N° Ordre : 14/06

Université de Valenciennes et du Hainaut-Cambrésis

Thermique Ecoulement Mécanique Matériaux Mise en Forme PrOduction TEMPo

Thèse de doctorat

Pour obtenir le grade de Docteur de l'Université de

VALENCIENNES ET DU HAINAUT-CAMBRESIS

Spécialité Automatique et Génie Informatique

Présentée et soutenue par Antoine LE MORTELLEC

Le 30/01/2014 à Valenciennes

Ecole doctorale :

Sciences Pour l'Ingénieur (SPI)

Equipe de recherche, Laboratoire :

Laboratoire : Thermique Ecoulement Mécanique Matériaux Mise en Forme Production (TEMPO)
Equipe : Production, Services, Information (PSI)

Proposition d'une architecture de surveillance "active" à base d'agents intelligents pour l'aide à la maintenance de systèmes mobiles

Application au domaine ferroviaire

JURY

Président du jury
Pierre Castagna. Professeur à l'Université de Nantes.

Rapporteurs
Zineb Simeu-Abazi. Maître de conférences HDR à l'Université Joseph Fourier de Grenoble.
Benoît Iung. Professeur à l'Université de Lorraine.
Samir Lamouri. Professeur à l'Ecole Nationale Supérieure d'Arts et Métiers Paris Tech.

Examinateur
Audine Subias. Maître de conférences HDR à l'Université de Toulouse.

Directeur de thèse
Damien Trentesaux. Professeur à l'Université de Valenciennes et du Hainaut Cambrésis.

Co-encadrants de thèse
Yves Sallez. Maître de conférences HDR à l'Université de Valenciennes et du Hainaut Cambrésis.
Thierry Berger. Maître de conférences à l'Université de Valenciennes et du Hainaut Cambrésis.

Membre invité
Frédéric Grzesiak. Directeur technique de la société PROSYST de Valenciennes.

Avant-propos

Cette thèse s'inscrit dans le cadre du projet SURFER (SURveillance active FERroviaire) conduit par Bombardier-Transport, en collaboration avec la société PROSYST, le groupe HIOLLE INDUSTRIES, l'Université de Valenciennes et du Hainaut-Cambrésis (UVHC) et l'Institut Français des Sciences et Technologies des Transports, de l'Aménagement et des Réseaux (IFSTTAR). Elle a été financée dans le cadre d'un contrat de collaboration de Recherche entre le laboratoire TEMPO EA4542 et la société PROSYST. Le temps de travail a été partagé entre ces deux entités.

Le projet SURFER est supporté financièrement par le Fonds Unique Interministériel (FUI) et la région Nord Pas-de-Calais, et sponsorisé par les Pôles de compétitivité *i-Trans* et *Advancity*.

Les accords de consortium établis entre les différents partenaires du projet SURFER ont restreint la divulgation de certains développements scientifiques et technologiques réalisés pendant mes travaux de Recherche. Ces éléments ne sont donc pas publiés dans ce manuscrit.



Remerciements

Les travaux présentés dans ce manuscrit ont été réalisés au sein de l'équipe PSI du laboratoire TEMPO de l'Université de Valenciennes et du Hainaut Cambrésis en collaboration avec la société PROSYST.

Je tiens à remercier en premier lieu le professeur Damien Trentesaux, responsable de l'équipe PSI, qui a dirigé cette thèse pour la qualité de son encadrement, ses conseils pertinents et son aide tout au long de ces trois années de doctorat. Je remercie mes co-encadrants de thèse, M. Yves Sallez, Maître de Conférences HDR, et M. Thierry Berger, Maître de Conférences, pour leur disponibilité, leurs critiques constructives et leurs encouragements, ainsi que pour le temps qu'ils ont consacré à relire et à corriger ce manuscrit.

J'exprime toute ma gratitude à Mme Zineb Simeu-Abazi, Maître de Conférences HDR au laboratoire G-SCOP de Grenoble, à M. Benoît Iung, professeur au CRAN de Nancy et à M. Samir Lamouri, professeur à l'École Nationale Supérieure d'Arts et Métiers Paris Tech, qui m'ont fait l'honneur de rapporter cette thèse. J'ai été particulièrement sensible à leur lecture attentive de ce manuscrit, aux nombreuses discussions et perspectives scientifiques qu'ils ont apportées à mes travaux. Je remercie vivement Mme Audine Subias, Maître de Conférences HDR au LAAS de Toulouse pour avoir accepté d'examiner cette thèse et M. Pierre Castagna, professeur à l'IRCCyN de Nantes, pour avoir accepté de présider ce jury.

Mes remerciements s'adressent également à l'ensemble du personnel de la société PROSYST et à son dirigeant, M. Abdallah Asse, pour leur accueil et leur sympathie. Je remercie chaleureusement M. Frédéric Grzesiak, directeur technique et M. Khaled El Sanwar, ingénieur de Recherche et Développement, pour leur aide, la vision qu'ils m'ont transmise sur les différents aspects du sujet de thèse et la confiance qu'ils m'ont accordée.

Je tiens à remercier M. Guillaume Branger, ingénieur de Recherche et Développement au sein de Bombardier-Transport France et pilote du projet SURFER, pour son soutien, l'expérience qu'il m'a apportée et l'intérêt qu'il a porté à l'égard de mes travaux.

Je remercie évidemment les membres de l'équipe TEMPO-PSI ainsi que tous les "surfeurs" qui ont contribué à la concrétisation de cette thèse. Je pense en particulier à Joffrey Clarhaut, Rebiha Bekrar, Jean Gandibleux, Quentin Coutadeur et Vivien Basselot.

Je remercie ma famille pour m'avoir soutenu pendant toutes ces années d'études. Merci à mes sœurs, Cécile et Aurore, sans qui cette thèse n'existerait probablement pas.

Je n'oublie pas mes amis, notamment mes amis Rennais, qui m'ont toujours encouragé et motivé pendant ces années passées dans le "ch'Nord". Merci à vous.

À ma maman...,

Table des matières

Introduction générale ***9***

Ch. I Problématique de la surveillance pour l'aide à la maintenance ***12***

Introduction 12
1. Contexte général 13
1.1 Approche systémique adoptée 13
1.2 Propriétés des systèmes cibles considérés 14
1.3 Terminologie relative à la maintenance 16
1.4 Activités de la maintenance 17
2. Fonctions d'un système de surveillance 20
2.1 Détection 22
2.2 Diagnostic 22
2.3 Pronostic 23
2.4 Elaboration de conseils 23
3. Modèle d'architecture de surveillance 24
3.1 Standards d'architectures de surveillance 24
3.2 Problématiques de la surveillance des systèmes cibles considérés 30
4. Conclusion 34

Ch. II Surveillance de systèmes mobiles : état de l'art et approches de surveillance active ***35***

Introduction 35
1. Etat de l'art des architectures de diagnostic 35
1.1 Diagnostic distant 36
1.2 Diagnostic embarqué 38
1.3 Typologie des architectures de diagnostic 43
2. Approches de surveillance active 45
2.1 Concept de surveillance active 46
2.2 Approche multi-agents 47
2.3 Approche holonique 49
2.4 Comparaison des approches 52
3. Conclusion 53

Ch. III Proposition d'une architecture holonique pour la surveillance active de systèmes cibles mobiles ***54***

Introduction 54
1. Système de surveillance holonique 54
1.1 Approche de modélisation du système cible à surveiller 55
1.2 Spécification d'une architecture holonique de surveillance 58
1.3 Architecture de surveillance proposée 61
2. Structure interne d'un holon de surveillance 66
2.1 Données exploitées par la fonction de surveillance d'un holon 66
2.2 Hypothèses pour l'élaboration d'un diagnostic 67
2.3 Modules internes d'une fonction de surveillance 67
2.4 Niveaux de diagnostic d'une fonction de surveillance 69
3. Conclusion 71

Ch. IV Mise en œuvre de l'architecture holonique de surveillance active ***72***

Introduction 72
1. Diagnostic PSC individuel non-contextualisé 73
1.1 Nature des systèmes diagnostiqués 73
1.2 Méthode de diagnostic mise en œuvre 73
1.3 Limitations de la méthode de diagnostic mise en œuvre 74
2. Contextualisation du diagnostic 75
2.1 Contextualisation individuelle 76
2.2 Contextualisation collective 78
3. Description comportementale des holons de surveillance 80
3.1 Comportements des holons élémentaires 82

3.2 Comportements des holons composés 84
4. Mise en œuvre des holons de surveillance 87
4.1 Implémentation de la partie informationnelle d'un holon 87
4.2 Implantation physique des holons de surveillance 88
5. Conclusion 89

Ch. V Application au domaine ferroviaire 90

Introduction 90
1. Contexte industriel du projet SURFER 90
1.1 Objectifs du projet SURFER 90
1.2 Utilisateurs du système SURFER 91
2. Application de notre proposition pour un système train 92
2.1 Instanciation de l'architecture holonique de surveillance 93
2.2 Algorithmes des holons de surveillance 95
2.3 Exemples de scénarios de diagnostic 99
3. Plateforme expérimentale 102
3.1 Architecture de la plateforme PEMAS 103
3.2 Implémentation des agents dans Jade 105
4. Résultats obtenus 108
4.1 Implémentation de la couche acquisition embarquée 108
4.2 Mise en œuvre de la méthode de diagnostic à base de modèle 109
4.3 Contextualisation du diagnostic 110
5. Conclusion 110

Références 117

Liste des figures 129

Annexes 131

Annexe A : Complément sur l'approche systémique 132

Annexe B : Etude comparative des méthodes de diagnostic 133

B.1 Méthodes à base de modèles (MBD) 134
B.2 Méthodes basées sur des données historiques 139
B.3 Comparaison des méthodes de diagnostic 141

Annexe C : Méthode de diagnostic mise en œuvre 144

Annexe D : Prototype SURFER 146

Introduction générale

Pour se démarquer et rester compétitif, les industriels misent notamment sur le développement et l'intégration rapide de nouvelles fonctionnalités offertes par les évolutions technologiques de l'information et de la communication.

Ces deux dernières décennies, les systèmes embarqués ont été introduits dans de nombreux domaines d'application (transport, industrie, habitat, médical...). Ces systèmes se sont vu confier des tâches plus importantes pour délivrer de nouveaux services aux utilisateurs avec des délais de mise sur le marché toujours plus courts et à moindre coût. Dans certains domaines, ces systèmes ont également permis de répondre aux exigences accrues en matière de sécurité et de fiabilité.

Paradoxalement, le développement de ces équipements embarqués a contribué à accroitre le nombre de pannes des systèmes dans lesquels ils sont enfouis. Ces pannes, aux conséquences généralement coûteuses, voir catastrophiques, sont devenues inacceptables pour les industriels qui commercialisent ces produits ainsi que pour leurs clients. L'identification des causes de certaines pannes représente actuellement un véritable challenge dans les activités de la maintenance. Elles entrainent une indisponibilité excessive des équipements.

Dans le domaine des transports, ces constats ont conduit un acteur majeur de l'industrie ferroviaire à améliorer la disponibilité de ses véhicules sur rail à travers une approche innovante visant à optimiser la maintenance. Ainsi, le projet SURFER, qui est à l'origine des travaux présentés dans ce manuscrit, a été proposé. Ce projet repose sur la conception et le déploiement d'une architecture évoluée pour la surveillance de matériels roulants ferroviaires.

Cette thèse se situe à la rencontre de plusieurs communautés de Recherche, dans laquelle la notion d'intelligence (ou activité) rattachée à un produit a pour objet l'évaluation et la gestion de la santé des équipements. Elle regroupe notamment des concepts relatifs à la surveillance dans un contexte de maintenance, au diagnostic et au développement de produits "actifs" tout au long de son cycle de vie.

Notre contribution principale concerne la proposition d'une architecture générique de surveillance "active" pour l'aide à la maintenance de systèmes mobiles, basée sur des entités de surveillance "intelligentes". Cette architecture a vocation à supporter des méthodologies de diagnostic développées par différentes communautés de Recherche. L'élaboration d'une nouvelle méthode de diagnostic n'est donc pas l'objet de notre proposition.

Le projet SURFER étant le cadre applicatif de nos travaux, le domaine des transports est une référence pour illustrer notre approche, sans pour autant restreindre l'application de notre contribution à ce domaine.

Nous avons choisi de structurer ce manuscrit en cinq chapitres, dont l'organisation est illustrée par la figure 1.

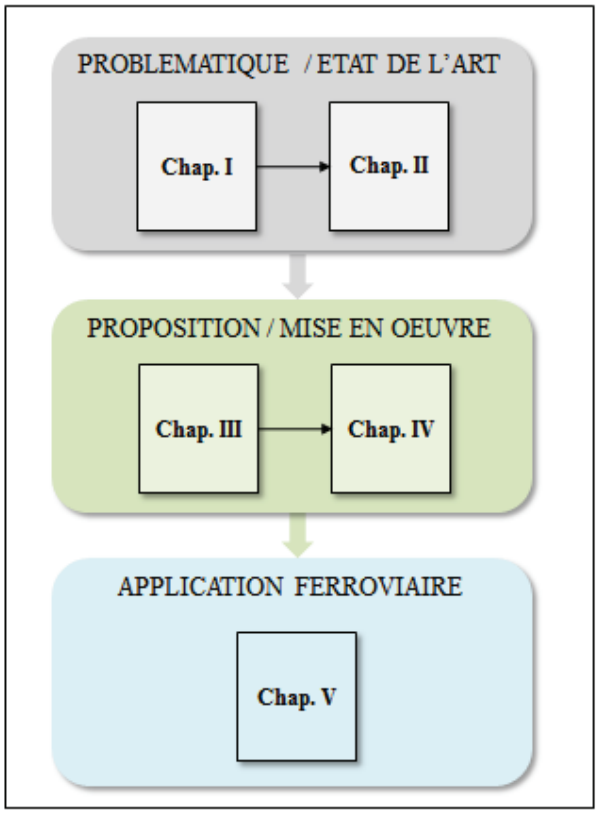

Figure 1. Organisation des chapitres de ce manuscrit.

Le **chapitre I** introduit le contexte général de nos travaux en présentant tout d'abord les caractéristiques des systèmes à maintenir. Ce chapitre positionne ensuite notre étude par rapport à un modèle d'architecture de surveillance de la littérature. Les différents verrous liés à la surveillance des systèmes considérés sont alors exposés et les exigences à satisfaire dans le cadre de nos travaux sont définies.

Le **chapitre II** dresse un état de l'art des approches de la littérature pour la surveillance de systèmes mobiles, en proposant une typologie des architectures de diagnostic. Une classe d'architecture est retenue dans le contexte de notre étude. Suivant ce type d'architecture, ce chapitre s'oriente par la suite vers une approche de surveillance "active" en adoptant une approche de modélisation à base d'entités autonomes et communicantes.

Le **chapitre III** est consacré à notre proposition. Une architecture générique pour la surveillance de systèmes mobiles est proposée. Cette architecture repose sur les choix réalisés dans le précédent chapitre. Ce chapitre spécifie l'architecture proposée et détaille la structure interne des entités de surveillance qui composent cette architecture de surveillance.

Le **chapitre IV** met en œuvre l'architecture de surveillance proposée. Dans ce chapitre, les processus internes aux entités de surveillance et les comportements attendus de ces entités sont définis. Par la suite, une approche est adoptée pour implémenter ces entités de surveillance.

Le **chapitre V** applique les propositions des deux chapitres précédents au domaine ferroviaire dans le cadre du projet SURFER. Dans un premier temps, ce chapitre instancie l'architecture proposée pour la surveillance d'un train de passagers. Par la suite, la plateforme expérimentale conçue pour implémenter cette architecture et valider notre approche est détaillée. Enfin, les résultats obtenus dans ce cadre applicatif sont présentés.

Suite à ce dernier chapitre, la conclusion de nos travaux et les perspectives de Recherche envisagées sont exposées.

Chapitre I

Problématique de la surveillance pour l'aide à la maintenance

Introduction

Dans un contexte économique hautement concurrentiel, les fabricants et intégrateurs de produits à haute valeur technologique sont contraints de concevoir des systèmes à des coûts de plus en plus faibles sur l'ensemble des phases du cycle de vie. L'objectif étant de proposer des produits compétitifs en terme de coûts d'acquisition, d'exploitation et de maintenance (coût global de possession ou LCC : *Life Cycle Cost*). Par conséquent, limiter les coûts de maintenance et d'exploitation tout en améliorant la disponibilité des systèmes est une préoccupation majeure des industriels.

La maintenance est l'un des éléments clef dans la réduction de l'indisponibilité. Elle représente également un point sensible, soumise notamment à des contraintes financières, sécuritaires et énergétiques (Rabatel et al., 2009). Ainsi, la maintenance est un levier stratégique de compétitivité et d'innovation pour les industriels (Muller et al., 2008). Elle permet de préserver les caractéristiques fonctionnelles des systèmes et vise à améliorer leurs disponibilités dans des conditions économiques acceptables.

L'optimisation de la maintenance se traduit par une réduction des temps d'immobilisation du système maintenu et repose donc sur la pertinence des actions réalisées sur le système lors des interventions de maintenance. Un système de maintenance efficace doit permettre d'orienter au mieux le personnel de maintenance, en délivrant des informations directement exploitables et précises sur les opérations de maintenance à réaliser.

Dans ce contexte, un système de surveillance observant en permanence un système à maintenir lors de son fonctionnement et assurant la tâche de diagnostic de manière autonome offre une aide précieuse aux services de maintenance (Jardine et al., 2006). Les informations fournies par un tel système contribuent notamment à anticiper les pannes et à accélérer la remise en service du système maintenu après l'occurrence de la défaillance. Ces informations de surveillance peuvent apporter des précisions sur l'origine des dysfonctionnements observés et sur les interventions de maintenance à réaliser.

Ce premier chapitre présente les systèmes à maintenir considérés dans nos travaux et introduit les principales notions relatives à la surveillance dans un contexte de maintenance. Par la suite, les différentes fonctions d'un système de surveillance seront présentées et les problématiques inhérentes à la surveillance des systèmes étudiés seront exposées.

1. Contexte général

Les avancées technologiques de l'information et de la communication ont favorisé l'intégration massive de fonctions automatisées au sein de produits industriels (Brahim-Djelloul et al., 2012 ; Kiencke et al., 2006). Ces fonctions permettent de délivrer de nouveaux services aux utilisateurs et d'atteindre des objectifs de plus en plus élevés en termes de sécurité et de fiabilité. Ainsi, de nombreuses fonctionnalités sont assurées par des systèmes embarqués. Un système embarqué est un système complexe intégrant une partie logicielle et une partie matérielle conçues conjointement pour fournir des fonctionnalités données (Simeu, 2005). Ces équipements embarqués peuvent réaliser des fonctions plus évoluées lorsqu'ils interagissent entre eux au sein d'un système plus vaste (système de transport, bâtiment industriel, systèmes de production, par exemple). Ces équipements interconnectés forment alors un système réparti (Dievart et al., 2010).

La complexité croissante des systèmes dotés de ces équipements embarqués tend à rendre leurs opérations de maintenance bien plus délicates. Par exemple, une intervention de maintenance sur un tel système nécessite une connaissance approfondie du fonctionnement de l'ensemble du système et des différents équipements embarqués qui le composent. Pour faire face à cette complexité, des approches de maintenance dites "intelligentes" ont été rendues possibles grâce aux évolutions de l'informatique, de l'électronique et des télécommunications (Campos, 2009). Ces approches de maintenance innovantes visent à évaluer l'état d'un système et à déterminer les interventions de maintenance nécessaires en fonction de cet état. Les objectifs attendus de ces nouvelles approches de maintenance sont d'accroitre la durée de fonctionnement des équipements, de réduire les coûts de maintenance, d'exploitation et de possession, et d'améliorer la sécurité (Vachtsevanos et Lewis, 2006).

