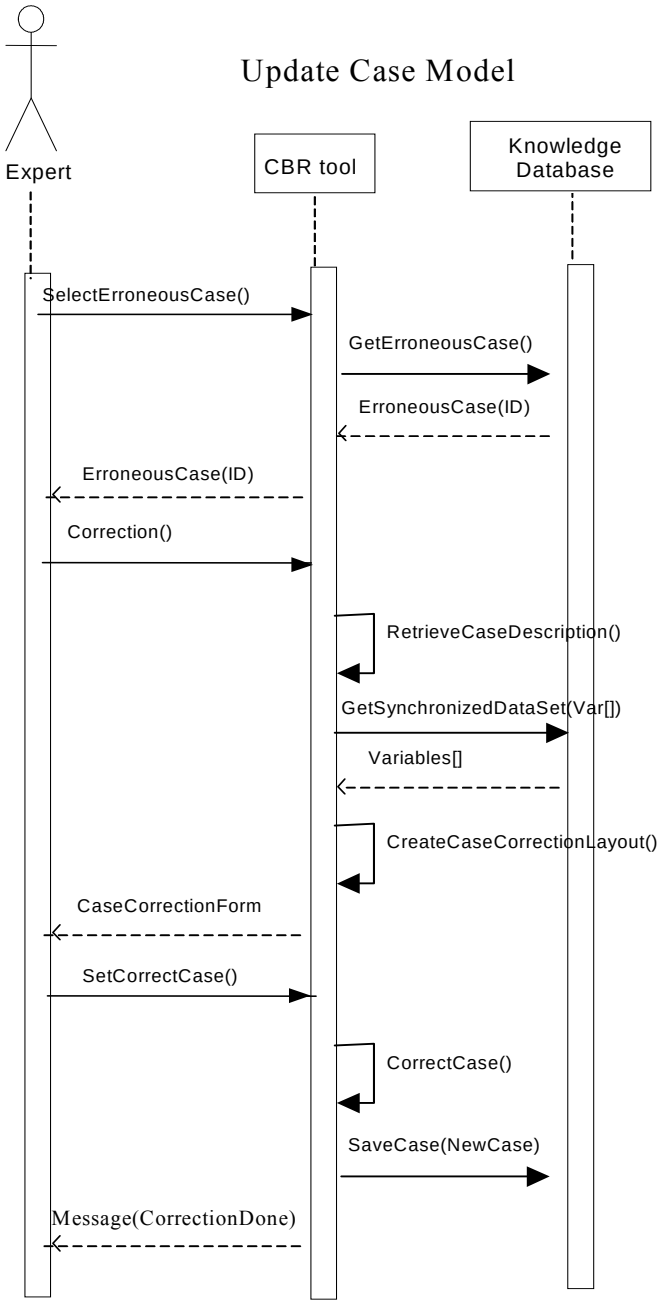


Figure D.7. Mise à jour d'un cas de la base de cas

Proposed name of the function	selectErroneousCase
Inputs	/
Outputs	Case_ID
Description	Select the case that does not correspond to the report.

Proposed name of the function	Correction
Inputs	Case_ID
Outputs	/
Description	

Proposed name of the function	setCorrectCase
Inputs	Case_ID
Outputs	DescriptionVariables[], CaseSolution[]
Description	

Proposed name of the function	saveCase
Inputs	Case_ID, DescriptionVariables[], CaseSolution[]
Outputs	/
Description	Store the new case into the storage service (database)

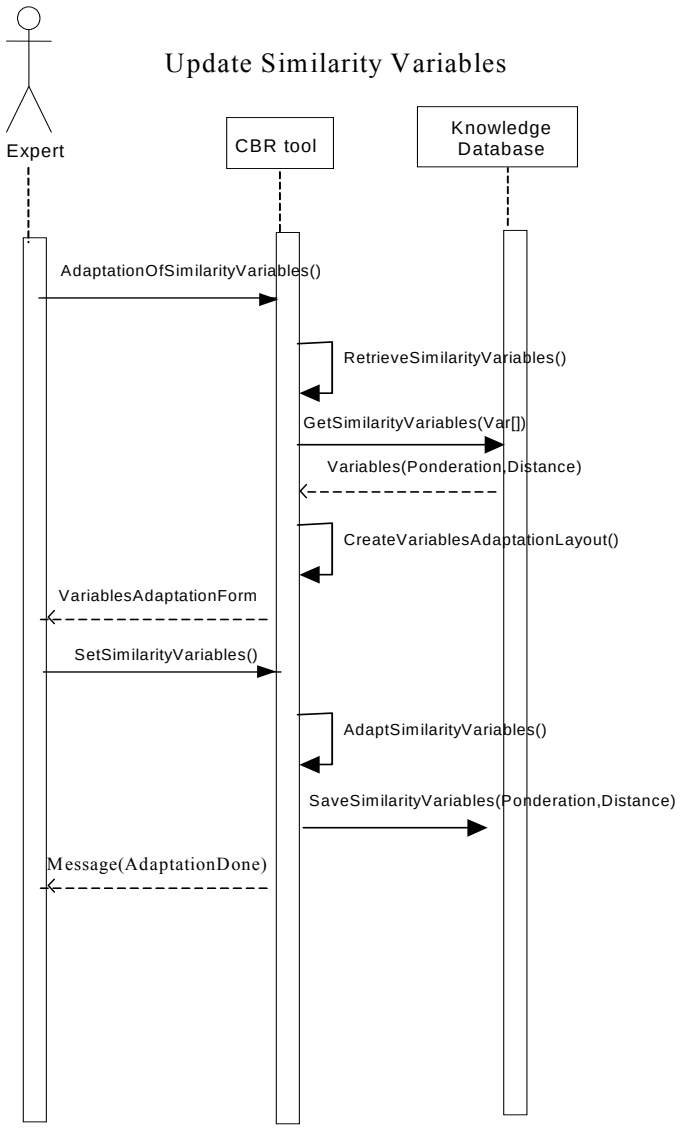


Figure D.8. Mise à jour des variables de similarité

Proposed name of the function	AdaptationOfSimilarityVariables
Inputs	Model_ID
Outputs	/
Description	Adapt the similarity variables.