Cette partie positionne tout d'abord les systèmes considérés dans nos travaux en adoptant une approche systémique. Par la suite, les propriétés des systèmes à maintenir considérés sont définies et la terminologie adoptée dans nos travaux est présentée. Enfin, cette partie décrit les différents types de maintenance et précise le rôle de la surveillance au sein d'un système de maintenance.

1.1 Approche systémique adoptée

Ce mémoire se focalise sur l'aide à la maintenance de systèmes constitués d'équipements embarqués répartis, désignés ci-après *sous-systèmes*, et organisés suivant une structure hiérarchisée de contrôle. Par la suite, un système constitué de ces sous-systèmes sera nommé *système cible*. Un système cible est supposé être maintenu par un *système de maintenance*.

Un système de maintenance regroupe différentes entités en charge de planifier et de réaliser les interventions de maintenance (centre de maintenance, personnel de maintenance, par exemple).

La figure 1.1 ci-dessous présente un exemple de système cible supposé être situé dans son environnement. Ce système cible se compose de différents sous-systèmes organisés suivant une structure hiérarchisée de contrôle : un sous-système commande un ou plusieurs sous-systèmes de niveau inférieur. De plus, les sous-systèmes interagissent entre eux dans le domaine physique et dans le domaine informationnel. Les relations maître-esclave illustrées dans cet exemple représentent les liens "logiques" de subordination entre sous-systèmes.

Les liens "physiques" entre ces sous-systèmes peuvent être réalisés par différentes technologies (bus ou réseaux, par exemple).

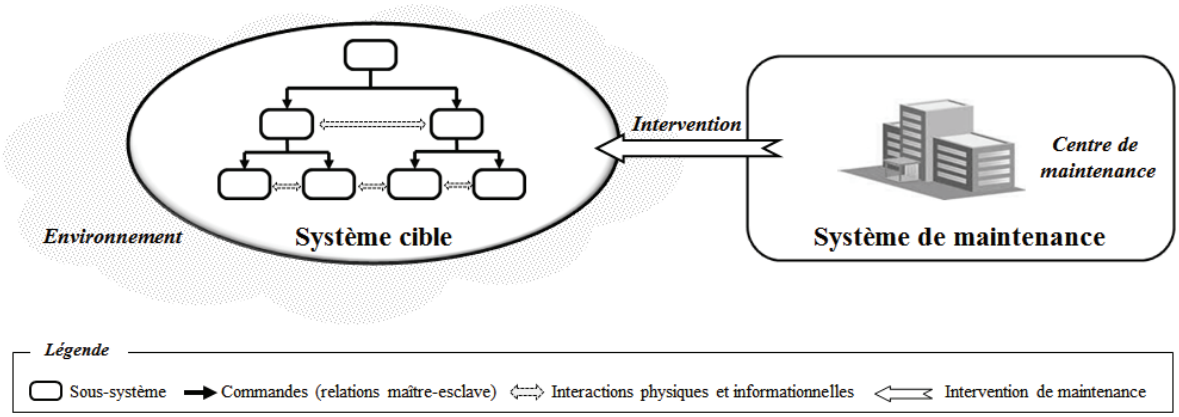

Figure 1.1. Exemples de système cible et de système de maintenance considérés.

Le système de maintenance est supposé disposer d'un centre de maintenance, dont l'objectif est de maintenir le système cible tout au long de son usage. Lorsqu'une approche de maintenance doit être appliquée à un système cible, des exigences sont alors à allouer au système de maintenance comme au système cible. La section suivante définit les propriétés des systèmes cibles considérées dans nos travaux.

1.2 Propriétés des systèmes cibles considérés

Les systèmes cibles considérés dans nos travaux sont caractérisés par les propriétés suivantes :

- Décomposition : le système cible est supposé être décomposable en un ensemble fini de sous-systèmes.
- Hétérogénéité technologique : chaque sous-système est supposé être constitué d'une partie contrôle et d'une partie sous-contrôle. Ces sous-systèmes sont supposés exploiter différentes technologies (électrique, électronique, mécanique, hydraulique ou pneumatique, par exemple).
- Variabilité : un sous-système est supposé varier d'un même système cible à un autre (évolution de la technologie, modification de composants, modernisation, par exemple).
- Interactions : les sous-systèmes sont supposés interagir entre eux dans le domaine physique (échanges de flux de matière ou d'énergie, par exemple) et/ou dans le domaine informationnel (relations de type maître-esclave ou échanges d'informations, par exemple).
- Environnement : le système cible est supposé être situé dans un environnement. Par conséquent, le système cible comme chaque sous-système est supposé opérer dans un contexte opérationnel spécifique. Ce contexte peut être de nature physique (température, vibrations, par exemple) et/ou informationnelle (mode de fonctionnement, état du système, par exemple).
- Maintenabilité : le système cible est supposé être réparable et être maintenu au sein du système de maintenance par un centre de maintenance.

Le tableau 1.1 ci-après présente les domaines d'applications les plus représentatifs des systèmes cibles considérés dans nos travaux. Les principales caractéristiques associées à ces systèmes cibles sont également mentionnées.

Tableau 1.1. Exemples de systèmes cibles considérés, inspiré de (Simeu, 2005).

Application	Mobilité	Sévérité de l'environnement	Intervention immédiate sur le système	Haut niveau de fiabilité
Transport				
Automobile	✓	✓	-	✓
Ferroviaire	✓	✓	-	✓
Aérien	✓	✓	-	✓
Maritime	✓	✓	-	✓
Aérospatial				
Fusée navette	✓	✓	-	✓
Satellites	✓	✓	-	✓
Industrie				
Systèmes de production	-	✓	✓	✓
Centrales nucléaires	-	✓	✓	✓
Habitat				
Bâtiments (*immotique*)	-	-	✓	-
Habitations (*domotique*)	-	-	✓	-

Légende

- : caractéristique absente ou négligeable ✓ : caractéristique présente

Dans certains domaines d'applications, la mobilité du système cible entraine des contraintes spécifiques en termes de maintenabilité. En effet, lorsque le système cible est mobile, celui-ci doit communiquer avec un centre de maintenance distant (Jianjun et al., 2007). Par exemple, un système de transport (automobile, ferroviaire, aérien ou maritime) est un système cible mobile évoluant dans un environnement changeant (climat, état de l'infrastructure, profil du conducteur, par exemple). Les interventions de maintenance ne peuvent être exécutées immédiatement sur ce système (système en service commercial et éloigné du centre de maintenance, par exemple) (Umiliacchi et al., 2011). Aussi, un système de transport doit répondre à des exigences strictes en matière de fiabilité.

La disponibilité d'un système cible repose sur les objectifs de fiabilité fixés lors de sa conception et sur l'efficacité des interventions de maintenance réalisées au cours de sa phase

d'utilisation. Les sections suivantes présentent la terminologie et les principaux concepts associés à la maintenance.

1.3 Terminologie relative à la maintenance

La maintenance est définie comme l'ensemble de toutes les actions techniques, administratives et de management 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 (NF-EN-13306, 2010). Un *bien* désigne un élément, composant, mécanisme, sous-système, unité fonctionnelle, équipement ou système qui peut être décrit et considéré individuellement. Différents événements peuvent compromettre le fonctionnement désiré d'un bien ou la mission pour laquelle un bien a été conçu. Les entraves au bon fonctionnement d'un bien sont les *fautes*, les *erreurs* et les *défaillances* (Isermann et Ballé, 1997). Une faute est une déviation non acceptable d'au moins une propriété caractéristique exprimée par le cahier des charges ou d'un paramètre d'un bien. Une erreur représente un changement d'état d'un bien pouvant entraîner une défaillance. Elle est produite par une ou plusieurs fautes et peut engendrer de nouvelles erreurs. Une défaillance est la cessation de l'aptitude d'un bien à accomplir une fonction requise et peut être causée par une ou plusieurs erreurs. Après la défaillance, le bien est en état de *panne*, c'est-à-dire qu'il est inapte à accomplir une fonction requise. Une panne peut être complète ou partielle et résulte toujours d'une défaillance.

Dans les travaux présentés dans cette thèse, le système cible considéré est supposé être affecté par des fautes physiques (fautes matérielles) et informationnelles (fautes logicielles) survenant dans les différentes phases de son cycle de vie (fautes de développement et fautes opérationnelles, par exemple). Ces fautes induisent une ou plusieurs erreurs lorsqu'elles sont actives (états non désirés du système). Dès lors que le service délivré aux utilisateurs est impacté par ces erreurs et ne correspond plus au service attendu, une défaillance survient. La figure 1.2 ci-dessous illustre la relation entre les concepts de faute, erreur et défaillance survenant au sein d'un système cible. La présence d'une faute dans un sous-système du système cible engendre une erreur interne lors de son activation, qui conduit à la défaillance de ce sous-système. Cette défaillance peut constituer une faute pour un autre sous-système du système cible en provoquant à son tour une erreur interne et une défaillance de ce sous-système. Cette seconde défaillance entraine par la suite la défaillance du système cible en impactant le service délivré aux utilisateurs.

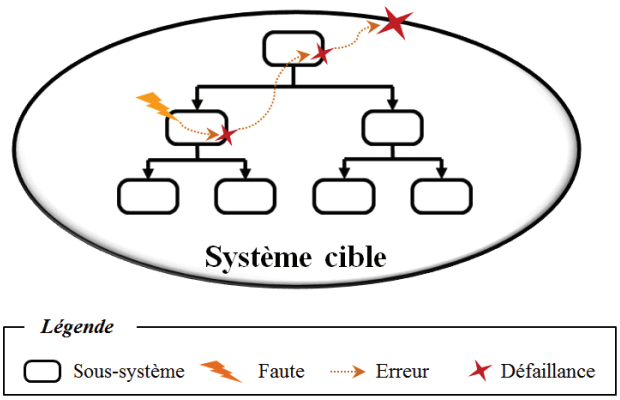

Figure 1.2. Relation causale entre faute, erreur et défaillance.

Il est à noter que les fautes et les défaillances peuvent se manifester de manière progressive dans le temps (phénomènes d'usures, par exemple), de manière soudaine (rupture d'une

liaison électrique, par exemple), ou présenter un caractère non permanent (défaillances temporellement bornées ou furtives, par exemple). Une défaillance produit des effets perceptibles par les utilisateurs du système et éventuellement décelables par un moyen de détection. Le terme *symptôme* sera utilisé par la suite pour désigner les effets observables d'une défaillance.

La tâche de diagnostic de défaillances consiste alors à analyser ces symptômes dans le but de rechercher les éléments défectueux du système cible responsables de la défaillance (fautes). Elle peut être réalisée par un opérateur de maintenance ou être exécutée par un système d'aide à la maintenance. Les différentes activités de la maintenance sont exposées ci-après.

1.4 Activités de la maintenance

La définition de la maintenance évoquée précédemment englobe différentes activités mises en œuvre tout au long du cycle de vie d'un produit. Ces activités concernent aussi bien des aspects opérationnels de la maintenance (expertise métier, diagnostic de défaillances, par exemple) que des aspects organisationnels (planification des interventions, approvisionnement des pièces de rechanges, par exemple). Dans ce mémoire, les aspects opérationnels de la maintenance seront principalement abordés. Cette section présente les différentes approches de maintenance et précise le rôle de la surveillance au sein des activités de la maintenance.

1.4.1 Typologie de la maintenance

L'étude des différentes approches de maintenance repose sur trois concepts : l'événement déclencheur d'une opération de maintenance, le type de maintenance correspondante et la nature des opérations de maintenance réalisées sur le système (Ribot, 2009). Selon la norme (NF-EN-13306, 2010), les approches de maintenance sont classées en deux catégories : la maintenance *préventive* et la maintenance *corrective*. La première vise à initier des opérations de maintenance sur un système avant que des défaillances ne se produisent, tandis que la seconde consiste à intervenir sur un système en état de panne. La figure 1.3 ci-dessous résume les principaux types de maintenance recensés dans la littérature selon ces deux catégories et précise les événements déclencheurs des interventions de maintenance.

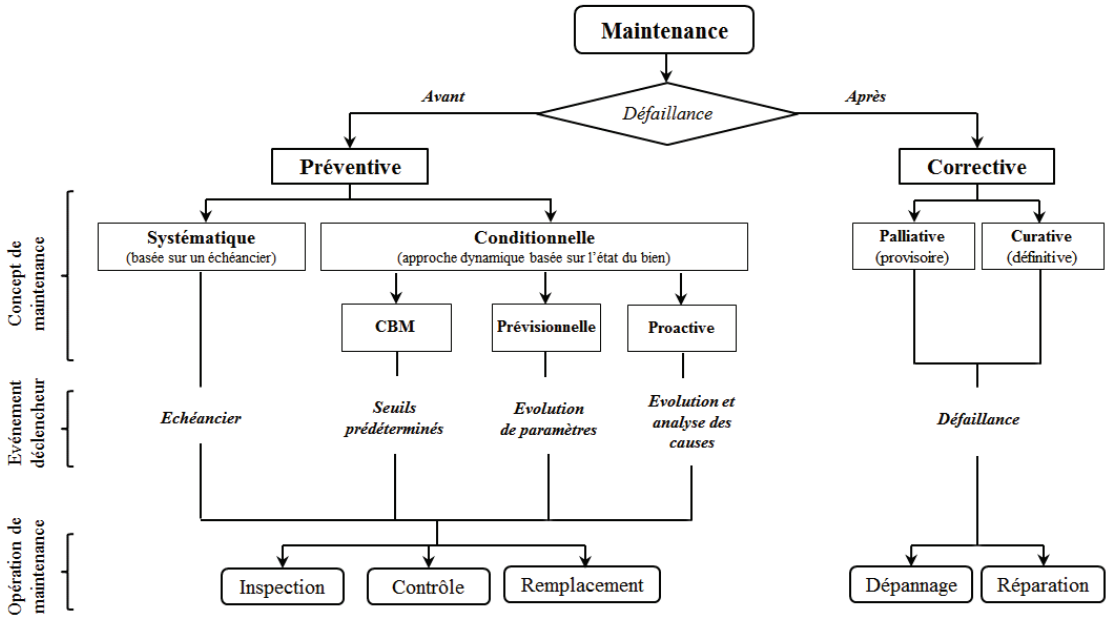

Figure 1.3. Différents types de maintenance, inspirée de (Alali Alhouaij, 2010).

La maintenance préventive est exécutée à intervalles prédéterminés ou selon des critères prescrits. Elle est destinée à réduire l'occurrence de défaillances et à éviter les pannes lors de la sollicitation du système. La maintenance préventive se décline en trois sous-types normalisés, détaillés ci-après : la maintenance préventive à caractère systématique, conditionnelle et prévisionnelle (NF-EN-13306, 2010).

- **Systématique** (*Predetermined Maintenance*) : maintenance préventive exécutée conformément à un échéancier, établi à partir d'un nombre d'unités d'usage (durée de fonctionnement, par exemple) ou suivant des données statistiques (recommandations constructeurs, par exemple). Elle vise principalement à rajeunir le matériel (remplacement périodique, par exemple) ou à limiter les dégradations (tâches d'entretiens courants), quel que soit l'état réel du bien maintenu (Zille, 2009). Seule la périodicité des interventions peut être ajustée pour optimiser ce type de maintenance.
- **Conditionnelle** (*Condition Based Maintenance*) : maintenance préventive subordonnée à l'analyse de l'évolution surveillée de paramètres significatifs de la dégradation ou de la baisse de performance d'un bien (Deloux, 2008). Les paramètres significatifs de dégradations concernent par exemple des mesures de caractéristiques physiques d'un système et précurseurs de pannes (bruit, vibration, vitesse anormale, épaisseur trop faible d'un matériau, par exemple). Ce type de maintenance nécessite d'établir des seuils à priori pour les différents paramètres mesurés (seuils prédéterminés). Le dépassement d'un seuil indique alors la nécessité d'une opération de maintenance (par exemple, le remplacement d'un pneumatique est nécessaire lorsque son épaisseur atteint la limite fixée par son témoin d'usure). Bien que la maintenance conditionnelle soit plus dynamique que la maintenance systématique puisqu'elle considère l'état d'un bien, le temps restant avant l'occurrence de la défaillance reste indéterminée avec ce type de maintenance.
- **Prévisionnelle** (*Predictive Maintenance*) : maintenance conditionnelle effectuée suivant les prévisions extrapolées de l'analyse et de l'évaluation de paramètres significatifs de la dégradation d'un bien (El koujok, 2010). Ce type de maintenance vise à prédire l'évolution de paramètres significatifs de dégradations d'un bien dans le futur et à estimer le temps de fonctionnement restant avant la défaillance (pronostic). Elle peut être qualifiée de maintenance *prédictive*, reposant sur les évolutions de paramètres surveillés dans le futur, et de maintenance *proactive,* qui consiste à déterminer les causes à l'origine des dégradations et des défaillances (Deloux, 2008).

La maintenance corrective a pour objectif de remettre un bien en état de panne dans un état fonctionnel. Elle peut prendre la forme d'une maintenance palliative ou curative (Ribot, 2009) :

- **Palliative :** maintenance corrective assurant une remise en état provisoire d'un bien (dépannage temporaire d'un pneumatique, par exemple). Ce type de maintenance consiste également à intervenir sur les effets néfastes d'une défaillance. Une action de maintenance palliative doit être suivie systématiquement d'une action de maintenance curative.

- **Curative :** maintenance corrective permettant une remise en état de manière durable et définitive d'un bien. Contrairement à la maintenance palliative, la maintenance curative a pour objectif de supprimer les causes des défaillances.

La maintenance préventive et la maintenance corrective sont complémentaires. En pratique, une politique de maintenance combinant ces deux approches est généralement appliquée aux systèmes. En effet, une maintenance préventive permet de diminuer la probabilité de défaillances, mais une part de maintenance corrective incompressible subsiste (défaillances imprévisibles). Ainsi, différentes politiques de maintenance *mixte* ont été définis (Ly et al., 2013). L'élaboration d'une politique de maintenance repose notamment sur un équilibre économique amenant à un meilleur compromis (coût et disponibilité, par exemple). La sous-section suivante positionne la surveillance suivant ces deux types de maintenance.

1.4.2 Rôle de la surveillance

Dans une approche de maintenance innovante, la surveillance occupe une place fondamentale (Raccoceanu, 2006). La surveillance est considérée dans nos travaux comme un dispositif passif, informationnel, qui n'influence pas le comportement du système cible sous surveillance. Ainsi, un système de surveillance est non-intrusif vis-à-vis d'un système cible. Ce dispositif délivre des informations pertinentes sur l'état d'un système cible pour déclencher des interventions de maintenance (maintenance préventive conditionnelle) et assister au mieux un opérateur de maintenance pour rétablir un système en panne (maintenance corrective). Par exemple, une défaillance ayant conduit à une panne d'un système cible peut être décryptée spontanément par un système de surveillance. Le système de surveillance transmet ainsi des informations sur l'état de ce système au centre de maintenance (figure 1.4). Ces informations concernent, par exemple, les éléments potentiellement responsables de la panne ainsi que des actions de maintenance à réaliser sur le système cible pour le rétablir dans un état fonctionnel. Le centre de maintenance génère alors une demande d'intervention aux opérateurs de maintenance chargés d'intervenir sur le système cible (tâche de maintenance corrective, par exemple). Les interventions de maintenance corrective sont alors mieux préparées et les durées des réparations sont maîtrisées.