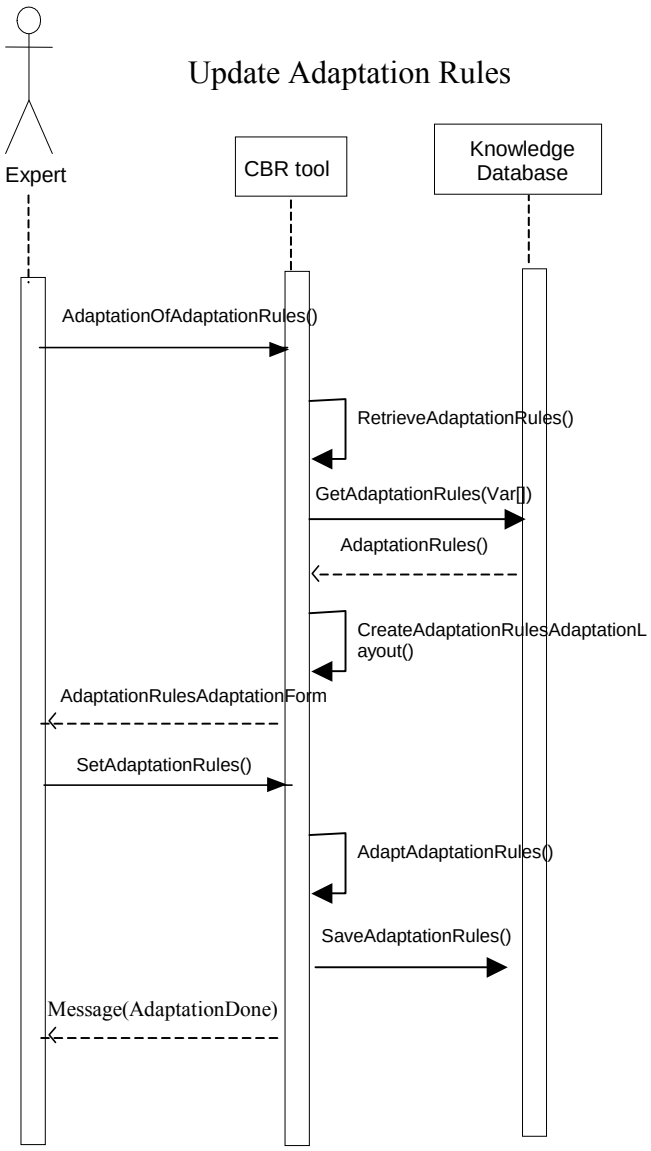


Figure D.9. Mise à jour des règles d'adaptation

Proposed name of the function	adaptationOfAdaptationRules
Inputs	Model_ID
Outputs	/
Description	Adapt the adaptation rules.

Interface homme machine

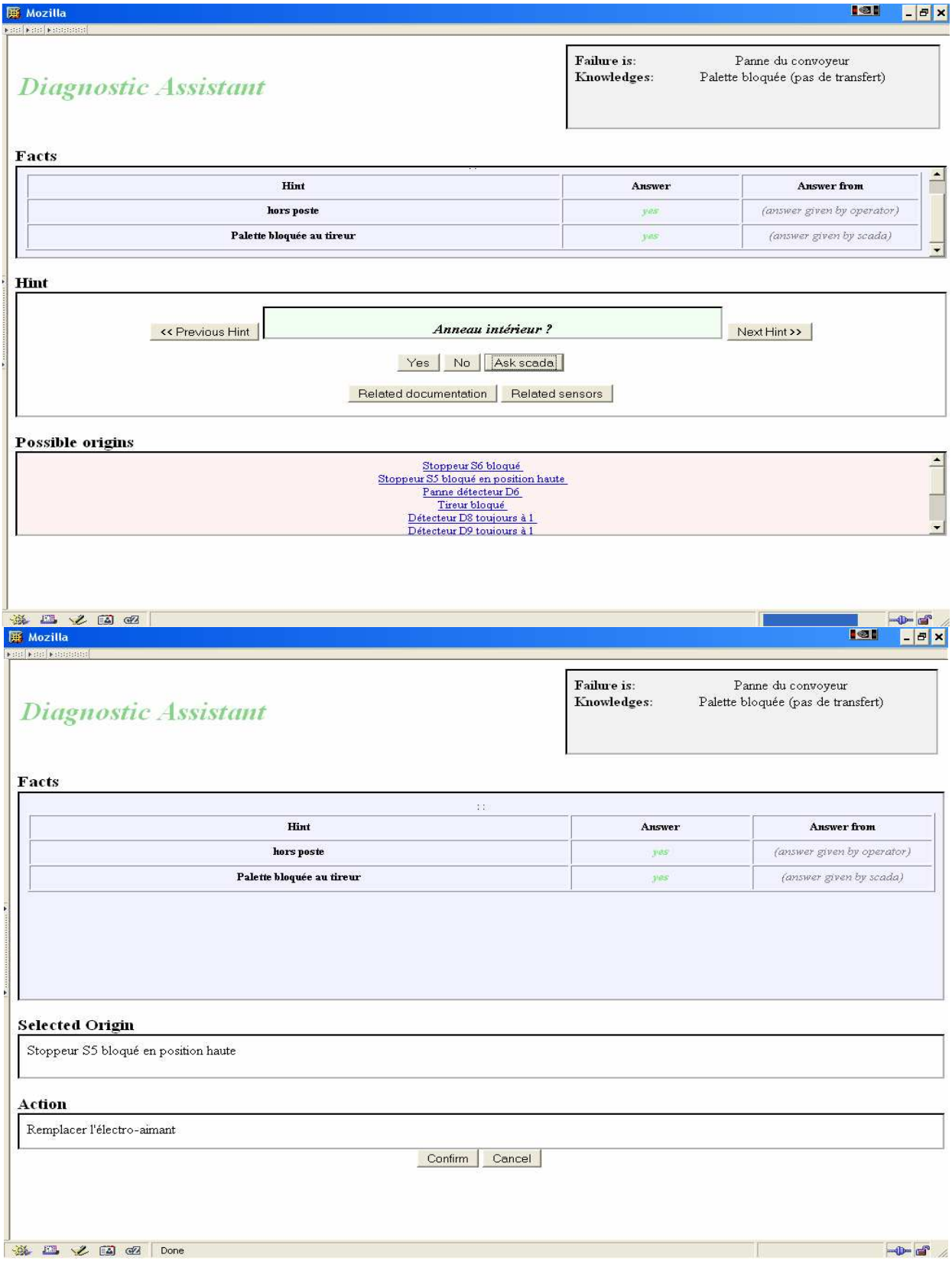


Figure D.10. Un aperçu des écrans de dialogue homme - machine

Références bibliographiques

[Aamodt, 2001] Aamodt A., Modeling the knowledge contents of CBR systems. Proc. of the Workshop Program at the 4th International Conference on Case-Based Reasoning, Vancouver, Canada, 2001.

[Aamodt, 2004] Aamodt A., Knowledge-Intensive Case-Based Reasoning and Sustained Learning. Proc. of the 9th European Conference on Artificial Intelligence, ECCBR'04, *Lecture Notes in Artificial Intelligence*, pp.1-15, Springer, 2004.

[Aamodt & Plaza, 1994] Aamodt A. et Plaza E., Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, 7(i): pp 39-59, 2004.

[Anor, 2001] Maintenance terminology. European standard, NF EN 13306, 2001.

[Aha et al., 1999] Aha D.W., Breslow L.A. et Muñoz-Avila H., Conversational Case-Based Reasoning. *Applied Intelligence*, Kluwer Academic Publishers, Boston, pp. 1-25, 1999.

[Alavi, 1997] Alavi M., KPMG Peat Marwick U.S. One Gain Brain, Harvard Business School (Case), July, 1997.

[Althoff et al., 1995] Althoff K., Auriol E., Bergmann R., Breen S., Dittrich S., Johnston R., Manago M., Traphöner R., Wess S., Case-based reasoning for decision support and diagnostic problem solving: The INRECA Approach. Proc. of the 3rd German Workshop on CBR, University of Kaiserslautern, 1995.

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

[Andersen, 1996] Andersen A., The Knowledge Management Assesment Tool: External Benchmarking Version, The American Productivity and Quality Center, Winter, 1996.

[Armengol et al., 1993] Armengol E. et Plaza E., Case-Based Reasoning at the Knowledge Level: An analysis of Chef. Proc. of the 1st European Workshop on Case-Based Reasoning, EWCBR'93, University of Kaiserslautern, Germany, pp. 290-295, 1993.

[Bachimont, 2000] Bachimont B., Engagement sémantique et engagement ontologique : Conception et réalisation d'ontologies en ingénierie des connaissances, *Ingénierie de connaissances*, pp. 305-323, 2000.

[Baizet, 2004] Baizet Y., La gestion des connaissances en conception. *Thèse de doctorat*, Université Joseph Fouriet, Grenoble, mars 2004.

[Bareiss et al., 1993] Bareiss E., Porter B. et Wier C., PROTOS: un système apprenti utilisant des cas typiques. *Apprentissage symbolique : une approche de l'intelligence artificielle*, Vol. 1, pp. 105-120, 1993.

[Bartlett, 1964] Bartlett F., Remembering (5th ed.). Cambridge University Press, 1964.

[Beckman, 1996] Beckman T., A Methodology for Knowledge Management. Proc. of the IASTED International Conference on AI and Soft Computing, 1997.

[Bello-Thomas et al., 2004] Bello-Thomas J.J., Gonzalez-Calero P.A. et Diaz-Agudo B., JColibri: An Object-Oriented Framework for Building CBR Systems. *Advances in Case-Based Reasoning*, Proc. of the 7th European Conference on Case-Based Reasoning, ECCBR 2004, pp. 32-46, 2004.

[Bergmann, 2001] Bergmann R., Highlights of the European INRECA Projects. Aha & Watson (Eds.), Proc. of the 4th International Conference on Case-Based Reasoning, ICCBR, Vancouver, Canada , 2001.

[Bergmann & Stahl, 1998] Bergmann R. et Stahl A., Similarity Measures for Object-Oriented Case Representations. *Advances in Case-Based Reasoning*, Proc. of the 4th European Workshop, EWCBR, Dublin, Ireland, 1998.

[Bergmann et al., 2003] Bergmann R., Althoff K.D., Breen S., Göker M., Manago M., Traphöner R. et Weiss S., Developing Industrial Case-Based Reasoning Applications: The INRECA Methodology. *Lecture Notes in Artificial Intelligence*, LNAI 1612, Springer-Verlag, Berlin, 2003.

[Berners-Lee et al., 2001] Berners-Lee T., Hendler J. et Lassila O., The semantic web. *Scientific American*, mai, 2001. http://www.sciam.com/article.cfm?article ID=00048144-10D2-1C70-84A9809EC588EF21.

[Bézivin, 2000] Bézivin J., De la programmation par objets à la modélisation par ontologies. *Ingénierie de connaissances*, pp. 209-222, 2000.

[Biébow & Szulman, 2000] Biébow B. et Szulman S., Une approche terminologique pour catégoriser les concepts d'une ontologie. *Ingénierie de connaissances*, pp.325-336, 2000.

[Börner, 1998] Börner, K. CBR for Design. Lenz M., Bartsch-Spörl B., Burkhard H.D., Wess S. (Eds.), *Case-Based Reasoning Technology: From Foundations to Applications*, Lecture Notes in Artificial Intelligence, Springer Verlag, 1998.

[Boucly, 1990] Boucly F., Le management de la maintenance assistée par ordinateur, Afnor, 1990.

[Bourne, 1997] Bourne. C., Catégorisation et formalisation des connaissances. Fouet(Coord.), *Connaissances et Savoir-faire en entreprise*, Hermès, pp. 179-197, 1997.

[Capponi, 1995] Capponi C., Identification et exploitation des types dans un modèle de connaissances à objet, *Thèse de doctorat*, Université Joseph Fourier, Grenoble, 1995.