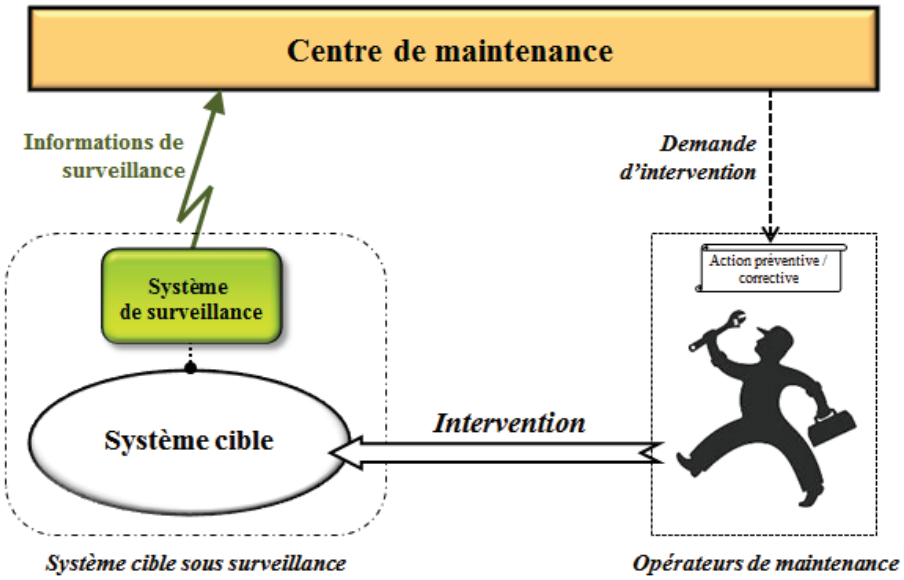

Figure 1.4. Système de surveillance pour l'aide à la maintenance.

Dans une approche de maintenance préventive conditionnelle (maintenance prédictive ou CBM : *Condition Based Maintenance*), l'objectif d'un système de surveillance est d'émettre des alertes avant la défaillance du système cible pour prévenir les pannes. Cette surveillance prédictive (ou dynamique) se focalise ainsi sur les tendances de l'évolution de l'état du système sous surveillance dans le temps. Les informations transmises aux personnels de maintenance permettent alors de signaler des dégradations ou des baisses de performance des éléments du système à un stade précoce de leur développement.

La nature des informations fournies par un système de surveillance dépend des profils des utilisateurs finaux et diffère donc selon leur expertise (opérateurs de maintenance, concepteurs du système, par exemple) (Alanen et al., 2006). Dans le cadre d'une surveillance pour l'aide à la maintenance au sens intervention, un système de surveillance doit incriminer des éléments à remplacer ou à réparer bien connus des opérateurs de maintenance. Par exemple, certains systèmes cibles comportent des Unités Remplaçables en Ligne (URL ou LRU : *Line Replaceable Unit*). Une URL peut être déposée et échangée rapidement pour optimiser les interventions de maintenance et réduire ainsi les temps d'indisponibilité des systèmes (Dievart et al., 2010). Par conséquent, une tâche de maintenance associée à une URL constitue une information pertinente pour un opérateur de maintenance. Néanmoins, suivant le niveau de complexité de la tâche de maintenance à réaliser, un opérateur de maintenance peut être habilité ou non à intervenir sur le système cible. Les informations de surveillance doivent donc être adaptées au niveau de qualification des utilisateurs du système de surveillance.

La surveillance est généralement réalisée *en-ligne,* pendant le fonctionnement du système cible sous surveillance, mais certaines fonctions de la surveillance peuvent être exécutées *hors-ligne*. C'est le cas notamment du diagnostic, dont l'exécution a posteriori peut être possible sur la base des symptômes détectés sur le système (Azarian et al., 2010). La partie suivante décrit les principales fonctions d'un système de surveillance.

2. Fonctions d'un système de surveillance

Un système de surveillance regroupe deux composantes essentielles : "voir" et "comprendre" (Basseville et Cordier, 1996). La première permet d'observer et de détecter des déviations du comportement du système sous surveillance par rapport à un comportement de référence (perception). La seconde détermine les composants potentiellement responsables des phénomènes observés et recherche les causes (interprétation). Dans cette partie, les principales fonctions d'un système de surveillance mises en œuvre pour des approches de maintenance de type conditionnelle (maintenance prédictive ou CBM) et corrective sont présentées.

Les fonctions assurées par un système de surveillance s'inscrivent dans différentes communautés de Recherche, et sont notamment au centre des thématiques SHM (*System Health Management*) et PHM (*Pronostics and Health Management)* (Medina-Oliva et al., 2012 ; Reed et al., 2011). Dans ce cadre, un système de surveillance comprend généralement l'acquisition de données, la détection d'anomalies de fonctionnement, le diagnostic, le pronostic et l'élaboration de recommandations de maintenance (Guillén et al., 2013 ; Vachtsevanos et Lewis, 2006). Ces différentes fonctions ont pour but de convertir un flux de données brutes, difficilement exploitable pour un opérateur de maintenance, en informations pertinentes, relatives à l'état de santé du système sous surveillance et aux opérations de maintenance à réaliser selon cet état (Abichou, 2013). Ainsi, un système de surveillance

permet de déterminer l'état d'un système (état normal ou anormal d'un système cible, par exemple) et lorsque ce système présente des anomalies de fonctionnement, il exécute les tâches de diagnostic et de pronostic. Selon les symptômes du système et les conclusions du diagnostic et du pronostic, il prescrit des recommandations de maintenance ou suggère des changements opérationnels afin que le système puisse accomplir sa mission.

La première tâche d'un système de surveillance consiste à recueillir des données brutes sur le système sous surveillance. Cette fonction d'acquisition de données est généralement réalisée à l'aide de méthodes dites *non-destructives*, n'altérant pas le fonctionnement du système sous surveillance (IAEA, 2007). Suivant l'état de santé du système sous surveillance, différentes fonctions sont par la suite sollicitées. L'activité d'un système de surveillance diffère selon la présence ou non d'une défaillance au sein du système sous surveillance. La figure 1.5 ci-dessous présente les principales fonctions d'un système de surveillance avant l'occurrence d'une défaillance (surveillance dynamique) et après la défaillance du système sous surveillance (surveillance classique).

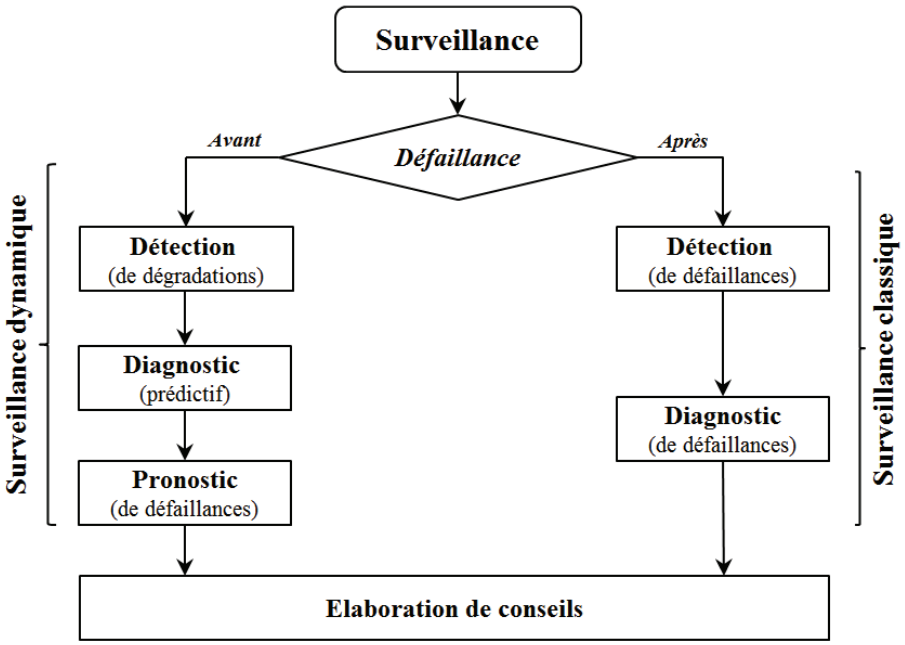

Figure 1.5. Fonctions de surveillance dynamique et classique, adaptée de (Raccoceanu, 2006).

Comme évoqué précédemment, la surveillance dynamique vise à éviter la défaillance du système sous surveillance. Elle consiste alors à détecter et à diagnostiquer des dégradations ou des baisses de performance du système sous surveillance (Bengtsson, 2003). Par la suite, l'objectif est de pouvoir prédire (pronostiquer) des défaillances dans le futur, en projetant l'état de santé actuel du système sous surveillance dans le temps (Tobon-Mejia et al., 2012). La surveillance dynamique regroupe ainsi les fonctions de détection prédictive de dégradations, de diagnostic prédictif et de pronostic. Notons que les résultats du diagnostic et du pronostic doivent être délivrés dans un temps opportun aux personnels de maintenance afin que des actions de maintenance puissent être exécutées sur le système avant la panne. De plus, cette surveillance dynamique doit être suffisamment robuste pour ne pas délivrer des alertes intempestives aux opérateurs de maintenance.

La surveillance classique intervient après l'occurrence d'une défaillance. Dans ce cas, cette surveillance consiste à détecter les défaillances survenues dans le système sous surveillance et à les diagnostiquer. Lorsque le système sous surveillance est en état de panne, le diagnostic permet de rechercher les causes et de localiser les éléments défaillants du système. Cette tâche peut s'avérer laborieuse pour un opérateur de maintenance, notamment lorsqu'il s'agit d'un système cible constitué de multiples éléments (Azarian et al., 2010 ; Khol et Bauer, 2010). Un diagnostic difficile peut conduire un opérateur de maintenance à déposer des éléments non contributeurs de la panne ou à remplacer des éléments non défaillants, engendrant des coûts excessifs de maintenance. Différentes approches ont ainsi été proposées pour automatiser cette tâche devenant trop complexe pour un opérateur de maintenance (Feldman, 2010).

Enfin, un système de surveillance s'appuie sur les informations de diagnostic et de pronostic pour élaborer des recommandations de maintenance et délivrer ces informations aux personnels de maintenance. Les différentes fonctions de surveillance illustrées sur la figure 1.5 sont détaillées dans les sections suivantes.

2.1 Détection

La détection relève de la composante "voir" d'un système de surveillance. A partir de données recueillies sur le système sous surveillance, la fonction détection permet de déceler la présence de dégradations (surveillance dynamique) ou de défaillances (surveillance classique) dans ce système, et donc de qualifier le comportement observé du système de normal ou d'anormal. La fonction détection génère des signaux porteurs de symptômes à partir des observations, également nommés *résidus* ou *discordances* par certaines communautés (Cordier et al., 2000 ; Roth et al., 2011). Par exemple, une approche usuelle permettant de générer des symptômes consiste à comparer certaines grandeurs observées à des seuils caractérisant une situation anormale. Le franchissement d'un seuil indique une anomalie de fonctionnement et provoque le déclenchement d'une alarme (Basseville et Cordier, 1996).

Les inconvénients de ces tests de dépassement de seuils résident dans la génération de fausses alarmes (seuil trop faible) et dans la non-détection de certains symptômes (seuil trop élevé). Certains équipements exécutent ce type de test pendant leur fonctionnement pour générer des alarmes et fournir une aide aux opérateurs de maintenance lors de la phase de diagnostic (Lefebvre, 2000). Cependant, les alarmes générées par ces tests intégrés (BIT : *Built-In Test*) représentent essentiellement des symptômes sans informations relatives à leurs causes. Le diagnostic établi par un opérateur de maintenance sur la base de ces événements est parfois erroné (Pecht et al., 2001).

2.2 Diagnostic

Le diagnostic prend en charge l'interprétation des anomalies de fonctionnement du système sous surveillance et constitue la composante "comprendre" d'un système de surveillance. La fonction diagnostic doit permettre de lever l'ambiguïté sur le phénomène à l'origine d'une dégradation ou d'une défaillance particulière à partir des symptômes délivrés par la fonction détection. Pour ce faire, la fonction diagnostic fait appel à une connaissance du système sous surveillance. Cette connaissance intègre, par exemple, une référence du fonctionnement normal (comportement nominal) ou du fonctionnement anormal (comportement en présence de défaillances) du système sous surveillance (Ribot, 2009). Les principales méthodes de diagnostic développées par différentes communautés de Recherche sont présentées en annexes de ce manuscrit.

La fonction diagnostic évalue l'état de santé actuel d'un système en recherchant les causes (fautes) à l'origine d'une dégradation (diagnostic prédictif) ou d'une défaillance particulière

(diagnostic de défaillances) et en localisant les éléments du système à remplacer ou à réparer. Il s'agit d'une fonction commune à la maintenance prédictive et à la maintenance corrective permettant d'incriminer des éléments défectueux et de déterminer les actions à entreprendre (IAEA, 2007). Le diagnostic prédictif se focalise avant tout sur l'analyse de défaillances progressives (dégradations) pour éviter la panne et planifier à l'avance les interventions de maintenance (commande de pièces de rechanges et gestion du personnel de maintenance, par exemple). Néanmoins, il existe toujours des défaillances soudaines (imprévisibles) ne pouvant être anticipées et devant faire l'objet d'un diagnostic (Jardine et al., 2006).

Contrairement aux signaux émis par la fonction détection, la fonction diagnostic est en mesure de fournir des informations de haut niveau à un opérateur de maintenance, en précisant les causes probables des dysfonctionnements détectés (Basseville et Cordier, 1996). Suivant les besoins des utilisateurs du système de surveillance, le résultat d'un diagnostic peut permettre d'identifier par exemple un composant physique défectueux ou un mode de défaillance (Mathur et al., 2001).

2.3 Pronostic

La fonction pronostic se concentre sur l'évolution des défaillances progressives et exploite généralement l'état de santé actuel d'un système (résultats du diagnostic prédictif) pour évaluer son état de santé futur. Bien qu'il n'existe aucune définition stabilisée et unanime du concept de pronostic, différentes définitions convergent vers une fonction capable de déterminer la durée de vie restante avant la défaillance d'un système, nommée RUL (*Remaining Useful Life*), ou la probabilité d'un système à fonctionner pendant un certain temps (El koujok, 2010). Le pronostic est spécifique à la maintenance prédictive et essentielle dans une approche CBM (Alanen et al., 2006). Il permet d'anticiper au mieux les interventions de maintenance suivant la prédiction de l'état futur d'un équipement (Gulledge et al., 2010). Par nature, le pronostic vise à anticiper l'état d'un système dans le futur et amène des incertitudes (Medina-Oliva et al., 2012). En effet, le pronostic de l'état futur d'un système doit non seulement considérer l'état actuel du système, mais également des données relatives à ce système dans le futur (prévisions d'utilisation du système, informations sur l'environnement futur du système, futures opérations de maintenance exécutées sur le système, par exemple), qui influent sur son état de dégradation. Ainsi, le pronostic doit prévoir l'état futur d'un système et fournir une durée de vie résiduelle à partir d'une connaissance à priori sur ce système (lois de dégradations, par exemple), d'une connaissance d'informations passées (historique des modes de fonctionnement passés), présentes (état courant) et futures (Cocheteux et al., 2007). Cette durée de vie résiduelle avant la défaillance peut être exprimée, par exemple, en unités de temps (heures de fonctionnement), en distances à parcourir (nombre de kilomètres) ou en nombre de sollicitations (nombre de cycles) (Ribot, 2009).

2.4 Elaboration de conseils

En fonction de l'état de santé actuel et de la prévision de l'état de santé du système, délivrés respectivement par les fonctions de surveillance classique et dynamique, le système de surveillance élabore des recommandations sur les opérations de maintenance à réaliser (actions préventives et correctives), ou privilégie un mode de fonctionnement ou une configuration à adopter pour que le système puisse achever sa mission en évitant la panne (Bengtsson, 2003). L'état de santé d'un système est le support principal pour orienter les services de maintenance sur les interventions à effectuer, et peut être défini par le diagnostic et le pronostic du système (Dievart, 2010). La fonction d'élaboration de conseils assiste principalement l'opérateur de maintenance dans le choix des actions de maintenance à

entreprendre (remplacement d'un élément du système en priorité selon son état de santé ou son RUL, par exemple). Par ailleurs, le bilan de santé dressé par cette fonction peut également alerter les services de conception, pour améliorer la qualité du système, revoir sa conception, ou impacter la conception et la fabrication d'une future génération de produits (Gulledge et al., 2010 ; Vichare et al., 2007). Cette fonction d'aide à la maintenance s'inscrit également dans les activités de gestion de la santé des équipements (*Equipment Health Management*) dans le cycle de vie des produits.

3. Modèle d'architecture de surveillance

Après avoir décrit précédemment les principales fonctions d'un système de surveillance, cette partie présente deux architectures considérées comme des références dans la littérature pour le développement de systèmes de surveillance. Une architecture sera retenue en tant que modèle dans nos travaux pour la surveillance des systèmes cibles considérés. Un modèle d'architecture de surveillance doit être suffisamment générique pour supporter de nombreux algorithmes de traitement et de diagnostic (Park et al., 2006). Parmi les architectures de surveillance recensées dans la littérature s'insérant dans les thématiques SHM-PHM (*System Health Management - Pronostics and Health Management*), une architecture apparaît comme un standard de référence (Gucik-Derigny, 2011 ; Kunche et al., 2012 ; Sreenuch et al., 2013). Il s'agit du standard OSA-CBM (*Open System Architecture - Condition-Based Maintenance*) diffusé par le consensus MIMOSA (*Machinery Information Management Open Systems Alliance)* (OSA-CBM, 2006).

Le standard OSA-CBM résulte initialement des travaux de Recherche de (Lebold et Thurston, 2001) et (Lebold et al., 2003), impliquant différents industriels et visant à définir un modèle d'architecture de surveillance. Ces travaux ont permis d'établir un modèle d'architecture pour le développement de systèmes de surveillance dans un cadre de maintenance CBM. Le modèle d'architecture proposé repose sur différentes couches fonctionnelles implémentées sous forme de modules. Ce modèle générique a donné lieu à une série de normes enregistrées sous le numéro ISO 13374 et à une implémentation de référence nommée OSA-CBM.

Dans cette partie, ces deux standards complémentaires d'architectures seront dans un premier temps présentés. Par la suite, les différentes problématiques liées à la surveillance des systèmes cibles seront exposées. Enfin, les principales exigences attendues d'un système de surveillance seront définies.

3.1 Standards d'architectures de surveillance

Les sous-sections suivantes précisent les différentes couches fonctionnelles définies par la norme ISO 13374, ainsi que les interfaces spécifiées par le standard OSA-CBM. Certains standards additionnels et spécifiques à la mise en œuvre d'un système de surveillance ne seront pas abordés dans nos travaux (interfaces physiques, configurations, interopérabilité, diffusion des informations, par exemple) (Sheppard et al., 2009).

3.1.1 Norme ISO 13374

Depuis 2003, la norme ISO 13374 établit les lignes directrices générales des spécifications pour la surveillance et le diagnostic d'état des machines. Ces spécifications logicielles portent sur le traitement, la communication et la présentation de données de surveillance indépendamment de la technologie utilisée (ISO13374-1, 2003). La première partie de cette norme (partie 1 : lignes directrices générales) définit un modèle d'architecture de surveillance

en six couches successives de traitement et deux couches d'interface transversales (figure 1.6). Les couches de traitement #1 à #3 sont liées à la technologie du système sous surveillance et les couches #4 à #6, de plus haut niveau, assurent les tâches de diagnostic, de pronostic et d'aide à la maintenance. Cette norme définit également deux couches d'interface (#7 et #8) permettant de présenter des informations issues des différentes couches de traitement aux utilisateurs, et d'assurer une liaison avec des systèmes externes (transmission et stockage des données, paramétrage des différentes couches, par exemple).