[Carbonell, 1986] Carbonell J.G., Derivational Analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition. Michalski R., Carbonell J. et Mitchell T. (Eds.), *Machine Learning, an Artificial Intelligence Approach*, Morgan Kaufmann, Vol 2, pp. 271-391, 1986.

[Chandrasekaran, 1983] Chandrasekaran B., Towards a taxonomy of problem solving types. The AI Magazine, Winter/Spring, pp. 9-17, 1983.

[Charlet et al., 1996] Charlet J., Bachimont B., Bouaud J. et Zweigenbaum P., Ontologie et réutilisabilité: expérience et discussion. *Acquisition et ingénierie des connaissances : tendances actuelles*, 1996.

[Charlet et al., 2000] Charlet J., Zacklad M, Kassel G. et Bourigault D., Knowledge engineering: Recent evolutions and new challenges, Editions Eyrolles, Paris, 2000.

[Charlet, 2005] Charlet J., Les connaissances médicales à l'épreuve de l'informatisation : un exemple paradigmatique de l'IC ?, Séminaire C2EI, Paris, mars, 2005.

[Chebel-Morello et al., 2005] Chebel-Morello B., Rasovska I. et Zerhouni N., Knowledge capitalization in system of equipment diagnosis and repair help. Proc. of IJCAI'05 Workshop on Knowledge Management and Organizational Memories, Edinburgh, Scotland, 2005.

[Choo, 1996] Choo C., An integrated Information Model of the Organization : The Knowing Organization,1996. http://www.fis.utoronto.ca/people/faculty/choo/FIS/KO/KO.html1#contents

[Condamine & Rebeyrolle, 2000] Condamine A. et Rebeyrolle J., Construction d'une base de connaissances terminologiques à partir de textes : expérimentation et définition d'une méthode, *Ingénierie de connaissances*, pp. 225-241, 2000.

[Corbel, 1997] Corbel J.C., Méthodologie de retour d'expérience : démarche MEREX de Renault. *Connaissances et savoir-faire en entreprise*, Hermès, pp. 93-110, 1997.

[Cuignet, 2002] Cuignet R., Management de la maintenance : améliorez les performances opérationnelles et financières de votre maintenance. Dunod, Paris, 2002.

[d'Aquin, 2005] d'Aquin M., Un portail sémantique pour la gestion des connaissances en cancérologie. *Thèse de doctorat*, Université Henri Poincaré, Nancy, décembre, 2005.

[Davenport & Prusak, 1997] Davenprot T.H. et Prusak L., Information Ecology : Mastering the Information and Knowledge Environment. Oxford, New York, 1997.

[Dedun et al., 2005] Dedun I. et Seville M., Les systèmes d'information interorganisationnels comme médiateurs de la construction de la collaboration au sein des chaînes logistiques : Du

partage d'information aux processus d'apprentissages collectifs. Proc. du 6ième congrès international du Génie Industriel, Besançon, juin, 2005.

[Deliverable D2.3, 2004] Deliverable D2.3., Specification of the decision tool WP2, rapport de Proteus, PROTEUS-D-23-all-30072004, 2004.

[Demarest, 1997] Demarest M., Understanding Knowledge Management. *Long Range Planning*, Vol. 30, No.3, pp. 374-384, 1997.

[Despujols, 2004] Despujols A., Optimisation de la maintenance par la fiabilité (OMF). *Techniques de l'Ingénieur*, MT 9 310, 2004.

[Dieng et al., 1999] Dieng R., Corby O., Giboin A. et Ribière M., Methods and Tools for Corporate Knowledge Management. International Journal of Human-Computer Studies, 51 :567-598, Academic Press, 1999.

[Dubuisson et al., 2001] Dubuisson B., Boutleux E., Dague P., Denoeux T., Didelet E., Gandvalet Y. et Masson M., Diagnostic, intelligence artificielle et reconnaissance de formes, Hermès, 2001.

[Duribreux-Cocquebert & Houriez, 2000] Duribreux-Cocuebert M. et Houriez B., Application industrielle d'une approche mixte de modélisation des connaissances. *Ingénierie de connaissances,* pp. 25-41, 2000.

[Durstewitz, 1994] Durstewitz M., Report on Workshop on Corporate Memory. Toulouse, 1994. http://www.delab.sintef.no/MNEMOS/external-info/cm-eurisko.txt

[Eriksson, 2000] Eriksson A.E. et Penker M., Business Modeling with UML-Business Patterns at Work, John Wiley & Sons, New York, 2000.

[Ermine, 1996] Ermine J.L., Les systèmes de connaissances, Ed. Hermès, Paris, 1996.

[Ermine, 2001] Ermine J.L., Capitaliser et partager les connaissances avec la méthode MASK. *Traité IC2 (Information, Commande, Communication*), Volume Ingénierie et capitalisation des connaissances, Hermès, 2001.

[Fouet, 1997] Fouet J.M., Connaissances et savoir-faire en entreprise, intégration et capitalisation, Hermès, 1997.

[Francastel, 2001] Francastel J.C., Externalisation de la maintenance : Stratégies, méthodes et contrats. Dunod, Paris, 2003.

[Fréderic, 2003] Fréderic M., Mettre en œuvre une GMAO. Dunod, Paris, 2003.

[Fuchs et al., 2006] Fuchs B., Lieber J., Mille A. et Napoli A., Réflexions sur la place du RàPC dans trois domaines de recherche actuels. Actes du 14ième atelier du raisonnement à partir de cas, Besançon, mars, 2006.

[Fuchs et al., 2006] Fuchs B., Lieber J., Mille A. et Napoli A., Une première formalisation de la phase d'élaboration du raisonnement à partir de cas. Actes du 14ième atelier du raisonnement à partir de cas, Besançon, mars, 2006.

[Fuchs, 1997] Fuchs B., Représentation des connaissances pour le raisonnement à partir de cas : Le système ROCADE. *Thèse de doctorat*, Laboratoire Image Signal Acoustique de CPE Lyon, 1997.

[Grant et al., 1996] Grant P.W., Harris P.M. et Moseley L.G., Fault Diagnosis for Industrial Printers Using Case-Based Reasoning. *Engineering Applications of Artificial Intelligence*. Vol.9, No.2, pp.163-173, 1996.

[Grundstein & Barthès, 1996] Grundstein M. et Barthès J.P., An Industrial View of the Process of Capitalizing. Actes du Congrès ISMICK'96, Rotterdam, Pays Bas, pp. 258-264, 1996.

[Grundstein, 1994] Grundstein M., Développer un système à base de connaissances : un effort de coopération pour construire en commun un objet inconnu, CP2I, Paris, 1994.

[Grundstein, 1995] Grundstein M., La capitalisation des connaissances de l'entreprise, 1996.

[Grundstein, 2000] Grundstein M., Management des connaissances de l'entreprise: problématique, axe de progrès, orientations, 2000.

[Guiochet, 2003] Guiochet J., Maîtrise de la sécurité des systèmes de la robotique de service : Approche UML basée sur une analyse du risque système. *Thèse de doctorat*, Laboratoire d'Etude des Systèmes Informatiques et Automatiques, 2003.

[Hammond, 1986] Hammond K.J., CHEF: a model of case-based planning. Proc. of AAAI'86, Morgan Kaufman, pp. 267-271, 1986.

[Harel, 1987] Harel D., Statecharts: a visual formalism for complex systems. Science of Computer Programming, 8, pp. 231-274, 1987.

[Hinrichs, 1988] Hinrichs T.R., Towards an architecture for open world problem solving. Kolodner J. (Ed.), Proc. of Case-Based Reasoning Workshop (DARPA), Morgan Kaufmann, pp. 182-189, 1988.

[Holsapple & Joshi, 1999] Holsapple C.W. et Joshi K.D., Description and Analysis of Existing Knowledge Management Frameworks. Proc. of the 32nd Hawaii International Conference on System Sciences, IEEE, 1999.

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

[Journal, 2001] Le journal de la production. Mettez en place la nouvelle maintenance, N° 37, décembre, 2001.

[Kaffel, 2001] Kaffel H., La maintenance distribuée: concept, évaluation et mise en œuvre. Thèse de doctorat, Université Laval, Quebec, 2001.

[Kahn, 2003] Kahn J., Overview of MIMOSA and the Open System Architecture for Enterprise Application Integration. Proc. of COMADEM 2003, pp. 661-670, Växjö University, Sweden, 2003.

[Kaufman, 1987] Kaufman A., Nouvelles logiques pour l'intelligence artificielle. Editions Hermès, Paris, France, Mai, 1987.

[Koehler, 1996] Koehler J., Planning from Second Principles. *Artificial Intelligence*, 87, pp. 145-186. 1996.

[Kolodner, 1993] Kolodner J. L., Case-Based Reasoning. Morgan Kaufmann, 1993.

[Koton, 1988] Koton P., Reasoning about evidence in causal explanations. Proc. of AAAI'88, pp. 256-261, 1988.

[Kramer, 2003] Kramer I., Proteus - Modélisation terminologique, INRIA, 2003.

[Küssel et al., 2000] Küssel R., Liestmann V., Spiess M. et Stich V., „Teleservice" a customer-oriented and efficient service ? *Journal of Materials Processing Technology*, 107(1-3), pp. 363-371, 2000.

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

[Lamontagne & Lapalme, 2002] Lamontagne L. et Lapalme G., Raisonnement à base de cas textuels – état de l'art et perspectives. *Revue d'Intelligence Artificielle*, Hermès, Paris, 16(3), pp. 339-366, 2002.

[Laprie & al., 1995] Laprie J.C. et al., Guide de la sûreté de fonctionnement. Ouvrage collectif du Laboratoire d'Ingénierie de la Sûreté de fonctionnement, Cépaduès Editions, 1995.

[Laverde et al., 2003] Laverde A.M., Baragaño A.F. et Dominguez J.M., Knowledge processes: On overview of the principal models. Proc. of the 3rd European Knowledge Management Summer School, Sab Sebastian, Spain, 2003.

[Le Bozec et al., 2000] Le Bozec C., Jaulent M.C. et Zapletal E., IDEM: remémoration de cas pour l'aide au diagnostic en anatomie pathologique. *Ingénierie de connaissances*, pp. 370-386, 2000.

[Le Priol, 2000] Le Priol F., Extraction et capitalisation automatiques de connaissances à partir de documents textuels. SEEK-JAVA : identification et interprétation de relations entre concepts. *Thèse de Doctorat*, Paris, décembre, 2000.

[Leake & Wilson, 1998] Leake D.B. et Wilson D.C., Categorizing maintenance: Dimensions and directions. *Advances in Case-Based Reasoning*, Proc. of EWCBR-98, pp. 196-207, 1998.

[Leake & Wilson, 2000] Leake D.B. et Wilson D.C., Remembering why to remember: Performance-guided case-base maintenance. *Advances in Case-Based Reasoning*, Proc. of EWCBR-2K, Springer-Verlag, 2000.

[Leake et al., 1996] Leake D.B., Kinley A. et Wilson D., Linking Adaptation and Similarity Learning. Proc. of the 8th Annual Conference of the Cognitive Science Society, 1996.

[Lebold & Thurston, 2001] Lebold M. et Thurston M., Open standards for Condition-Based Maintenance and Prognostic Systems. Proc. of 5th Annual Maintenance and Reliability Conference (MARCON 2001), Gatlinburg, USA, 2001.

[Lenz et al., 1998] Lenz M., Auriol E. et Manago M., Diagnosis and Decision Support. Lenz M., Bartsch-Spörl, B., Burkhard, H.D. et Wess, S. (Eds.), *Case-Based Reasoning Technology: From Foundations to Applications*, Lecture Notes in Artificial Intelligence, Springer Verlag, 1998.

[Lenz, 1999] Lenz M., Case Retrieval Nets as a Model for Building Flexible Information Systems. *Thèse de doctorat*, Université de Humboldt, 1999.

[Leonard-Burton, 1995] Leonard-Burton D., Wellspring of Knowledge. Boston: Harvard Business School Press, 1995.

[Malvache & Prieur, 1993] Malvache P. et Prieur P., Mastering corporate experience with the Rex method. Proc. of ISMICK'93, Compiègne, octobre, 1993.