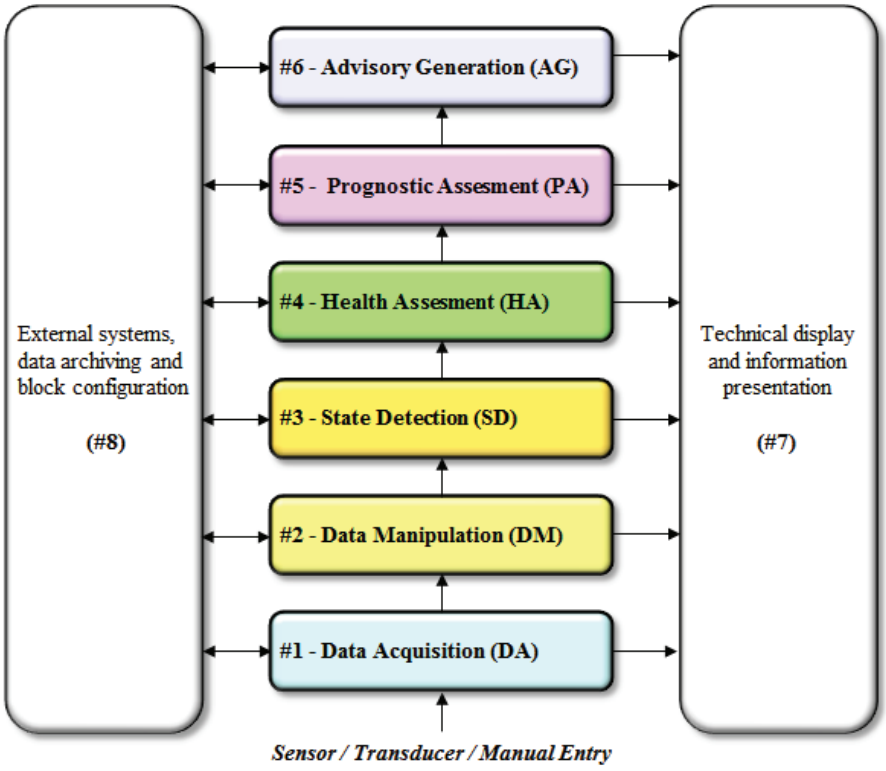

Figure 1.6. Modèle d'architecture ISO 13374, adaptée de (ISO 13374-1, 2003).

Les couches fonctionnelles définies par cette norme sont détaillées ci-après. Les termes associés aux différentes couches sont repris de la version originale de la norme.

- **Data Acquisition** (#1-DA) : cette couche d'acquisition de données permet de convertir un signal délivré par un capteur ou un transducteur en un paramètre numérique représentant une quantité physique, auquel est associé différentes informations (datation, données de calibration, configuration du capteur utilisé, par exemple).
- **Data Manipulation** (#2-DM) : cette couche de manipulation de données analyse des signaux, effectue un calcul de descripteurs significatifs (caractéristiques), et réalise des traitements et transformations de signaux à partir de mesures brutes.
- **State Detection** (#3-SD) : cette couche de détection d'état recherche des anomalies sur les signaux traités et doit déterminer la zone d'anomalie correspondante (si elle existe) sur laquelle se situe ces signaux, dans le but de générer des alertes ou des alarmes.

- **Health Assessment** (#4-HA) : cette couche d'évaluation de la santé réalise le diagnostic des fautes détectées et détermine l'état de santé courant de l'équipement ou du processus, en considérant l'ensemble des informations d'états.
- **Pronostic Assessment** (#5-PA) : cette couche d'évaluation pronostic détermine les états de santé et les modes de défaillances futurs à partir de l'évaluation de l'état de santé courant et de la projection de l'utilisation de l'équipement ou du processus, et détermine des prédictions de durées de vie résiduelle (*Remaining Useful Life*).
- **Advisory Generation** (#6-AG) : cette couche de génération de conseils fournit des informations claires et concrètes sur des actions de maintenance nécessaires ou concernant des modifications opérationnelles à apporter pour optimiser la durée de vie du processus et/ou de l'équipement.
- **Information Presentation** (#7) : cette couche est une interface permettant de présenter des données pertinentes issues des différentes couches à des personnels qualifiés, telles que des courbes de tendances et des zones d'anomalies associées. Elle doit fournir aux analystes les données nécessaires pour identifier, confirmer ou comprendre un état anormal. Par ailleurs, cette couche assure une transformation de données en une forme présentant clairement les informations nécessaires pour prendre des décisions sur les actions de maintenance à réaliser.
- **External Systems** (#8) : cette couche permet une liaison rapide avec des systèmes externes, et notamment avec un système de maintenance (accès à l'historique des interventions de maintenance, transmission des recommandations de maintenance ou de requêtes d'intervention de maintenance, par exemple), et avec un système de contrôle d'exploitation (accès aux prévisions de l'utilisation de l'équipement, transmission de requêtes de changements opérationnels ou de mise hors service de l'équipement, par exemple).

Le concept d'état de santé mentionné dans la norme ISO 13374 (*health state*) n'est pas clairement défini. L'état de santé d'un équipement repose généralement sur l'expertise métier et dépend des traitements réalisés par les couches #4 à #6 du modèle. Néanmoins, la première partie de cette norme évoque un indice de santé associé à un équipement (*health index*) allant de 0 (défaillance complète de l'équipement) à 10 (équipement comme neuf).

La seconde partie de la norme ISO 13374 (partie 2 : traitement des données) caractérise les différents modules associés aux couches successives de traitement présentées sur la figure 1.6 (ISO 13374-2, 2007). Un module peut implémenter des fonctionnalités relatives à une ou plusieurs couches de traitement du modèle. De plus, cette norme suggère la manière dont ces modules peuvent être agencés pour former une architecture opérationnelle de surveillance. Ces modules peuvent notamment être répartis sur différents supports d'exécution interconnectés entre eux. La figure 1.7 ci-dessous illustre un exemple d'architecture opérationnelle, dans laquelle chaque module est supporté par un dispositif matériel et implémente les fonctionnalités d'une couche du modèle.

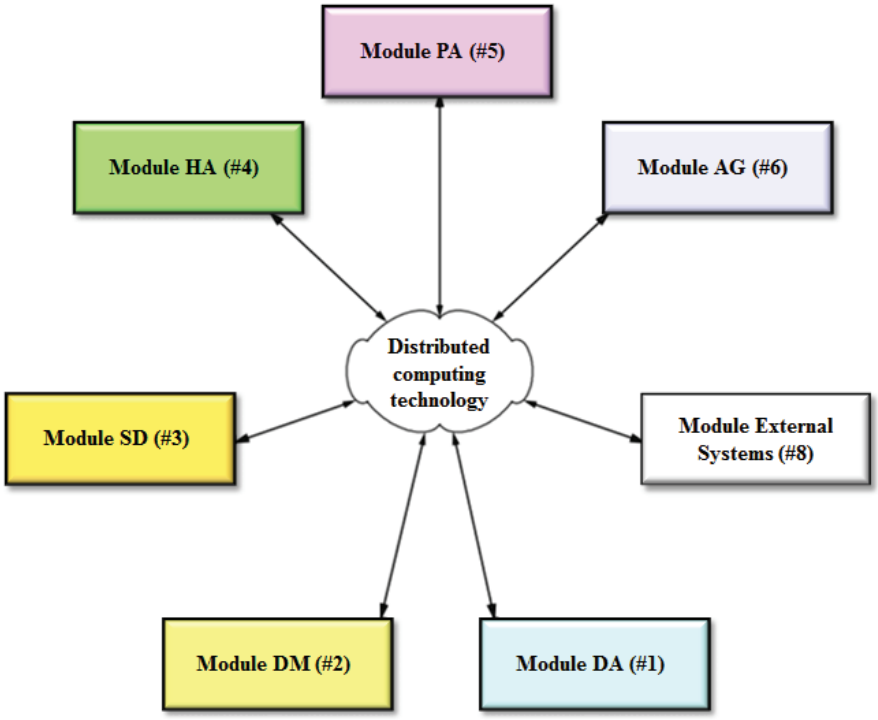

Figure 1.7. Exemple d'architecture opérationnelle, adaptée de (ISO 13374-2, 2007).

Le medium de communication situé au centre de la figure 1.7, représente le support des différentes communications entre ces modules et peut être réalisé au travers de différentes technologies. Ainsi, les modules de l'architecture ne résident pas nécessairement sur le même support et peuvent être distants les uns des autres. Cette approche permet également d'intégrer des modules développés par différents fournisseurs pour former un système de surveillance modulaire. Des exemples d'échanges d'informations entre ces modules sont proposés dans la troisième partie de la norme (partie 3 : échange), diffusée en 2012 (ISO 13374-3, 2012).

3.1.2 OSA-CBM

Le standard OSA-CBM implémente précisément les six couches de traitement définies par la norme ISO 13374 (figure 1.5). Comme la norme ISO 13374, le standard OSA-CBM ne spécifie pas les traitements et les algorithmes internes aux différents modules. En revanche, ce standard définit les types et les structures de données à utiliser pour le traitement et la communication des résultats, ainsi que la manière dont les informations doivent être transférées entre les modules et les lieux de stockage (OSA-CBM, 2006). Il s'agit d'un standard d'interface, imposant des interfaces entre les modules d'un système de surveillance suivant une approche CBM, et favorisant l'interopérabilité entre ces modules. Ainsi, différents fournisseurs peuvent proposer des algorithmes propriétaires sous forme de modules au standard OSA-CBM. Ce standard permet de dissocier les algorithmes de traitements encapsulés dans les modules (algorithmes propriétaires) des interfaces génériques des modules (figure 1.8).

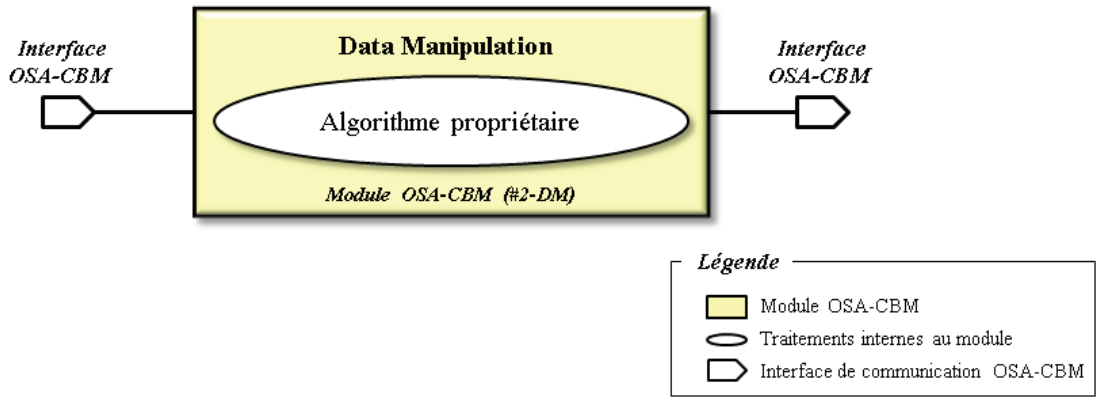

Figure 1.8. Exemple de module OSA-CBM, adaptée de (OSA-CBM, 2006).

Les spécifications du standard OSA-CBM sont diffusées sous forme de diagrammes de classes UML (*Unified Modeling Language*) et sont indépendantes de la technologie mise en œuvre pour le développement des modules de traitement. Le concepteur du système de surveillance doit alors sélectionner une technologie spécifique pour la mise en œuvre des modules de traitement et définir les communications nécessaires entre ces modules. Le standard OSA-CBM définit une typologie des échanges possibles entre modules sur la base des recommandations de la norme ISO 13374 (partie 3 : communication). La figure 1.9 ci-dessous illustre les différents types de communications définis par le standard OSA-CBM.

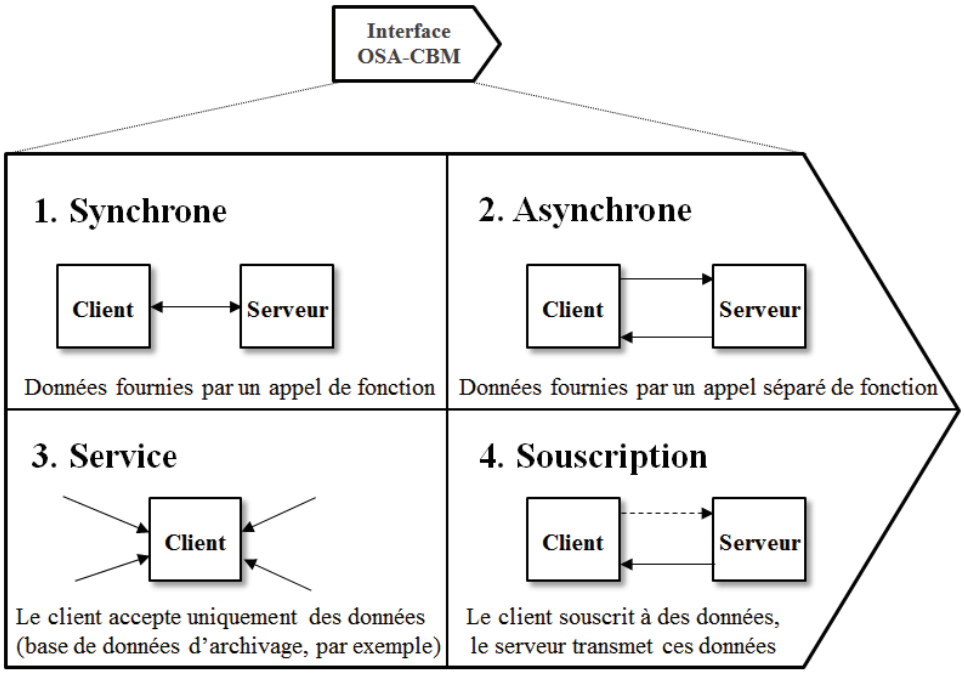

Figure 1.9. Interfaces entre modules, adaptée de (OSA-CBM, 2006).

Un module OSA-CBM peut communiquer avec un ou plusieurs modules selon quatre types d'interface :

- **Synchrone** (type 1) : ce mécanisme permet un échange direct entre un module producteur d'informations (serveur) et un module consommateur (client). Un module consommateur indique à un module producteur les informations qu'il souhaite recevoir et attend que ces informations soient disponibles et transmises par le module producteur avant de poursuivre ses traitements.
- **Asynchrone** (type 2) : ce mécanisme permet à deux modules d'entretenir et de maintenir une communication bidirectionnelle. Dans ce mode, un module consommateur souhaitant obtenir des données de la part d'un module producteur n'attend pas que ces données soient disponibles avant de continuer ses traitements (dialogue de type requêtes-réponses entre deux modules, par exemple). Par ailleurs, un module producteur peut transmettre spontanément des informations à un ou plusieurs modules consommateurs de manière périodique ou événementielle.
- **Service** (type 3) : ce mécanisme permet une communication unidirectionnelle entre un ou plusieurs modules producteurs d'informations et un module consommateur. Cette interface peut être mise en œuvre, par exemple, pour des besoins de stockage d'informations (archivage des informations par un module consommateur), ou pour la gestion des requêtes d'intervention de maintenance (transmission des ordres de maintenance vers un système de maintenance consommateur).
- **Souscription** (type 4) : ce mécanisme permet à un module consommateur de souscrire à des informations auprès d'un module producteur. Le module consommateur souscripteur reçoit alors ces informations de manière asynchrone (régulièrement ou sur alertes), dès que les informations élaborées par le module producteur sont disponibles.

Les modules de traitement sont généralement successifs, afin de transformer des données brutes collectées sur un système sous surveillance en informations pertinentes. Cependant, ces liens logiques entre modules ne sont pas imposés par le standard OSA-CBM ou la norme ISO 1337. Ainsi, un module peut en principe accéder à n'importe quel autre module lorsque ces modules partagent un même medium de communication. Le standard OSA-CBM a été développé sur la base d'un autre standard MIMOSA, décrit selon un même système d'information relationnel commun (CRIS) : le standard OSA-EAI (*Open System Architecture Enterprise Application Integration*). Il s'agit d'un standard visant à faciliter l'échange d'informations entre différentes applications métiers d'une entreprise, y compris des informations relatives à la maintenance, au diagnostic et au pronostic (OSA-EAI, 2010).

La norme ISO 13374 étant moins restrictive sur la mise en œuvre des couches fonctionnelles du modèle que le standard OSA-CBM, elle sera retenue dans la suite de nos travaux. La section suivante soulève les différentes problématiques liées à la surveillance des systèmes cibles suivant les spécifications de cette norme.

3.2 Problématiques de la surveillance des systèmes cibles considérés

La norme ISO 13374 a été initialement développée pour la surveillance de machines ou d'équipements industriels isolés (machine tournante, processus d'usinage, par exemple). En l'absence de standards dédiés à la surveillance des systèmes cibles considérés dans nos travaux, ce modèle offre une décomposition fonctionnelle générique d'un système de surveillance. De plus, cette norme constitue un modèle de référence pour la surveillance de certains systèmes cibles, notamment dans les domaines des transports et de l'industrie (Gorinevsky et al., 2010 ; Naedele et al., 2004).

Cependant, des difficultés se présentent lorsque ce modèle d'architecture doit être instancié pour la surveillance d'un système cible. La figure 1.10 ci-dessous illustre les principales difficultés à appréhender pour la surveillance d'un tel système suivant ce modèle d'architecture. A partir des six couches fonctionnelles de ce modèle, plusieurs alternatives se présentent aux concepteurs d'un système de surveillance. Ces couches fonctionnelles peuvent être réparties selon deux axes :

- **répartition "verticale" des couches** (1) : les traitements associés aux couches fonctionnelles ISO 13374 peuvent être réalisés au plus près du système cible (traitements embarqués) ou être pris en charge par le centre de maintenance (traitements distants).
- **répartition "horizontale" des couches** (2) : la surveillance d'un système cible passe par la surveillance de ses sous-systèmes. Par conséquent, les traitements associés aux couches fonctionnelles ISO 13374 doivent être déployés suivant les sous-systèmes du système cible, dont la technologie diffère d'un sous-système à un autre.

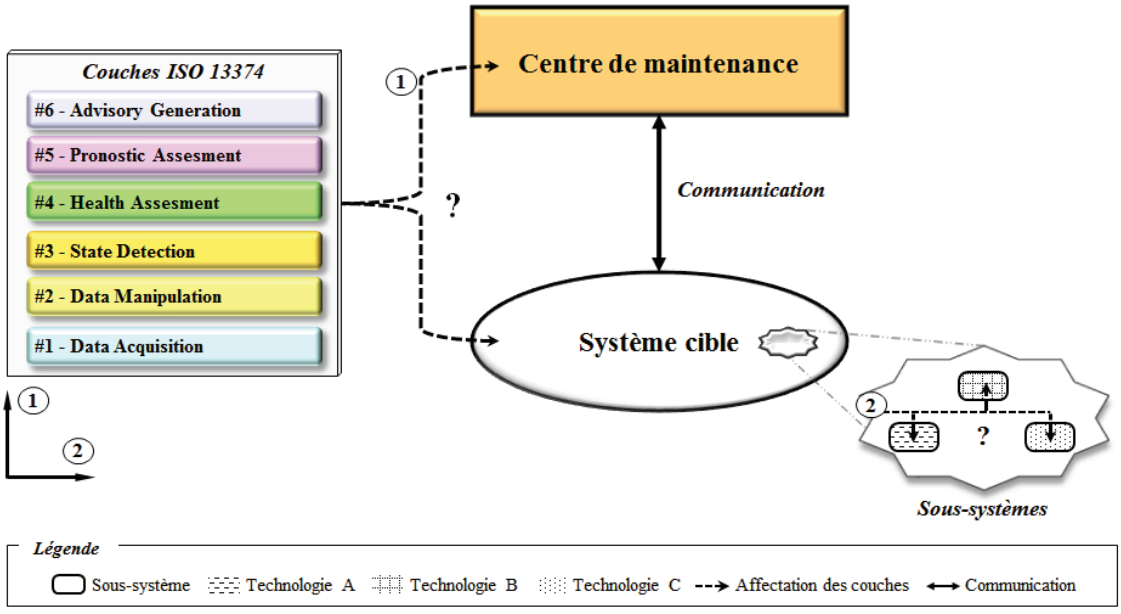

Figure 1.10. Surveillance d'un système cible suivant les couches fonctionnelles ISO 13374.