[Mark W. McElroy, 2002] Mark W., Simoudis E. et Hinkle D., Case-Based Reasoning: Expectations and Results. Leake, D.B. (Ed) *Case-Based Reasoning: Experiences, Lessons and Future Directions*, pp.269-294, MIT Press, 1996.

[Matta et al., 1999] Matta N., Corby O. et Ribière M., Méthodes de capitalisation de mémoire de projets. Rapport de recherche INRIA n° 3819, novembre, 1999.

[Matta et al., 2001] Matta N., Ermine J.L., Aubertin G. et Trivin J.Y., Knowledge. Capitalization with a knowledge engineering approach : the MASK method. Proc. of IJCAI'2001 workshop on Knowledge Management and Organizational Memory, 2001.

[McElroy, 2002] McElroy M., The New Knowledge management. *Complexity, Learning and Sustainable Innovation*. ISBN 0-7506-7608-6, Burlington, England, 2002.

[Mille, 1995] Mille A., Raisonnement basé sur l'expérience pour coopérer à la prise de décision, un nouveau paradigme en supervision industrielle. *Thèse de doctorat*, Université de Saint Etienne, 1995.

[Mille, 1999] Mill. A., Tutorial CBR : Etat de l'art de raisonnement à partir de cas. Plateforme AFIA'99, Palaiseau, 1999.

[Mille et al., 1996] Mille A., Fuchs B. et Herbeaux O., A unifying Framework for Adaptation in Case-Based Reasoning. Workshop on Adaptation in Case-Based Reasoning, ECAI-96, pp. 22-28, Budapest, Hungary, August, 1996.

[Mitchell et al., 1998] Mitchell J., Bond T., Bever K. et Manning N., MIMOSA – Four Years Later. Sound and Vibration, pp. 12-21, November, 1998.

[Monchy, 2000] Monchy F., Maintenance - méthodes and organization. Dunod, Paris, 2000.

[Motta et al., 1990] Motta E., Rajan T. et Eisenstadt M., Knowledge Acquisition as a Process of Model Refinement. *Knowledge Acquisition*, 2, pp. 21-49, 1990.

[Muller, 2005] Muller A., Contribution à la maintenance prévisionnelle des systèmes de production par la formalisation d'un processus de pronostic. *Thèse de doctorat*, Université Henri Poincaré, Nancy, juin, 2005.

[Nonaka et Takeuchi, 1995] Nonaka I. et Takeuchi H., The Knowledge-Creation Company : How Japanese Companies Create the Dynamics of Innovation. New York/Oxford, Oxford University Press, 1995.

[Perron, 2000] Perron L., La réutilisation de cas : une problématique commune à l'intelligence artificielle et à l'ergonomie cognitive, mais des points de vue différents. *Ingénierie de connaissances*, pp.501-513, 2000.

[Piechowiak, 2003] Piechowiak S., Intelligence artificielle et diagnostic. *Techniques de l'ingénieur*, S 7 217, décembre, 2003.

[Pieri et al., 2001] Pieri G., Klein M.R. et Milanese M., MAIC: a data and knowledge-based system for supporting the maintenance of chemical plant. *Production Economics*, pp.1-17, 2001.

[Pomian, 1996] Pomian J., Mémoire d'entreprise, techniques et outils de la gestion du savoir. Ed. Sapientia, 1996.

[Prax, 1997] Prax J.Y., Manager les connaissances dans l'entreprise. INSEP éditions, Paris, 1997.

[Probst, 2002] Probst G., Managing Knowledge, Building Bloskc for Success. ISBN 0471-99768-4, Wiley, Sussex, England, 2002.

[Proteus-WP2 Team, 2004] WP2 Team: Déchamp L., Dutech A., Montroig T., Qian X., Racoceanu D., Rasovska I., Brézillon P., Charpillet F., Jaffray N., Moine N., Morello B., Müller S., Nguengang G., Palluat N., Pellisier L., On the Use of Artificial Intelligence for

Prognosis and Diagnosis in the PROTEUS e-maintenance platform, IEEE Mechatronics & Robotics, E-maintenance Special Session, Aachen, Allemagne, 2004.

[Rasovska et al., 2004a] Rasovska I., Chebel-Morello, B. et Zerhouni N., A conceptual model of maintenance process in unified modelling language. Proc. of the 11th Symposium on Information Control Problems in Manufacturing, INCOM'2004, Salvador-Bahia, Brésil, 2004a.

[Rasovska et al., 2004b] Rasovska I., Chebel-Morello B. et Zerhouni N., Modélisation des connaissances de maintenance : aide au diagnostic et à la réparation. Proc. du 5ième Conférence Francophone de MOdélisation et SIMulation MOSIM, Nantes, France, 2004b.

[Rasovska et al., 2005a] Rasovska I., Chebel-Morello B. et Zerhouni N., Un système de raisonnement à partir de cas dans une plate-forme de e-maintenance. Després S. (Ed.), actes du 13ème atelier de raisonnement à partir de cas, Nice, France, juin 2005.

[Rasovska et al., 2005b] Rasovska I., Chebel-Morello B. et Zerhouni N., Développement d'une base de cas pour un système de raisonnement à partir de cas en maintenance industrielle. Proc. of PENTOM 2005, 2ème édition du colloque international francophone : Performance et Nouvelles Technologies en Maintenance, Marrakech, Maroc, avril, 2005.

[Rasovska et al., 2005c] Rasovska I., Chebel-Morello B. et Zerhouni N., Process of s-maintenance: decision support system for maintenance intervention. Proc. of 10th IEEE International Conference on Emerging Technologies and Factory Automation ETFA'05, Italie, 2005.

[Rasovska, 2002] Rasovska I., Modélisation technico-économique d'un projet de télémaintenance, DEA IAP, Université de Franche Comté, 2002.

[Rastogi, 2000] Rastogi P.N., Knowledge Management and Intelectual Capital – The new virtuous reality of competitiveness. Human Systems Management, 2000.

[Retour et al., 1993] Retour D., Bouche M. et Plauchu V., Où va la maintenance industrielle. *Problèmes économiques*, No. 2.159, pp. 7-13, Janvier, 1990.