Les sous-sections suivantes précisent les principales contraintes à considérer pour la surveillance de systèmes cibles. Notons que ces contraintes dépendent des domaines d'applications des systèmes cibles introduits au début de ce chapitre. Dans ces travaux, le domaine des transports ferroviaires a été retenu comme cadre applicatif. Par conséquent, la suite de ce manuscrit se concentre sur la surveillance de systèmes cibles mobiles.

3.2.1 Contraintes liées à la mobilité du système cible

En considérant la répartition "verticale" des couches du modèle, une séparation des couches doit être établie suivant les traitements à effectuer à bord du système cible et les traitements à réaliser au centre de maintenance. La première couche du modèle (#1-*Data Acquisition*) doit au minimum être implémentée à bord du système cible, pour permettre la collecte des données et la transmission de ces données au centre de maintenance. Lorsque la quantité de données à transmettre est importante, il est préférable d'inclure la couche #2 du modèle à bord (Alanen et al., 2006). Ainsi, plus les couches du modèle résident à bord du système cible, plus les traitements embarqués sont évolués et permettent de délivrer des informations pertinentes au centre de maintenance. Cependant, il est nécessaire de tenir compte des performances requises pour mettre en œuvre les traitements associés aux couches fonctionnelles du modèle (puissance de calcul disponible à bord, capacités de stockage et de communication, par exemple).

Dans le but d'optimiser les interventions de maintenance, les données relatives à la géolocalisation du système cible peuvent également être transmises au centre de maintenance. De ce fait, le système cible peut être acheminé vers l'atelier de maintenance le plus proche en fonction de son état de santé (Umiliacchi et al., 2011). Aussi, certains traitements nécessitent des données fournies par le centre de maintenance (historique des tâches de maintenance, prévisions d'utilisation du système cible, par exemple). Lorsque ces traitements sont embarqués, le centre de maintenance doit transmettre ces données au système cible. Une communication bidirectionnelle entre le système cible mobile et le centre de maintenance est alors nécessaire.

3.2.2 Contraintes liées aux sous-systèmes du système cible

Suivant les couches à implémenter à bord du système cible, la répartition "horizontale" des couches concerne l'affectation des modules de traitement aux différents sous-systèmes. Comme abordé précédemment, un module de traitement peut implémenter les fonctionnalités de une ou de plusieurs couches du modèle ISO 13374. De ce fait, un module de traitement peut être affecté à un sous-système, à un groupe de sous-systèmes, ou à l'ensemble des sous-systèmes. Les contraintes associées à la mise en œuvre de ces modules de traitement sont précisées ci-après.

- **Répartition spatiale des sous-systèmes :** les sous-systèmes sont généralement implantés dans différentes zones du système cible et peuvent interagir entre eux (interactions physiques et/ou informationnelles). L'affectation des modules de traitement aux sous-systèmes peut alors être orientée suivant l'implantation de ces sous-systèmes. Aussi, les interactions entre les sous-systèmes sont à considérer lors de la mise en œuvre des modules de traitements. Par exemple, une défaillance peut engendrer des dysfonctionnements au niveau de plusieurs sous-systèmes d'un système cible (Lefebvre et al., 2000). De multiples alarmes peuvent ainsi être générées

lorsqu'un module relatif à la couche détection du modèle ISO 13374 (#3-State Detection) est affecté à chaque sous-système.

- **Hétérogénéité technologique des sous-systèmes :** les sous-systèmes reposent sur des technologies différentes. Par conséquent, un sous-système basé sur une certaine technologie doit être associé à un module de traitement compatible avec cette technologie. De plus, le modèle ISO 13374 se concentre sur la surveillance de parties physiques d'un équipement. Cependant, les sous-systèmes considérés sont constitués de parties physiques et informationnelles. En effet, les défaillances logicielles non abordées dans la norme ISO 13374 ont un impact direct sur la disponibilité du système cible. Notons que la surveillance de parties logicielles peut être abordée de la même manière que la surveillance de parties physiques (Pipatsrisawat et al., 2009). Néanmoins, les défaillances logicielles se manifestent principalement de manière soudaine et peuvent difficilement être anticipées.

- **Variabilité des sous-systèmes :** les sous-systèmes sont conçus par différents fournisseurs et peuvent varier entre plusieurs systèmes cibles d'une même génération (changement de fournisseur, obsolescence de composants, ajout de fonctionnalités, par exemple) (Azarian et al., 2011). La répartition "horizontale" des couches du modèle doit alors être adaptée aux évolutions des différents sous-systèmes du système cible. Par exemple, un même sous-système peut présenter des caractéristiques différentes entre deux systèmes cibles, ou ne pas être implanté sur un autre système cible. Dans ce cas, une modification doit pouvoir être apportée aux modules de traitement sans impliquer une refonte de l'architecture de surveillance. Aussi, certains modules de traitement peuvent être réutilisés pour la surveillance de sous-systèmes ayant les mêmes caractéristiques.

3.2.4 Contraintes liées à l'environnement du système cible

Lorsque le système cible est immergé dans un environnement, il est soumis à différents contextes opérationnels (Monnin et al., 2011). L'environnement du système cible peut potentiellement provoquer des défaillances lors de son fonctionnement. Ces défaillances sont alors signalées au centre de maintenance par le système de surveillance associé au système cible. La figure 1.11 ci-après illustre un exemple de système cible mobile situé dans son environnement. Dans cet exemple, les couches #1 à #3 du modèle ISO 13374 sont implémentées à bord du système cible et les couches #4 à #6 sont assurées par le centre de maintenance (répartition "verticale" des couches). La défaillance du système cible est supposée apparaitre sous l'effet de perturbations provenant de son environnement. Cette défaillance déclenche alors des alarmes au niveau de la couche #3 du modèle. Ces alarmes sont par la suite transmises au centre de maintenance et analysées par un opérateur de maintenance ou traitées par les couches #4 à #6 du modèle lorsqu'elles sont implémentées au sol.

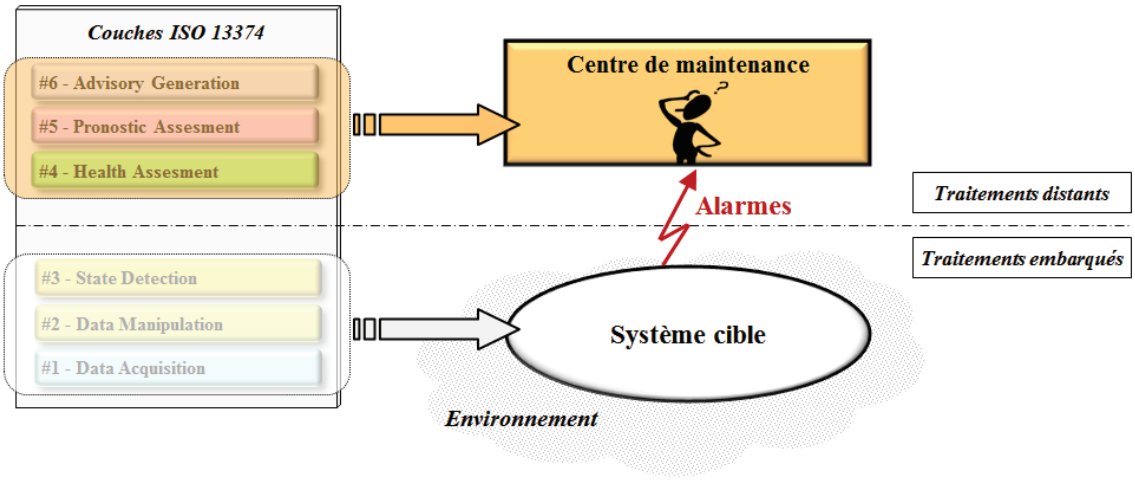

Figure 1.11. Exemple de surveillance d'un système cible mobile dans son environnement.

L'analyse de ces alarmes au sol est délicate pour un opérateur de maintenance. Le contexte opérationnel du système cible n'étant pas connu, l'opérateur n'est pas en mesure de déterminer si ces alarmes sont liées à l'environnement du système cible ou si le système cible nécessite réellement une intervention de maintenance. Ainsi, le diagnostic établi au sol par la quatrième couche du modèle à partir de ces alarmes risque d'être erroné. En effet, les informations de contextes non exploitées par la couche diagnostic (#4- *Health Assesment*) sont essentielles pour interpréter les dysfonctionnements du système cible. De plus, lorsque le système cible arrive au centre de maintenance, aucun dysfonctionnement ne peut être constaté. Le système cible n'opérant plus dans le même contexte opérationnel, il est alors difficile de reproduire les conditions d'exploitation ayant conduit à une anomalie de fonctionnement. Ce phénomène nommé NFF (*No Fault Found*) favorise le remplacement à tort d'éléments du système cible et engendre des coûts importants de maintenance (Lanigan et al., 2011).

A titre d'exemple, il peut être judicieux d'associer à une défaillance la localisation précise du système cible, qui peut être fournie aisément par un dispositif de géolocalisation (Bengston, 2003). Ceci permet d'écarter certaines défaillances liées à l'environnement du système cible.

3.2.5 Exigences d'un système de surveillance

Les principales exigences associées à un système de surveillance peuvent être établies à partir du modèle d'architecture proposé par la norme ISO 13374. Une exigence fondamentale concerne la surveillance d'un système cible lors de son fonctionnement (surveillance *en-ligne*). Un système de surveillance ne doit pas altérer la performance du système cible sous surveillance. Par conséquent, les moyens d'acquisition de données associés à la première couche du modèle ISO 13374 doivent être non-intrusifs vis-à-vis du système cible sous surveillance. Dans nos travaux, un système de surveillance doit également répondre aux exigences suivantes :

- Précision : le système de surveillance doit identifier précisément des éléments défectueux à remplacer ou à réparer du système cible parmi les différents sous-systèmes qui composent ce système cible.

- Facilité d'explication : le système de surveillance doit faciliter la compréhension des informations de surveillance fournies aux opérateurs de maintenance (présentation de données brutes ayant conduit à un résultat de diagnostic, par exemple).
- Adaptabilité : le système de surveillance doit être suffisamment modulaire pour faire face aux changements survenant dans le système cible (remplacement de sous-systèmes, ajout ou suppression de composants, par exemple).
- Réactivité : les informations de surveillance doivent être transmises dans un temps opportun au centre de maintenance afin d'optimiser les interventions de maintenance (planification d'une opération de maintenance suivant l'état de certains sous-systèmes du système cible, par exemple).
- Confiance : le système de surveillance doit être suffisamment robuste pour ne pas générer de fausses alarmes (diagnostic erroné transmis à un opérateur de maintenance, par exemple).

4. Conclusion

Ce premier chapitre a permis d'introduire les propriétés des systèmes cibles considérés dans nos travaux ainsi que les différentes notions relatives à la surveillance dans un contexte de maintenance. Au sein d'un système de maintenance, la surveillance peut permettre d'optimiser efficacement les interventions de maintenance préventive et corrective et accroître ainsi la disponibilité d'un système cible. Un système de surveillance assure deux composantes principales : percevoir les dysfonctionnements survenant dans un système sous surveillance et interpréter ces dysfonctionnements. Le diagnostic prend en charge l'interprétation des dysfonctionnements observés sur le système sous surveillance. Cette fonction essentielle doit permettre de déterminer les éléments défectueux du système à remplacer ou à réparer qui seront présentés aux opérateurs de maintenance.

Dans ce chapitre, deux standards complémentaires d'architectures de surveillance ont été présentés : un modèle proposé par la norme ISO 13374 et une implémentation de référence de ce modèle nommé OSA-CBM. Ces deux standards décomposent un système de surveillance en six couches fonctionnelles successives. Le modèle prescrit par la norme ISO 13374 s'avère plus générique que le standard OSA-CBM et a été retenu dans le cadre de nos travaux. Cependant, la surveillance d'un système cible suivant ce modèle présente certaines difficultés exposées à la fin de ce chapitre. En effet, un système cible est constitué de sous-systèmes hétérogènes et soumis à différents contextes opérationnels. De plus, un système cible mobile est rattaché à un centre de maintenance distant.

La problématique principale est de concevoir une architecture de surveillance reposant sur la norme ISO 13374 et répondant aux exigences listées en 3.2.5. Cette architecture doit permettre de surveiller un système cible mobile qui est lui-même composé de systèmes. Cette problématique constitue le verrou scientifique auquel nos travaux s'adressent.

Le chapitre suivant se focalise sur la surveillance de systèmes cibles mobiles et recense les différentes approches de la littérature pour établir le diagnostic d'un tel système suivant la quatrième couche (*Health Assesment*) définie par la norme ISO 13374.

Chapitre II

Surveillance de systèmes mobiles : état de l'art et approches de surveillance active

Introduction

Le précédent chapitre a mis en évidence un ensemble de problématiques et contraintes à considérer pour concevoir une architecture de surveillance associée à un système cible. Ce chapitre explore les différentes approches pour la surveillance de systèmes cibles mobiles (système de transport aérien, maritime ou ferroviaire, par exemple). Comme évoqué dans le précédent chapitre, ces systèmes cibles évoluent dans différents contextes opérationnels et communiquent avec un centre de maintenance distant. La première partie de ce chapitre présente un état de l'art des architectures de diagnostic et propose une typologie des architectures pour diagnostiquer un système cible mobile. A partir des exigences d'un système de surveillance établies dans le précédent chapitre, une architecture de diagnostic sera retenue dans la suite de nos travaux. La seconde partie de ce chapitre s'appuie sur le concept de surveillance active pour proposer des approches de modélisation de l'architecture de diagnostic choisie. Une approche de modélisation sera adoptée et reprise dans le chapitre suivant afin de proposer une architecture générique pour la surveillance de systèmes cibles mobiles.

1. Etat de l'art des architectures de diagnostic

La norme ISO 13374 détaillée dans le précédent chapitre définit un modèle générique d'architecture de surveillance en six couches fonctionnelles successives, dont la quatrième couche (#4-*Health Assesment*) élabore le diagnostic à partir de données fournies par les couches de niveaux inférieurs. Le diagnostic joue un rôle décisif dans la performance d'une architecture de surveillance, et peut également impacter la sécurité du système cible (Koutsoukos et al., 2010). En effet, le diagnostic incrimine des éléments potentiellement responsables d'une défaillance ou d'une dégradation particulière et implique des actions de maintenance bien précises. De notre point de vue, deux alternatives se dégagent pour réaliser le diagnostic d'un système cible mobile rattaché à un centre de maintenance distant. La première consiste à collecter des données brutes sur le système cible et à transférer ces données vers un centre de maintenance, où le diagnostic est exécuté. Dans cette première approche, le système cible peut être vu comme un dispositif d'acquisition de données, générant un flux de données brutes envoyé vers le centre de maintenance. Dans la seconde approche, le système cible est considéré comme un dispositif de calcul, capable de traiter et d'interpréter ces données brutes. Le diagnostic réside alors à bord du système cible, et seules des informations pertinentes de diagnostic sont délivrées au centre de maintenance (Alanen et al., 2006 ; Bengtson, 2003).

Ces deux approches, nommées respectivement *diagnostic distant* et *diagnostic embarqué* sont illustrées par la figure 2.1. Lorsque le diagnostic est distant (figure 2.1 a), les couches #1 à #3 du modèle ISO 13374 sont implémentées à bord du système cible, tandis que les couches #4 à #6 sont prises en charge par le centre de maintenance. Dans une approche de diagnostic embarqué (figure 2.1 b), les couches #1 à #4 du modèle sont implémentées à bord du système cible et les couches #5 et #6 sont assurées par le centre de maintenance.

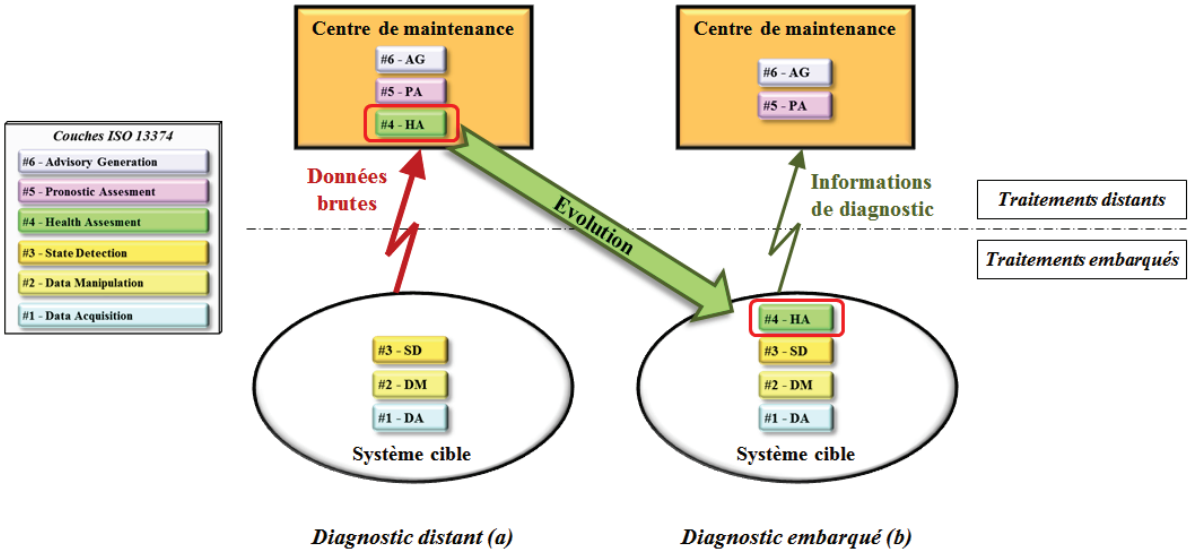

Figure 2.1. Partitionnement du diagnostic : diagnostic distant (a) et diagnostic embarqué (b).

Suivant cette dichotomie, cette partie recense les principales architectures de diagnostic pour la surveillance de systèmes cibles mobiles, en faisant abstraction des méthodes de diagnostic mises en œuvre. Ainsi, une typologie des architectures de diagnostic sera établie indépendamment des traitements internes réalisés par les couches du modèle ISO 13374.

1.1 Diagnostic distant

Cette approche consiste à réaliser le diagnostic à distance dans un centre de maintenance, à partir de données collectées et prétraitées à bord du système cible par les couches #1 à #3 du modèle. Les alarmes et autres signaux brutes du système cible sont alors transférés et analysés au centre de maintenance. Le diagnostic distant est généralement réalisé de manière centralisé et doit traiter une masse de données brutes pour rechercher les causes des symptômes détectés à bord du système cible. Cette architecture, nommée Diagnostic Distant Centralisé (DDC), permet d'exploiter pleinement les capacités de stockage et de traitement disponibles au centre de maintenance. Cependant, le diagnostic peut difficilement être établi pendant le fonctionnement du système cible en raison des contraintes de communication. En effet, des volumes de données importants doivent être transférés du système cible mobile vers le centre de maintenance, impliquant des délais de transmission et de traitement. Par conséquent, les données nécessaires à l'élaboration d'un diagnostic sont généralement stockées à bord du système cible et transmises de manière périodique ou événementielle au centre de maintenance (Jianjun et al., 2007). Ces données sont alors enregistrées *en-ligne* pendant le

fonctionnement du système et le diagnostic est exécuté *hors-ligne* à posteriori (approche de type "boîte noire").