[Reynaud & Tort, 2000] Reynaud Ch. et Tort F., Diriger la réutilisation de composants à l'aide d'ontologies. *Ingénierie de connaissances*, pp. 59-74. 2000.

[Richter & Wess, 1991] Richter M.M. et Wess S., Similarity, Uncertainty and Case-Based Reasoning in PATDEX. Automated reasoning, essays in honour of Woody Bledsoe, Kluwer, 1991, pp. 249-265.

[Richter et al., 1993] Richter M.M., Wess S., Althoff K.D. et Maurer F., First European Workshop on Case-Based Reasoning, University of Kaiserslautern, Germany, *Lecture Notes in Artificial Intelligence*, vol 837, Springer Verlag, Berlin, 1993.

[Richter, 1995] Richter M.M., The Knowledge Contained in Similarity Measures. Invited talk, First International Conference on Case-Based Reasoning (ICCBR'95), Sesimbra, Portugal, 1995.

[Richter, 1998] Richter M.M., Introduction (to Case-Based Reasoning). Lenz et al. (Eds.), *Case-based reasoning technology: from foundations to applications*, Springer-Verlag, 1998.

[Richter, 2006] Disponible sur http://wwwagr.informatik.uni-kl.de/~richter/Vorlesungen/KM/.

[Rosario, 1996] Rosario J.G., Much ado about knowledge capital. *Business World*, Philippines, 1996.

[Rousseau, 1988] Rousseau B., Vers un environnement de résolution de problèmes en biométrie – apports des techniques de l'intelligence artificielle et de l'interaction graphique. *Thèse de doctorat*, Université Claude Bernard, Lyon, 1988.

[Ruet, 2002] Ruet M,. Capitalisation et réutilisation d'expériences dans un contexte multiacteur. *Thèse de doctorat*, Laboratoire Génie de Production de l'Ecole Nationale d'Ingénieurs de Tarbes, 2002.

[Rumbaugh et al., 1999] Rumbaugh J., Jacobson L. et Booch G., The unified modeling language reference manual. Reading, Addison Wesley, 1999.

[Schreiber et al., 1994] Schreiber A., Terpstra A., Magn P. et Van Veelzen M., Analysing and Implementing VT Using CommonKads. Proc. of the 8th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, 1994.

[Schreiber et al., 1999] Schreiber G., Knowledge Engineering and Management: The CommonKADS Methodology. MIT Press, 1999.

[Simon, 1996] Simon G., Knowledge Acquisition and Modeling for Corporate Memory: Lessons learnt from Experience. Proc. Of KAW'96, Banff, Canada, pp. 11-18, 1996. Also in http://ksi.cpsc.ucalgary.ca/KAW/KAW96/KAW96ProcS.html

[Simpson, 1985] Simpson R.L., A Computer Model of Case-Based Reasoning in Problem Solving : An Investigation in the Domain of Dispute Mediation. *Thèse de doctorat*, School of Information and Computer Science, Georgia Institute of Technology, Atlanta,1985.

[Smyth & Keane, 1995] Smyth B. et Keane M.T., Remembering to forget: A competence preserving case deletion policy for case-based reasoning system. Proc. of the 14th International Joint Conference on Artificial Intelligence, pp. 377-382, 1995.

[Smyth & Keane, 1996] Smyth B. et Keane M.T., Using Adaptation Knowledge to Retrieve and Adapt Cases. *Knowledge-Based Systems*, 9(2), pp. 127-135, 1996.

[Smyth & McKenna, 1999] Smyth B. et McKenna E., Building compact competent case-bases. Proc. of the 3rd International Conference on Case-Based Reasoning, pp. 329-342, 1999.

[Sowa, 1984] Sowa J.F., Conceptuel Structures: Information Processing in Mind and Machine. Reading (Mass.), Addison-Wesley Publishing Company, 1984.

[Spadoni, 2004] Spadoni M., Système d'information centré sur le modèle CIMOSA dans un contexte d'entreprise étendue, JESA, Volume 38, n° 5, pp. 497-525, 2004.

[Steels, 1993] Steels L., Corporate knowledge management. Proc. of ISMICK'93, Compiègne, France, pp. 9-30, 1993.

[Sveiby, 1997] Sveiby K., The New Organization Wealth. San Francisco: Berrett-Koehler, 1997.

[Sycara, 1987] Sycara E.P., Resolving adversial conflicts : An approach to integrating case-based and analytic methods. *Thèse de doc*torat, School of Information and Computer Science, Georgia Institute of Technology, Atlanta, 1987.

[Szulanski, 1996] Szulanski G., Exploring Internal Stickiness: Impediments to the Transfer of Best Practice Within the Firm. *Strategic Management Journal*, 17 (Winter Special Issue), pp. 27-43, 1996.

[Talbi & Laublet, 2000] Talbi N. et Laublet P., Construction d'un modèle générique d'expertise à partir d'un exemple : une solution pragmatique. *Ingénierie de connaissances*, p.43-58, 2000.

[Tannembaum & Alliger, 2000] Tannenbaum S.I. et Alliger G.M., Knowledge Management : Clarifying the Key Issues. ISBN 0967923913, IHRIM, 2000.

[Taylor, 1997] Taylor R., Unisys Decision Support Systems Programme. City Gate London, février, 1996.

[Toscano & Lyonnet, 2003] Toscano R. et Lyonnet P., Synthèse de la fonction de classification d'un système de diagnostic industriel. JESA, 37(3), pp. 311-332, 2003.

[Tourtier, 1995] Tourtier P.A., Analyse préliminaire des métiers et de leurs interactions. Rapport intermédiaire, projet GENIE, INRIA, 1995.

[Van Craeynest et al., 2000] Van Craeynest J.M., Ermine J.L. et Chagnot Ch., Capitalisation des connaissances dans le cadre d'un transfert industriel. *Ingénierie de connaissances*, pp.465-480, 2000.

[Van der Spek & Spijkervet, 1997] Van der Spek R. et Spijkervet A., Knowledge Mangement: Dealing Intelligently with Knowledge. *Knowledge Management and Its Integrative Elements*, Liebowitz J. & Wilcox L.C. (Eds.), CRC Press, New York, 1993.