L'architecture DDC permet également d'assurer le diagnostic de plusieurs systèmes cibles mobiles (flotte de trains ou de navires, par exemple) (Monnin et al., 2011). D'un point de vue fiabilité, cette architecture présente un inconvénient majeur. Le diagnostic des différents systèmes cibles repose sur un seul et unique centre de maintenance. L'occurrence d'une défaillance en ce point sensible de l'architecture entraine une interruption du diagnostic pour l'ensemble des systèmes cibles (Ferrari, 2009). Par ailleurs, en examinant les exigences d'un système de surveillance établies dans le précédent chapitre, l'architecture DDC présente les limitations suivantes :

- Précision : les données traitées au centre de maintenance représentent essentiellement des symptômes émis sous forme d'alarmes. L'interprétation de ces événements par une fonction de diagnostic au sol est parfois erronée (Azarian et al., 2011). Dans certains domaines applicatifs des systèmes cibles, le diagnostic est couramment achevé manuellement au sol par un expert à partir des alarmes générées à bord du système cible (De Ambrosi et al., 2008).
- Réactivité : cette approche implique des quantités de données considérables à transférer au centre de maintenance, par exemple, par le biais de liaisons sans fil (technologies GPRS, GSMR ou Wifi) (Schwarzenbach et al., 2010). Ces transferts de données introduisent des retards non négligeables dans l'élaboration d'un diagnostic, et par conséquent, dans la planification des opérations de maintenance.
- Confiance : les symptômes détectés à bord du système cible ne sont pas contextualisés. Ainsi, une alarme peut être déclenchée par l'environnement du système cible ou être provoquée par des interférences entre certains sous-systèmes (défaillances fugitives, par exemple). Notons également que certaines données peuvent être superflues (alarmes redondantes, alarmes liées aux actions du conducteur ou émises lors des interventions de maintenance, par exemple) (Sammouri et al., 2012). Ces données sont néanmoins transmises et archivées au centre de maintenance, engendrant des coûts injustifiés de transmission et de stockage. Ainsi, le contexte opérationnel du système cible n'est pas considéré lors de l'élaboration d'un diagnostic sur la base de ces alarmes.

Néanmoins, cette architecture est relativement simple à mettre en œuvre et permet à des systèmes d'informations externes d'exploiter aisément les données collectées.

Plusieurs implémentations de l'architecture DDC ont été réalisées dans le domaine des transports. Par exemple, la première génération de systèmes HUMS (*Health and Usage Monitoring System*), apparue au début des années 90, était limitée à la collecte de données et à la génération d'alarmes sur certains aéronefs (Wiig, 2006). Les principaux inconvénients de ce système résident dans la génération de fausses alarmes et dans l'interprétation de ces masses de données brutes au centre de maintenance. Dans le domaine ferroviaire, le projet Européen TrainCom a contribué à standardiser les échanges de données entre un matériel roulant ferroviaire et un centre de maintenance distant (Gatti, 2002). La communication entre le véhicule ferroviaire et un centre de maintenance au sol (ROGS : *Railway Open Ground Station*) est assurée par une interface de communication nommée ROGate (*Railway Open Gateway*). Ces spécifications ont permis à différentes applications situées au centre de maintenance de traiter les données collectées à bord des trains. Cette infrastructure de

communication a été expérimentée en Europe dans le cadre du projet EuRoMain (*European Railway Open Maintenance System*) (Euromain, 2002). Le principal objectif de ce projet était de définir un système d'aide à la maintenance de véhicules ferroviaires en réalisant le diagnostic à distance des différents équipements embarqués.

Actuellement, les deux principaux constructeurs mondiaux de matériels ferroviaires ont adopté une architecture DDC. Les systèmes Orbita (Orbita, 2006) et TrainTracer (TrainTracer, 2006) développés respectivement par Bombardier-Transport et Alstom-Transport permettent de prélever des données en temps réel sur des véhicules ferroviaires et de les transférer vers un système central (figure 2.2).

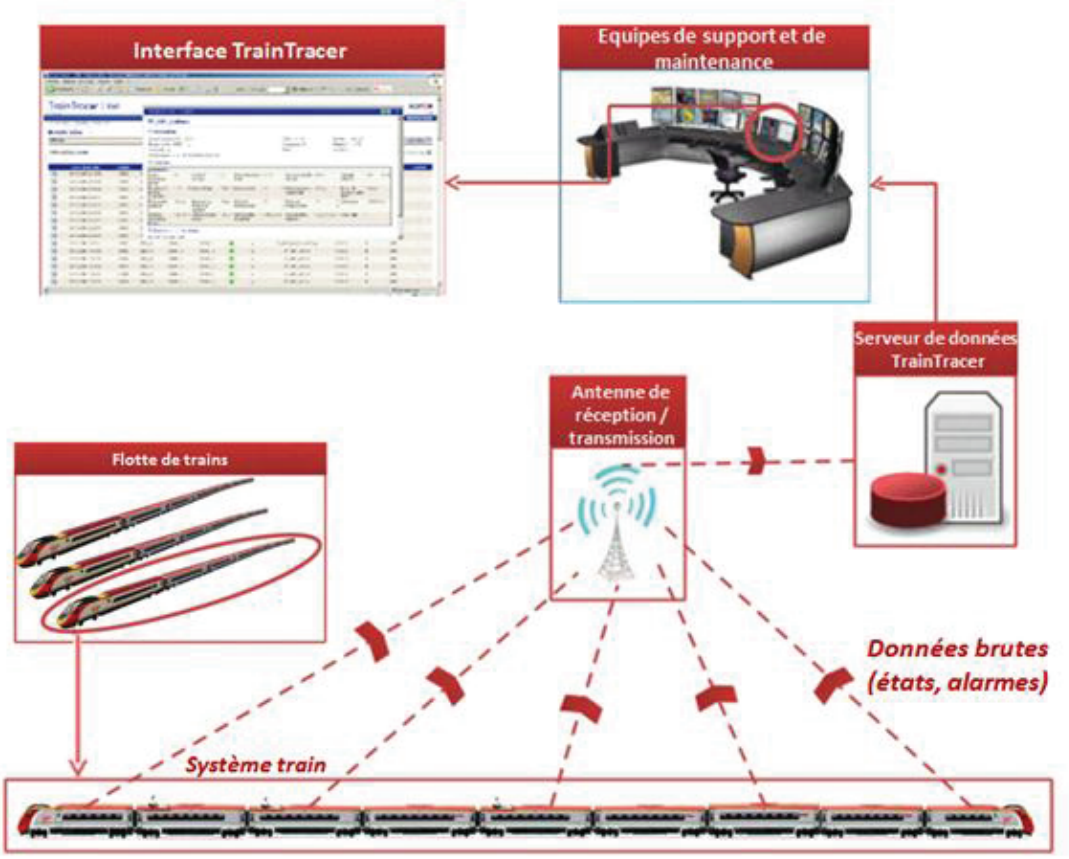

Figure 2.2. Architecture du système TrainTracer, adaptée de (Sammouri et al., 2012).

Les données relatives aux différents équipements implantés à bord des véhicules (accès voyageurs, climatisations, tractions, par exemple) sont transmises et stockées systématiquement sur un serveur de données (figure 2.2). Ces données brutes (états et alarmes des différents équipements embarqués, par exemple) sont exploitées par le personnel de maintenance ou traitées par différentes applications (extraction de données, analyse de tendances, diagnostic, par exemple). Ces applications sont développées par les constructeurs ou les exploitants de matériels ferroviaires.

1.2 Diagnostic embarqué

Lorsque la quatrième couche du modèle ISO 13374 (#4-*Health Assesment*) est intégrée au sein du système cible, le diagnostic est embarqué. Cette approche vise à exécuter le diagnostic pendant le fonctionnement du système cible et offre plus d'autonomie et de réactivité. Contrairement à une architecture DDC, les données brutes sont converties en informations de

diagnostic à bord du système cible (figure 2.3). Les délais et coûts de transmission sont alors réduits car seules des informations pertinentes émanent du système cible. De plus, le diagnostic embarqué permet d'exploiter des informations contextuelles disponibles à bord du système cible. Ces informations complémentaires contribuent largement à améliorer la performance du diagnostic embarqué (Byington et al., 2003).

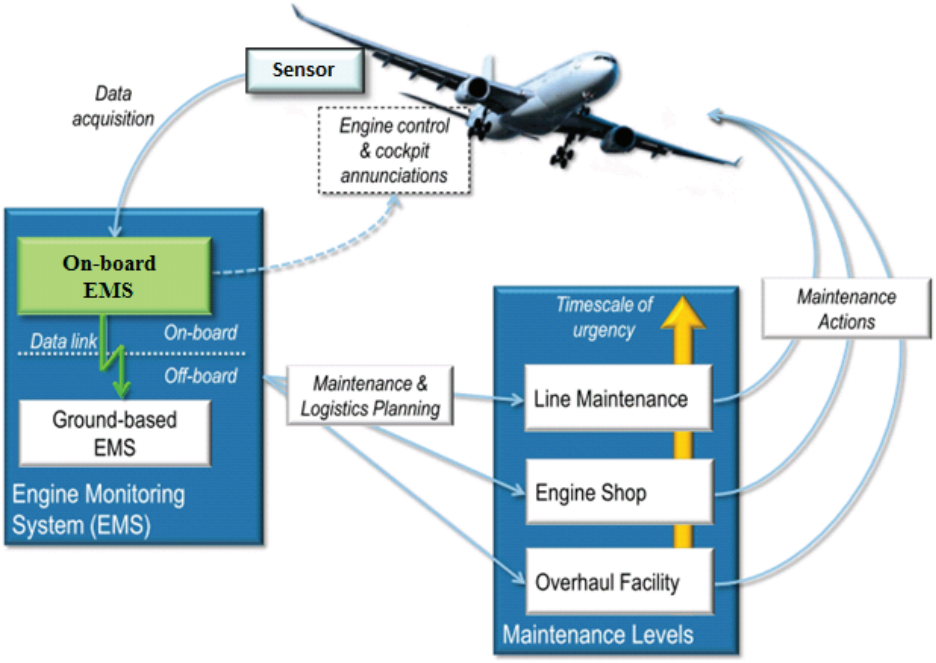

Figure 2.3. Exemple de diagnostic embarqué, adaptée de (Hunter et al., 2013).

Le diagnostic d'un système cible repose sur le diagnostic des différents sous-systèmes qui le composent (Kurien et al., 2002). Une première approche permettant de réaliser le diagnostic à bord d'un tel système consiste à déployer une couche de diagnostic sur l'ensemble des sous-systèmes. Ainsi, une entité centrale de diagnostic interprète les données relatives aux différents sous-systèmes et délivre un diagnostic global du système cible. Une seconde approche envisage de fractionner la couche de diagnostic en autant d'entités de diagnostic qu'il existe de sous-systèmes. Dans cette approche, une entité de diagnostic est affectée à chaque sous-système du système cible. Le diagnostic global du système cible est alors obtenu à partir des résultats produits par les entités locales de diagnostic. Inspiré des approches de la littérature pour diagnostiquer des systèmes répartis, nous proposons de distinguer trois classes d'architectures pour réaliser le diagnostic embarqué de systèmes cibles (figure 2.4). Le diagnostic peut être centralisé, décentralisé ou réparti (également nommé distribué : *distributed diagnosis*) (Fabre, 2007 ; Ferrari, 2009 ; Roychoudhury et al., 2009 ; Subias, 2006). Notons que le diagnostic décentralisé ou réparti est généralement associé à une méthode de diagnostic particulière dans la littérature (Genc et Lafortune, 2003 ; Pencolé et Cordier, 2005).

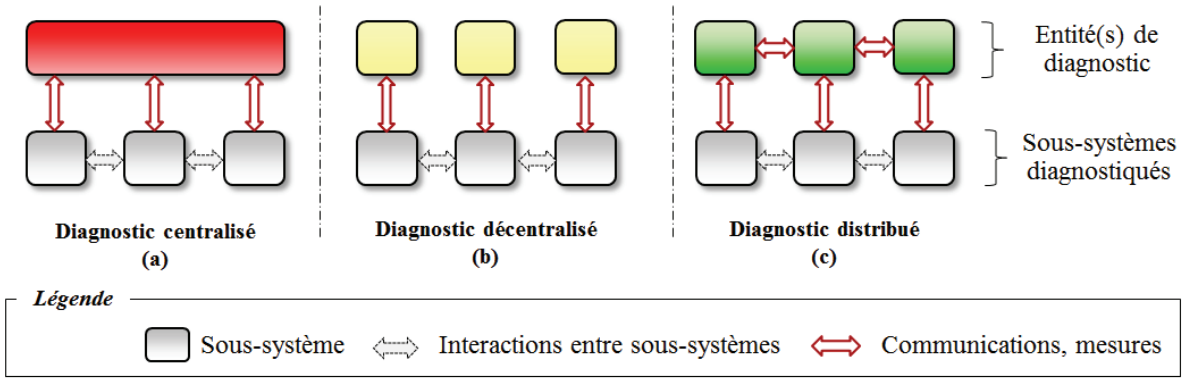

Figure 2.4. Diagnostic centralisé (a), décentralisé (b) et distribué (c), adaptée de (Ferrari, 2009).

Dans la première classe d'architecture, nommée Diagnostic Embarqué Centralisé (DEC), une seule entité de diagnostic est implémentée à bord du système cible et réalise le diagnostic des différents sous-systèmes. Dans une architecture de Diagnostic Embarqué Décentralisé (DED), une entité de diagnostic est implémentée pour chaque sous-système. Ces entités réalisent le diagnostic de manière indépendante et ne peuvent communiquer entre elles. Lorsque le diagnostic est distribué, chaque sous-système est également associé à une entité de diagnostic, mais ces entités peuvent communiquer entre elles. Le terme distribué fait généralement référence au medium de communication utilisé pour interconnecter ces entités de diagnostic. Dans les travaux présentés dans ce manuscrit, cette troisième classe d'architecture est nommée Diagnostic Embarqué Décentralisé et Coopératif (DEDC), afin de dissocier l'organisation de ces entités de diagnostic des solutions technologiques utilisées pour mettre en œuvre ce type d'architecture. Les sous-sections suivantes précisent ces trois catégories d'architectures.

1.1.1 Diagnostic embarqué centralisé

Le diagnostic embarqué d'un système cible est généralement réalisé par une entité centrale de diagnostic, en charge de traiter les données brutes des différents sous-systèmes (Dievart et al., 2010). L'architecture DEC suppose qu'une seule entité de diagnostic puisse intégrer une connaissance de l'ensemble des sous-systèmes et établir le diagnostic global du système cible (Kurien et al., 2002). Aussi, cette entité doit être capable d'absorber et de traiter de grandes quantités de données à bord du système cible. Cette architecture peut s'avérer irréaliste pour un système cible constitué de multiples sous-systèmes. En effet, les données relatives aux sous-systèmes doivent être centralisées en un même point à bord du système cible pour être analysées. Le système cible doit alors disposer de ressources suffisantes pour supporter ces flux de données et réaliser les traitements nécessaires (volume de données important à traiter en un temps donné, par exemple). Par ailleurs, cette architecture présente des limitations en termes de fiabilité et de modularité. Par exemple, l'ajout d'un sous-système entraine systématiquement une modification des traitements réalisés par cette entité centrale de diagnostic.

Un exemple d'implémentation d'une architecture DEC dans le domaine des transports aériens est le système de maintenance centralisé (CMS : *Central Maintenance System*). Dans cette architecture, une unité centrale effectue des traitements à partir de symptômes et autres données brutes pour élaborer un diagnostic (Balin et Stankunas, 2013). Ce système permet d'identifier les causes des anomalies de fonctionnement et élabore des recommandations pour

le personnel de bord. Les résultats du diagnostic sont archivés et peuvent également être consultés par des techniciens de maintenance.

Néanmoins, ce système central doit être suffisamment performant (puissance de calcul et taille mémoire notamment) pour fournir des informations dans des délais acceptables (situation d'urgence, par exemple).

1.1.2 Diagnostic embarqué décentralisé

L'architecture DED favorise le traitement des données au plus près des sous-systèmes à diagnostiquer. Dans cette architecture, chaque sous-système est diagnostiqué localement par une entité de diagnostic. Les résultats des diagnostics délivrés par ces différentes entités sont généralement transmis de manière disjointe au centre de maintenance. Le diagnostic global du système cible peut être envisagé de plusieurs manières. Il peut être considéré, par exemple, comme une juxtaposition des résultats des diagnostics locaux (Biteus, 2004). De ce fait, le diagnostic global correspond aux diagnostics des différentes parties du système cible, ce qui permet de présenter des informations plus compréhensibles aux opérateurs de maintenance. Aussi, le diagnostic global peut résulter d'une fusion des résultats de diagnostic locaux. Dans ce cas, une ou plusieurs entités supplémentaires sont chargées de collecter les résultats des diagnostics locaux et de les fusionner. Cependant, les entités de diagnostic opèrent de manière indépendante et exploitent uniquement les observations limitées de leurs sous-systèmes. En d'autres termes, ces entités de diagnostic ne communiquent pas entre elles (Qiu et Kumar, 2004). Par conséquent, une entité de diagnostic n'est pas en mesure de considérer les interactions entre les sous-systèmes lors de l'élaboration d'un diagnostic (Ferrari, 2009). Ainsi, une entité peut émettre un diagnostic erroné lorsque le sous-système auquel elle est associée est influencé par d'autres sous-systèmes.

Cette catégorie d'architecture a notamment été expérimentée pour réaliser le diagnostic à bord de véhicules routiers. Par exemple, dans (Nasri et al., 2012) les auteurs proposent une architecture DED limitée au diagnostic d'une fonction sécuritaire d'un véhicule. Cette architecture met en œuvre différentes entités locales de diagnostic (diagnostiqueurs) qui intègrent une méthode de diagnostic spécifique, et dont les résultats de diagnostic sont communiqués à une entité supérieure nommée coordinateur. Ce coordinateur est implémenté à bord du véhicule et délivre un diagnostic final en fusionnant les résultats des diagnostics locaux à partir de règles. Les entités de diagnostic mises en œuvre sont intégrées aux équipements embarqués existants faisant l'objet d'un diagnostic (unités de contrôle des sous-systèmes du véhicule). Les informations contextuelles relatives aux différents sous-systèmes diagnostiqués ne sont pas considérées par ces entités de diagnostic. Les résultats évoqués par les auteurs concernent l'influence des entités de diagnostic sur la fiabilité de l'architecture global du système et ne précisent pas réellement la performance du diagnostic obtenu.

1.1.3 Diagnostic embarqué décentralisé et coopératif

Dans cette catégorie d'architecture, les entités de diagnostic sont agencées de la même manière que dans une architecture DED, c'est-à-dire, une entité de diagnostic est affectée à chaque sous-système du système cible. A la différence d'une architecture DED, ces entités opèrent de manière proactive et peuvent dialoguer entre elles (Zhang et Zhang, 2013). Par exemple, une entité de diagnostic peut coopérer avec d'autres entités. Elle obtient ainsi des informations complémentaires pour enrichir ses observations locales et fournir un diagnostic plus robuste. Ces informations concernent, par exemple, les états et contextes d'autres sous-

systèmes (symptômes ou modes de fonctionnement de sous-systèmes évoluant dans le même environnement, par exemple). L'architecture DEDC permet d'établir un diagnostic global du système cible suivant un schéma de communication défini entre les différentes entités de diagnostic (Provan, 2002). Les échanges entre ces entités sont généralement calqués sur les interactions existantes entre les différents sous-systèmes diagnostiqués (interactions physiques et informationnelles, par exemple) (Fabre, 2007). Les communications entre entités limitent le nombre de fausses alarmes transmises aux opérateurs de maintenance. Néanmoins, cette architecture implique des flux d'informations supplémentaires à bord du système cible par rapport aux architectures DEC et DED. Cette contrainte est compensée par un allégement des capacités de calcul requises pour réaliser le diagnostic, et par une augmentation de la fiabilité et de la modularité de l'architecture (Ferrari, 2009). Par ailleurs, cette approche est adaptée à l'architecture des systèmes cibles, dans laquelle les sous-systèmes interagissent entre eux et présentent un dispositif de calcul local (unité de contrôle disposant de processeurs et de mémoires, par exemple) (Roychoudhury et al., 2009).