[Van Heijst et al., 1996] Van Heijst G., Van der Spek R. et Kruizinga. E., Organizing Corporate Memories. Proc. of KAW'96, Banff, Canada, pp. 42.1-17, 1996. http://ksi.cpsc.ucalgary.ca/KAW/KAW96/KAW96ProcS.html

[Veloso et al., 1995] Veloso M. et Aamodt A., The 1st International Conference on Case-Based Reasoning, ICCBR'95 – Sesimbra, Portugal, Springer-Verlag, Berlin, Germany, 1995.

[Veloso, 1994] Veloso M.M., Planning and Learning by Analogical Reasoning. *Lecture Notes in Computer Science,* 886, Springer, Berlin, 1994.

[Vernadat, 1996] Vernadat F., Enterprise Modeling and Integration : Principles and Applications. Chapman & Hall, London, 1996.

[Vogel, 1988] Vogel C., Génie cognitif. Collection Sciences Cognitives, Editions Masson, Paris, France, Juin 1988.

[Watson, 2001] Watson I., Case-Based Reasoning and Knowledge Management: a Perfect Match? Proc. of the 14th Int. FLAIRS Conference, Key West Florida, pp.118-123, AAAI Press, 2001.

[Watson & Marir, 1994] Watson I. et Marir F., Case-Based Reasoning : A Review. *The Knowledge Engineering Review*, 1994.

[Wielinga et al., 1992] Wielinga B., Schreiber A.T. et Breuker J., KADS: a modeling approach to knowledge engineering. *Knowledge acquisition*, 4(1), pp. 5-53, March 1992.

[Wielinga et al., 1993] Wielinga B., Van De Velde W., Schreiber G. and Akkermans H., 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, 1993.

[Wiig, 1993] Wiig K.M., Knowledge Management Foundations. Arlington: Schema Press, 1993.

[Wikipedia, 2005] Disponible sur: http://fr.wikipedia.org.

[Wilke & Bergmann, 1996] Wilke W. et Bergmann R., Adaptation with the INRECA System. Proc. of the 12th European Conference on Artificial Intelligence, ECAI 96, Workshop on Adaptation in CBR, Budapest, Hungary, 1996.

[Wilke & Bergmann, 1998] Wilke W. et Bergmann R., Techniques and Knowledge Used for Adaptation During Case-Based Problem Solving. Proc. of IEA-98-AIE, *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1998.

[Yang & Wu, 2000] Yang Q. et Wu J., Keep it simple : A case-based maintenance policy based on clustering and information theory. Proc. of Canadian AI Conference, Montreal, Canada, 2000.

[Yang & Zhu, 2001] Yang Q. et Zhu J. A case addition policy for case-based maintenance. *Computational Intelligence Journal*, A special Issue on Case-Base Maintenance, Blackwell Publishers, Boston MA UK, 17(2), pp. 250-262, 2001.

[Zwingelstein, 1995] Zwingelstein G., Diagnostic des défaillances: Théorie et pratique pour les systèmes industriels, Hermès, 1995.

Contribution à une méthodologie de capitalisation des connaissances basée sur le raisonnement à partir de cas : Application au diagnostic dans une plateforme d'e-maintenance

Résumé :
Face aux évolutions technologiques, à la complexité croissante des équipements industriels et à la dynamique des processus ainsi qu'aux changements organisationnels et à la mobilité du personnel, les responsables de maintenance cherchent à formaliser et pérenniser le savoir et le savoir-faire des employés. Pour répondre à cette problématique, notre objectif est de fournir un service d'aide à la maintenance qui utilise et capitalise les connaissances. Nos travaux se situent dans le cadre du projet Européen Proteus qui a permis de développer une plateforme distribuée d'e-maintenance intégrant les différents systèmes et applications existants en maintenance. Nous avons déterminé quatre niveaux d'applications associés chacun à un ensemble d'outils d'aide : le niveau d'analyse d'équipement, celui de diagnostic et d'expertise, celui de gestion des ressources et celui de gestion des stratégies de maintenance. Ces outils d'aide nécessitent une expertise capitalisée que nous proposons de préserver dans une mémoire d'entreprise. Afin d'élaborer la mémoire ainsi que notre outil d'aide au diagnostic et à la réparation, nous avons introduit une démarche de capitalisation des connaissances articulée autour d'une méthodologie de raisonnement à partir de cas (RàPC) guidé par les connaissances. La conception de l'outil passe par la modélisation des connaissances qui se décline en un modèle de représentation (une ontologie du domaine) et en un modèle de résolution de problèmes (RàPC). Les modèles proposés utilisent des technologies émergeantes du Web sémantique permettant de faire évoluer le concept d'e-maintenance vers un nouveau concept de s-maintenance (maintenance sémantique).

Mots-clés : e-maintenance, s-maintenance, diagnostic, capitalisation des connaissances, raisonnement à partir de cas, ontologie, mémoire d'entreprise.

Abstract:
Faced with the technological developments, the increasing complexity of the industrial plants and the processes dynamics as well as with organisational changes and staff mobility, maintenance managers want to formalise and capitalize the know and know-how of maintenance operators and experts. To deal with these factors, our objective is to provide a service of maintenance assistance that uses and capitalizes knowledge. Our work has been part of European project Proteus which goal was to develop a generic distributed platform of e-maintenance to integrate and to provide a set of different maintenance systems and applications. We specified four levels of maintenance applications associated each one with a set of decision help systems: equipment analysis, diagnosis and expertise, resource management and maintenance strategy management. These tools require an expertise which we propose to capitalize and preserve in a corporate enterprise memory. In order to create this memory and to develop our diagnostic and repair help system, we introduced a methodology based on the association of knowledge capitalization and knowledge intensive case based reasoning. The development of our system is based on knowledge modeling that consists of a representation model (domain ontology) and a problem solving model (case based reasoning). The suggested models use emerging technologies from the semantic Web which make possible the evolution of e-maintenance concept in a new concept of s-maintenance (semantic maintenance).

Key words: e-maintenance, s-maintenance, diagnosis, knowledge capitalization, case-based reasoning, ontology, corporate memory.