Les architectures DEDC développées pour la surveillance de systèmes mobiles sont relativement peu nombreuses. Par exemple, une architecture de type DEDC est proposée dans (Dievart et al., 2010) pour la surveillance d'un système de transport. Cette architecture est basée sur des entités de diagnostic reliées à une base de données commune et pouvant s'échanger des informations par le biais d'une mémoire partagée (tableau noir). Certaines entités sont autorisées à communiquer directement entre elles. Cependant, une méthode de diagnostic spécifique est exécutée par les différentes entités qui composent cette architecture. Par ailleurs, les informations contextuelles des sous-systèmes surveillés ne sont pas considérées pour améliorer la performance du diagnostic.

Plus récemment, certaines architectures DEDC ont été développées dans le cadre du programme IVHM (*Integrated Vehicle Health Management*), introduit à l'origine dans les secteurs aéronautique et spatial (Benedettini et al., 2009). Ce concept vise à accroitre la sécurité et la fiabilité de certains systèmes de transports (avions de ligne, véhicules aériens, par exemple) et à optimiser leurs coûts de maintenance, en implémentant à leur bord des fonctions avancées de diagnostic et de pronostic (Ferreiro et al., 2012 ; Schoeller et al., 2007). Avec la création d'un centre de Recherche en collaboration avec différents industriels (IVHM-Center, 2008), la communauté IVHM contribue activement à l'élaboration de nouveaux standards (SAE-HM1, 2013). Cette communauté s'appuie notamment sur la norme ISO 13374 pour concevoir et mettre en œuvre des modules de surveillance embarqués. Par exemple, le projet VIPR (*Vehicle Integrated Prognostic Reasoner*) a permis de définir un modèle d'architecture pour la surveillance de véhicules aérospatiaux (Bharadwaj et al., 2013). L'architecture VIPR combine différentes entités interagissant entre elles pour établir un diagnostic global d'un véhicule. Ces entités nommées HM (*Health Manager*) adhérent à la structure du véhicule à diagnostiquer et sont réparties suivant trois niveaux hiérarchiques (niveau équipement, niveau régional et niveau véhicule) (figure 2.5). Au niveau hiérarchique le plus bas, chaque entité élabore des informations locales de diagnostic et de pronostic (*DP*) relatives à un équipement (*LRU HM*). Ces informations sont ensuite confiées aux entités de niveau supérieur (*Area HM*), capables d'analyser ces résultats en considérant les différentes régions du véhicule (implantation géographique des équipements, interdépendances entre équipements, par exemple). Enfin, les conclusions établies par ces entités sont exploitées par une entité située au niveau supérieur (*Vehicle HM*) chargée de délivrer un diagnostic final du véhicule.

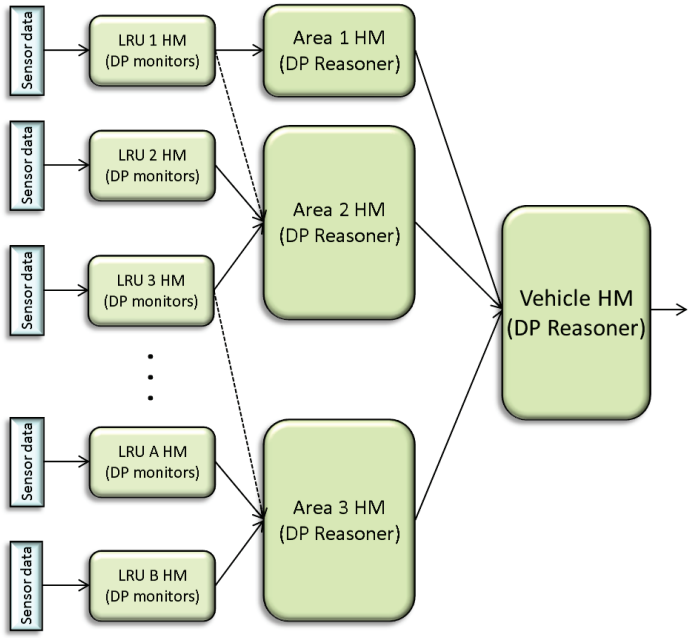

Figure 2.5. Architecture VIPR, adaptée de (Cornhill et al., 2013).

Cette architecture permet d'établir un diagnostic performant au niveau véhicule, dont le taux de fausses alarmes est faible (Cornhill et al., 2013). Néanmoins, ces entités sont généralement associées à des technologies spécifiques. De plus, le contexte opérationnel associé aux équipements n'est pas exploité par les entités qui composent cette architecture (informations relatives aux organes de contrôle des sous-systèmes, par exemple).

1.3 Typologie des architectures de diagnostic

A partir des architectures exposées précédemment, une typologie des architectures de diagnostic peut être articulée autour de trois axes.

- Le premier axe concerne l'implantation du diagnostic vis-à-vis du système cible à diagnostiquer. Par exemple, le diagnostic peut être exécuté à bord du système cible, à proximité du système cible (outil de diagnostic portatif, par exemple), ou être réalisé à distance dans un centre de maintenance (Alanen et al., 2006).

- Le deuxième axe précise la distribution de la couche diagnostic en fonction des différents sous-systèmes à diagnostiquer. Ainsi, une entité de diagnostic peut être affectée à l'ensemble des sous-systèmes (diagnostic centralisé), ou à chaque sous-système du système cible (diagnostic décentralisé). Il est également possible d'affecter plusieurs entités de diagnostic à un même sous-système. Certains auteurs mentionnent alors une distribution *spatiale* ou *sémantique* du diagnostic (Fröhlich et al., 1999 ; Roos et al., 2003). La distribution spatiale fait référence à une répartition géographique des entités de diagnostic, où chaque entité est affectée à une certaine partie du système cible et prend en charge le diagnostic de cette partie. Une partie peut correspondre, par exemple, à un sous-système ou à un regroupement de plusieurs sous-systèmes. Dans une distribution sémantique du diagnostic, des entités de diagnostic hétérogènes sont affectées à une même partie du système cible et réalisent

conjointement le diagnostic de cette partie. Une entité de diagnostic peut être focalisée sur certains aspects de cette partie ou être spécialisée dans la résolution de certains types de problèmes (analyse de défaillances particulières, par exemple).

- Le troisième axe de cette typologie détermine l'organisation des entités de diagnostic au niveau social. Suivant la structure organisationnelle des entités de diagnostic, une entité peut être autorisée ou non à coopérer avec d'autres entités. Par exemple, une entité peut interagir avec l'ensemble des entités ou avec un nombre restreint d'entités (Atlas et al., 2001).

La figure 2.6 ci-dessous illustre la typologie proposée et positionne les différentes classes d'architectures définies auparavant. Le deuxième et le troisième axe concernent uniquement les architectures permettant de réaliser le diagnostic à bord de systèmes cibles. En effet, les architectures implémentant le diagnostic à distance impliquent des délais significatifs dans l'analyse des données et dans l'exécution des opérations de maintenance (temps de transmission, de stockage et de traitement des données collectées). Par exemple, l'architecture RCD souffre d'un manque de robustesse et de précision (taux de fausses alarmes important, informations de diagnostic erronées, non pertinentes ou non contextualisées). Par conséquent, les architectures pour lesquelles le diagnostic est exécuté à distance ne seront pas retenues dans la suite de nos travaux.

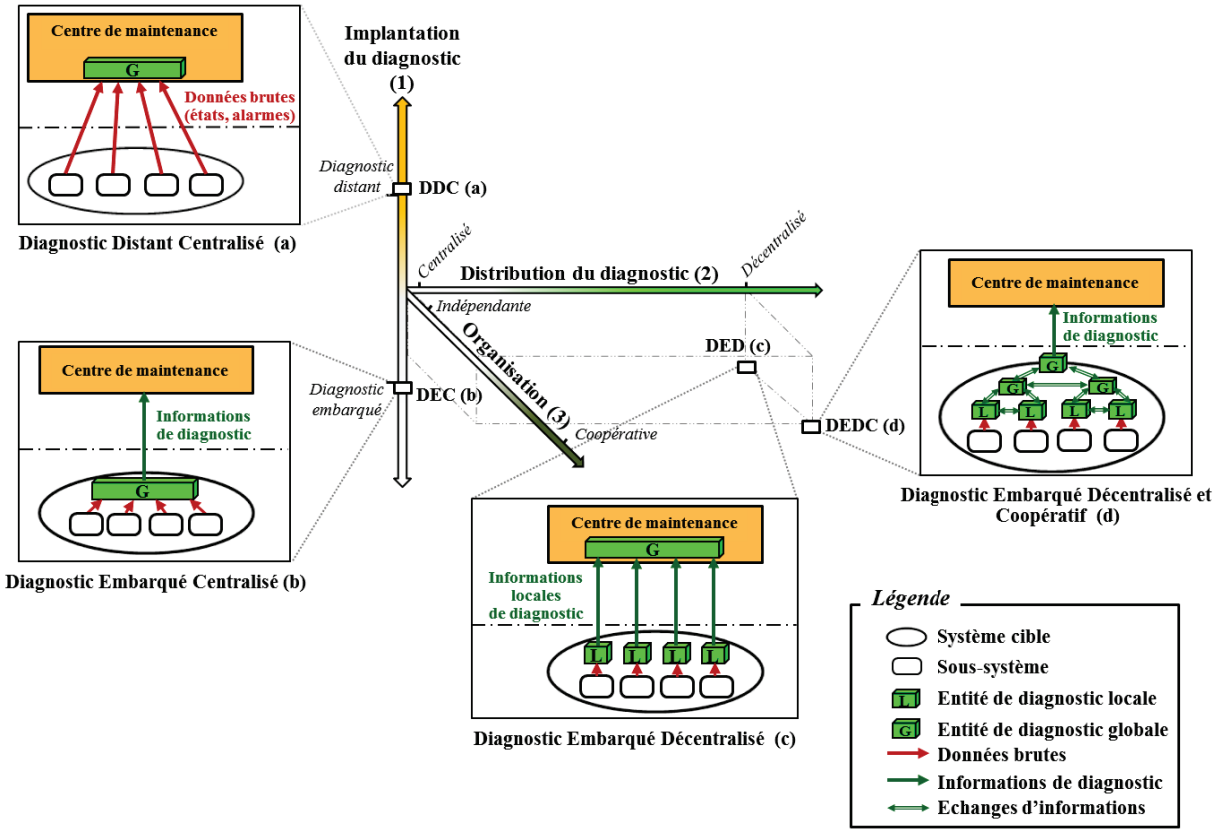

Figure 2.6. Positionnement des architectures de diagnostic suivant la typologie proposée.

Parmi les architectures de diagnostic embarqué, l'architecture DEDC s'avère être une approche pertinente pour élaborer le diagnostic de systèmes cibles mobiles. Cette architecture permet d'établir un diagnostic global du système cible à partir des informations délivrées par différentes entités de diagnostic implantées à bord du système cible et pouvant interagir entre

elles. En considérant les exigences d'un système de surveillance établies dans le précédent chapitre, cette architecture présente les avantages suivants :

- Précision : les données brutes sont traitées au plus près des équipements surveillés, ce qui permet de garantir la qualité des observations et par conséquent la précision du diagnostic.
- Adaptabilité : les changements survenant dans un sous-système du système cible (modification, ajout ou suppression de composants, par exemple) impactent uniquement l'entité de diagnostic qui lui est associée.
- Réactivité : les délais d'interventions de maintenance sont réduits car le système cible est capable de transmettre des informations de diagnostic directement exploitables par les services de maintenance en ciblant des actions de maintenance précises.
- Confiance : les informations contextuelles disponibles à bord du système cible peuvent être exploitées par les différentes entités de diagnostic. De plus, ces entités sont capables de communiquer entre elles pour obtenir davantage d'informations sur leur environnement et limiter ainsi le nombre de fausses alarmes.

Ces travaux se positionnent dans cette dernière catégorie d'architecture. Dans la suite de ce manuscrit, une architecture de type DEDC est retenue pour la surveillance de systèmes cibles mobiles. La partie suivante propose des approches permettant de modéliser les entités qui composent cette architecture en s'appuyant sur le concept de surveillance active.

2. Approches de surveillance active

L'architecture DEDC retenue précédemment requiert la définition de différentes entités en interaction dont l'objectif est de réaliser le diagnostic à bord d'un système cible. Ainsi, cette approche implique une augmentation des capacités informationnelles, décisionnelles et communicationnelles des sous-systèmes à diagnostiquer. Dans ce cadre, les travaux de Recherche menés au sein de notre équipe sur le développement de produits "actifs" ont permis de faire émerger la notion d'activité associée à un produit tout au long de son cycle de vie (Sallez, 2012). Cette notion d'activité s'oppose au caractère *passif* d'un produit classique, contraint de subir les interventions d'un système de soutien externe (système de distribution ou de maintenance, par exemple) et n'ayant pas la capacité de prendre des initiatives vis-à-vis de ce système. L'activité associée à un système cible (ou produit) est supportée par un *système d'augmentation*. Ce dernier réalise un ensemble de *fonctions d'augmentation* au plus près du système cible et lui confère, au minimum, la capacité de déclencher un événement. Cette faculté permet au système cible devenu "actif" d'exprimer un besoin en émettant un signal (demande d'une intervention ou d'un service particulier, par exemple). Bien entendu, cette activité minimale peut être complétée par d'autres activités plus complexes (mémorisation, apprentissage, coopération entre produits actifs, par exemple). Le concept d'*augmentation* consiste à associer un système d'augmentation à un produit passif lui permettant d'être "actif" pendant une ou plusieurs phases de son cycle de vie (phase de fabrication, de distribution et d'usage, par exemple) (Sallez et al., 2010).

Cette partie décrit le concept de surveillance active reposant sur le concept d'augmentation et suggère des approches permettant de modéliser l'activité associée à un système cible.

2.1 Concept de surveillance active

La surveillance active est une application du concept d'augmentation en phase d'usage qui consiste à doter un système cible de capacités d'auto-surveillance (Sallez et al., 2011). Ce concept implique une augmentation des sous-systèmes du système cible par des fonctions de surveillance. Ces fonctions sont supportées par différents systèmes d'augmentations situés au plus près des sous-systèmes surveillés. Le système cible constitué de sous-systèmes "actif" est donc lui-même "actif". Ce système cible "actif" peut alors délivrer des informations de surveillance de manière spontanée pendant son exploitation (état de santé, résultats de diagnostic, par exemple). Dans une perspective d'évolution du système cible (développement de nouvelles fonctionnalités, par exemple), les fonctions de surveillance supportées par un système d'augmentation pourront être prises en charge de façon native par le système cible.

Dans l'architecture DEDC retenue dans nos travaux, chaque sous-système réalise les couches fonctionnelles #1 à #4 du modèle ISO 13374. Un système d'augmentation peut alors être défini pour supporter les fonctionnalités associées à ces couches et élaborer le diagnostic d'un sous-système (figure 2.7). Ce système d'augmentation présente des capacités décisionnelles et interactionnelles lui permettant de dialoguer avec d'autres systèmes d'augmentation et de transmettre des informations de diagnostic à un centre de maintenance.

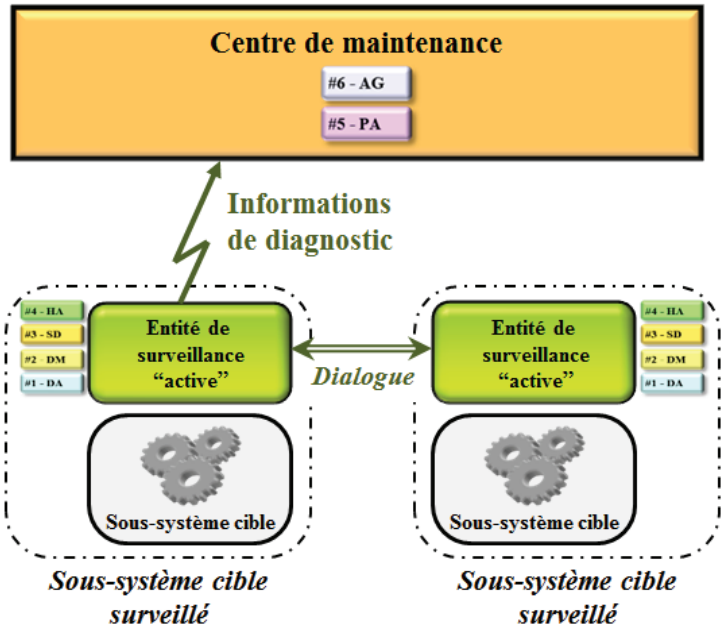

Figure 2.7. Exemple de surveillance active basée sur le concept d'augmentation.

Le concept de surveillance active est particulièrement adapté pour aborder la surveillance d'un système cible mobile suivant une architecture de type DEDC. En effet, chaque sous système du système cible peut être associé à une entité de surveillance "active" autonome, capable d'interagir avec d'autres entités. Inspiré des modèles d'architectures de systèmes de pilotage manufacturiers (Trentesaux, 2009), les sections suivantes présentent les principales approches permettant de modéliser ces entités de surveillance "actives" : l'approche multi-agents et l'approche holonique.

2.2 Approche multi-agents

L'intelligence artificielle distribuée (IAD), axe de l'intelligence artificielle (IA), s'intéresse aux domaines pour lesquels une seule entité est inadaptée ou inefficace pour résoudre certains types de problèmes. Cette communauté a introduit le concept de systèmes constitués d'entités autonomes ayant la capacité d'agir sur leur environnement et de communiquer avec d'autres entités pour atteindre un objectif donné. Un système constitué de ces entités ou *agents* est nommé Système Multi-Agents (SMA).

2.2.1 Notion d'agent

De nombreuses définitions du terme agent sont apparues depuis l'émergence des systèmes multi-agents. En effet, cette notion est utilisée dans de nombreux domaines (Tweedale et al., 2007). Bien qu'aucune définition précise ne soit acceptée, nous pouvons citer les deux définitions suivantes couramment rencontrées dans la littérature.

- Un agent est un système informatique, situé dans un environnement, et qui agit d'une façon autonome et flexible pour atteindre les objectifs (buts) pour lesquels il a été conçu (Wooldrige et Jennings, 1995).

- Un agent est une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle-même et sur son environnement, qui, dans un univers multi-agents, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents (Ferber, 1995).

Les agents sont décrits dans un système et diffèrent donc d'un système à un autre. La plupart des typologies d'agents établies dans la littérature sont axées sur le degré de raisonnement des agents. Un agent peut être vu comme un processus permettant de lier ses perceptions à ses actions. Une classification usuelle dissocie deux catégories d'agents définies ci-après : les agents réactifs et les agents cognitifs (Demazeau et Briot, 2001).

- Agents réactifs : ces agents sont des entités très simples, n'ayant quasiment pas de capacité de raisonnement et ne disposant pas de mémoire. Ils perçoivent leur environnement mais n'ont pas la capacité de le représenter symboliquement. Leurs capacités limitées répondent uniquement à des stimuli provenant de l'environnement. Les actions d'un agent sont directement liées à ses perceptions par une fonction reflexe (stimulation-exécution). Par conséquent, les agents réactifs adoptent des mécanismes de communication restreints. L'interaction entre ces agents et l'environnement fait émerger une organisation cohérente et intelligente au niveau d'un système d'agents. Ces agents sont qualifiés d'intelligents au niveau d'un groupe d'agents réactifs (Müller, 2000).

- Agents cognitifs : ces agents, au contraire, plus complexes, possèdent des capacités de raisonnement individuelles et une base de connaissance. Ces agents ont la capacité de percevoir et de constituer une représentation explicite de leur environnement (Ferber, 1995). Un agent cognitif peut être amené à raisonner suivant ses perceptions passées, ses interactions avec d'autres agents, ses actions antérieurement initiées, pour planifier au mieux ses actions futures. Ils ont potentiellement la capacité d'apprendre et de reconnaître certaines situations. Un agent cognitif adopte un langage de

communication plus élaboré (mécanismes d'envois de messages et requêtes, par exemple). Dans une organisation d'agents cognitifs, les agents accomplissent leur propre but en cherchant à optimiser leur utilité au sein du collectif d'agents. Jusqu'à présent, cette approche a donné lieu aux travaux les plus avancés.

Les agents réactifs et cognitifs sont généralement décrits comme deux entités antagonistes. Cependant, certains agents *hybrides* peuvent posséder des propriétés réactives et cognitives. A titre d'exemple, un agent hybride peut mener un raisonnement à l'aide de ses connaissances et présenter certaines aptitudes à réagir aux évènements.

2.2.2 Architectures à base d'agents

Un agent a généralement pour vocation d'agir au sein d'une organisation d'agents. Un système multi-agents (SMA) est un ensemble d'agents qui évoluent dans un même environnement. Un SMA peut être constitué, par exemple, d'agents purement réactifs, cognitifs ou hybrides. Dans une approche "voyelle" (Demazeau, 2001), on distingue quatre dimensions permettant d'élaborer un système multi-agents : l'Agent, l'Environnement, l'Interaction et l'Organisation (AEIO). L'organisation décrit les relations entre les agents du système suivant un modèle organisationnel (modèles bio-inspirés, par exemple). Dans un SMA, la communication est primordiale pour assurer les différentes interactions entre agents. Elle peut être assimilée à une forme d'action locale, par exemple, d'un agent vers un autre agent, ou d'un agent vers plusieurs agents (diffusion de messages au sein d'un groupe d'agents, par exemple).

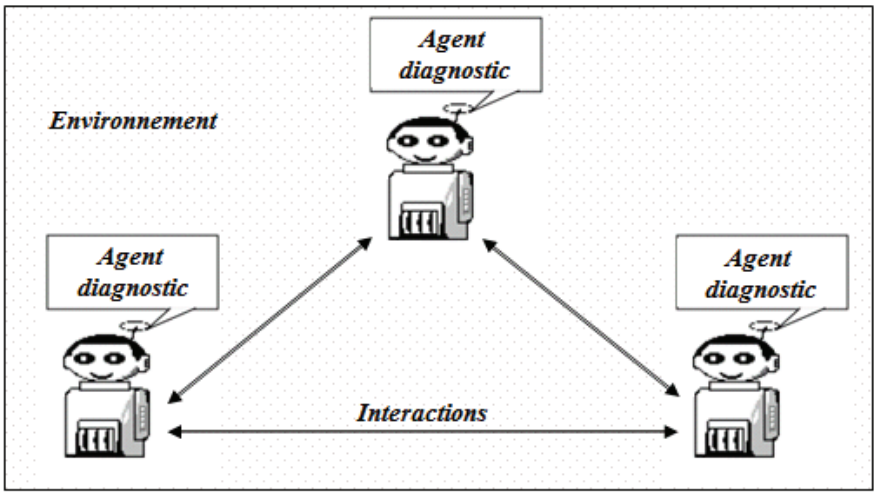

Figure 2.8. Exemple de SMA appliqué à la surveillance de systèmes, adaptée de (Ferber, 1995).

L'architecture interne d'un agent, c'est à dire, la décomposition modulaire utilisée pour relier les perceptions d'un agent à ses actions permet également de différencier les systèmes multi-agents. Les types d'architectures les plus connues sont les architectures suivantes (Grondin, 2008) :

- l'architecture horizontale monocouche : l'agent est constitué d'un seul module ayant accès à la perception et à l'action (figure 2.9a),
- l'architecture horizontale avec plusieurs couches : l'agent est constitué de plusieurs modules où chaque module a accès simultanément à la perception et à l'action (figure 2.9b),
- l'architecture verticale modulaire : l'agent est constitué de plusieurs modules, dont le premier accède à la perception et le dernier module accède à l'action (figure 2.9c),
- l'architecture verticale en couches : l'agent est constitué de plusieurs modules dont un seul module peut accéder à la perception et à l'action (figure 2.9d).

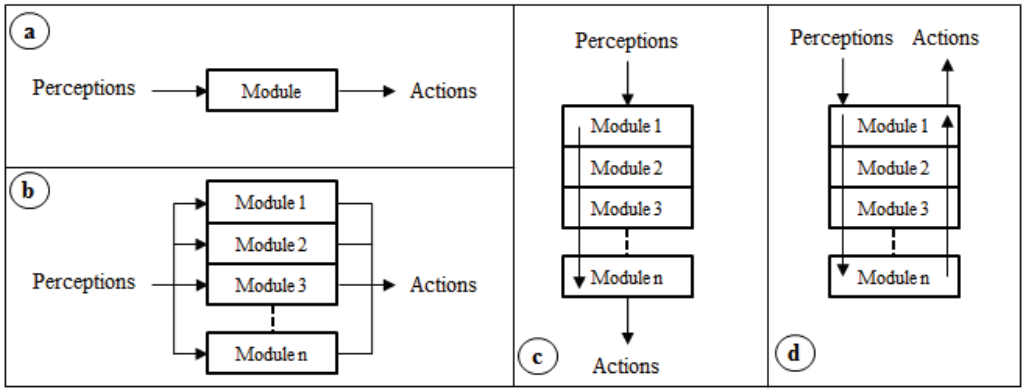

Figure 2.9. Exemples d'architectures internes d'un agent, adaptée de (Müller et al., 1995).

Les architectures horizontales sont majoritairement employées pour le développement d'agents réactifs, comme l'architecture de *subsumption* (Brooks, 1986), tandis que des architectures verticales sont plutôt associées à des agents cognitifs, telle que l'architecture BDI (*Belief Desire Intention*) (Rao et Georgeff, 1995). Un agent hybride peut présenter une architecture verticale ou horizontale. Par exemple, l'architecture *InteRRaP* est une architecture verticale modulaire d'agents hybrides qui comporte plusieurs couches fonctionnelles (Müller et al., 1995). L'approche multi-agents a notamment été exploitée pour la surveillance de systèmes répartis dans (Albert et al., 2001 ; Mangina et al., 2001 ; Naedele et al., 2004).

2.3 Approche holonique

Les systèmes holoniques ont été introduits par le philosophe hongrois Arthur Koestler (Koestler, 1967). Ces systèmes sont basés sur le concept de "holon", terme formé par le préfixe grec *Holos* (le tout) et du suffixe *on* qui suggère une particule élémentaire ou partie d'un tout. Une propriété importante de ces systèmes est la récursivité des entités mises en œuvre nommées holons. L'approche holonique peut s'appliquer à de nombreux systèmes, qui peuvent être naturels (systèmes biologiques) ou artificiels (systèmes conçus par l'homme).

2.3.1 Concept de holon

Un holon est décrit comme une entité autonome faisant partie d'un *tout* qui présente un caractère individuel et coopératif pour atteindre un objectif donné (Koestler, 1967). Le concept de holon repose sur la notion de récursivité (Suaréz et al., 2013). Un holon peut être vu comme une partie constitutive d'un holon de niveau supérieur, et comme un tout, composé lui-même de holons de niveaux inférieurs. Cette dualité associée au holon est appelée *effet Janus* par Koestler, en référence au nom de la divinité romaine regardant dans deux directions opposées. Ainsi, les deux principales propriétés d'un un holon sont l'autonomie et la coopération. La première confère au holon une identité propre d'un ensemble autonome et cohérent. La seconde représente l'aspect social d'un holon, contribuant partiellement au fonctionnement global du système suivant des mécanismes de coopérations avec d'autres holons.

2.3.2 Architectures holoniques

Un système de holons qui peuvent coopérer pour atteindre un but ou un objectif donné est nommé *holarchie*. Comme l'illustre la figure 2.10, une holarchie est formée par des holons récursifs organisés de manière hiérarchisée. Naturellement, un holon peut être composé d'autres holons, être lui-même considéré comme une holarchie et également appartenir à une holarchie plus vaste. Dans une holarchie, un holon peut interagir avec d'autres holons situés au niveau inférieur ou supérieur (interactions "verticales") et avec des holons situés au même niveau que lui (interactions "horizontales" représentées sur la figure 2.10) dans un espace de coopération.

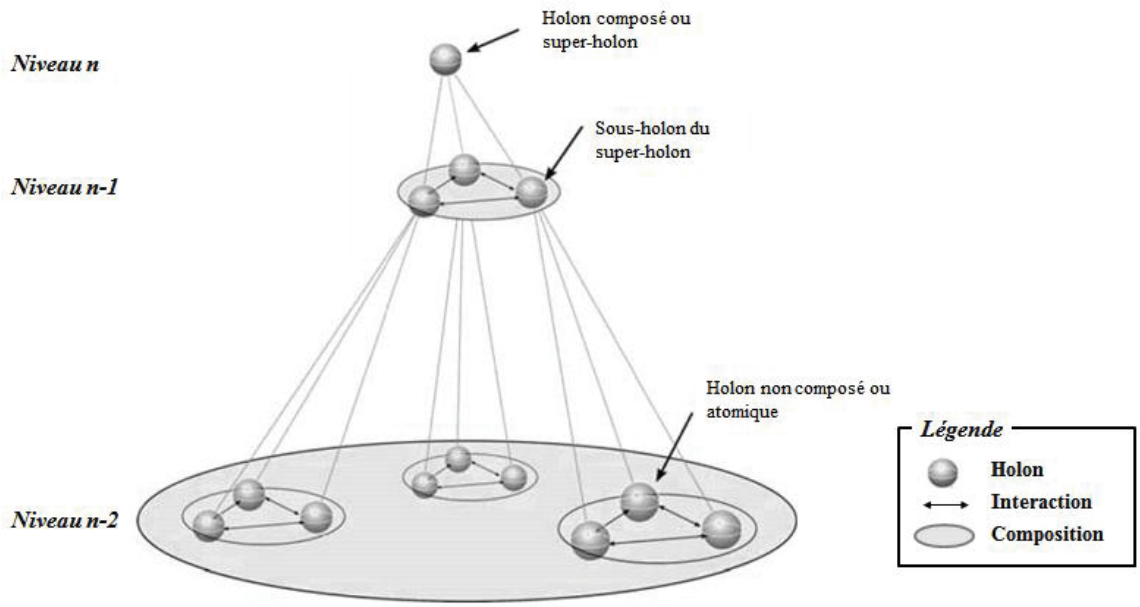

Figure 2.10. Exemple de système de holons ou holarchie, adaptée de (Gaud, 2007).

L'holarchie définit les règles de base pour la coopération entre holons et de ce fait limite leur autonomie (Tanaya et al., 1997). Les conflits résultants d'une coopération entre holons au sein d'un même espace de coopération peuvent être résolus par un holon de niveau supérieur. Ce holon émet alors une décision à partir des observations de ses holons subordonnés.

L'approche holonique a notamment été appliquée au pilotage de systèmes manufacturiers dans le cadre du projet HMS (*Holonic Manufacturing System)* entrepris par la communauté internationale IMS (*Intelligent Manufacturing System*) (IMS, 2013). Dans ce domaine, l'approche holonique a permis de concevoir de nouveaux modèles d'architectures, reposant sur une décentralisation des capacités de traitement et de décision à travers des entités autonomes et communicantes (Blanc et al., 2008 ; Deen, 2003). Les travaux importants proposés par cette communauté représentent une rupture avec les architectures centralisées de pilotage de systèmes manufacturiers et se proposent de répondre aux nouveaux besoins et contraintes des industriels (flexibilité, personnalisation de masse des produits, traçabilité des produits, par exemple) (Pujo et al., 2009).

Un HMS est une holarchie dans laquelle les holons sont autonomes, coopératifs et entretiennent des liens de subordinations avec des holons hiérarchiquement supérieurs (commandes, instructions, par exemple). Suivant les modèles d'architectures holoniques proposés (ou holarchies), différents types de holons ont été définis pour désigner des entités d'un système manufacturier. Les architectures PROSA (Van Brussel et al., 1998) et ADACOR (Leitão et Restivo, 2006) proposent par exemple d'associer des holons aux machines de production (holon ressource, holon machine), aux produits manufacturés (holon produit), aux opérations à effectuer (holon ordre, holon tâche), ou encore aux tâches de supervision (holon staff, holon supervision). La figure 2.11 présente une architecture générique de holon adoptée par cette communauté. Dans ce modèle, un holon comporte une partie "traitement de l'information" et une partie "traitement physique" (optionnelle). La partie "traitement de l'information" est subdivisée en trois modules : une interface de communication nécessaire pour interagir avec les autres holons (interface inter-holons), une intelligence décisionnelle pour élaborer un raisonnement interne et disposer d'une certaine autonomie (centre de décision), et une interface pour interagir avec des opérateurs humains (interface homme-holon).

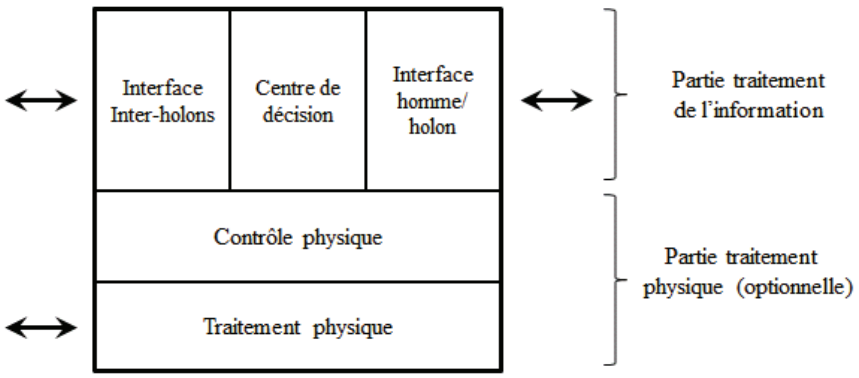

Figure 2.11. Architecture interne d'un holon, adaptée de (Bussman, 1998).

La partie "traitement physique" se compose d'un module de commande (contrôle physique) et d'un module opérant (traitement physique). Cette partie est décrite comme facultative, dans la mesure où un holon peut être constitué uniquement d'une partie "traitement de l'information".

2.4 Comparaison des approches

Les communautés holoniques et agents convergent vers des spécifications communes, car ces deux approches reposent sur les mêmes principes fondamentaux d'autonomie et de coopération en explorant la distribution et la décentralisation des entités et des fonctions (Leitão, 2004). Un holon possède des propriétés similaires à celles d'un agent, mais se distingue de par sa récursivité (Giret et Botti, 2004). En effet, dans la plupart des modèles d'agents, un agent est considéré comme une entité atomique (Gaud, 2007). Néanmoins, certaines approches considèrent que des agents peuvent se regrouper pour former un agent de niveau supérieur. Suivant ce concept, un système multi-agents holonique (*holomas*) est un SMA ayant une structure récursive. Les agents d'un certain niveau sont regroupés en organisations qui sont considérées comme des entités individuelles à un niveau supérieur. La communauté agent s'intéresse aux comportements des entités qui composent un SMA (agent réactif ou cognitif, par exemple) et au comportement global émergent d'un SMA. Dans une approche holonique, un holon est récursif et associé à une partie matérielle (tangible), qui n'est pas abordée dans les modèles d'agents. L'approche holonique est donc complémentaire à l'approche multi-agents. D'un point de vue conceptuel, le holon est un concept et l'agent est à la fois un concept et une technologie associée. Un modèle à base d'agents est approprié pour implémenter la modularité et la décentralisation des structures holoniques (Leitão, 2004 ; Maríik et al., 2002). Ainsi, la technologie agent peut fournir les outils nécessaires pour mettre en œuvre les traitements informationnels associés aux holons tout comme les mécanismes d'interactions entre ces entités.

Dans le cadre de nos travaux, une approche à base d'agents peut être exploitée pour modéliser les entités de surveillance de l'architecture DEDC. Comme l'illustre l'exemple présenté sur la figure 2.12, une entité de surveillance "active" peut être décrite par un agent. Sur ce même exemple, en considérant une approche holonique, un holon permet de considérer à la fois la partie physique surveillée d'un système cible et l'entité de surveillance "active" qui lui est associée. Cette seconde approche permet également de décrire la récursivité du système cible.

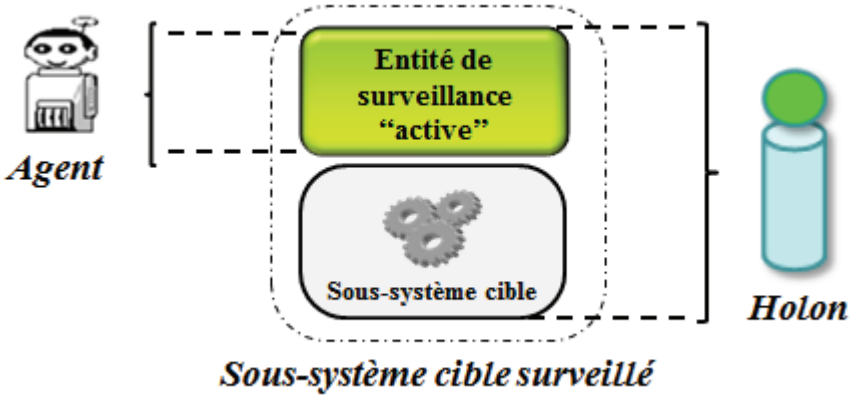

Figure 2.12. Périmètres de modélisation considérés dans le cadre de notre étude.

Une approche holonique est retenue dans la suite de nos travaux pour modéliser les sous-systèmes surveillés d'un système cible mobile suivant une architecture DEDC. Cette approche qui considère les aspects matériels et immatériels des systèmes a notamment été expérimentée pour la surveillance de systèmes naturels (Louati et al., 2012 ; Unland, 2003) et artificiels (Jarvis et Jarvis, 2003 ; McFarlane et al., 1995 ; Silva et al., 2012 ; Stecca et al., 2013).

3. Conclusion

Ce chapitre a dressé tout d'abord un état de l'art des architectures de diagnostic pour la surveillance de systèmes cibles mobiles. La surveillance d'un tel système nécessite une prise en charge de fonctions de surveillance par le système cible mobile et par un centre de maintenance distant. La première partie de ce chapitre a défini quatre classes d'architectures de diagnostic et a proposé une typologie des architectures de diagnostic. Afin de répondre aux exigences exprimées dans le précédent chapitre, une architecture de diagnostic dans laquelle le diagnostic est embarqué et réalisé de manière décentralisée et coopérative a été retenue pour la surveillance de systèmes cibles mobiles (architecture DEDC). Dans la littérature, les applications relatives à ce type d'architecture sont généralement spécifiques à une méthode de diagnostic et manquent de généricité. Par ailleurs, les informations contextuelles disponibles à bord d'un système cible ne sont pas suffisamment valorisées et considérées pour améliorer la performance du diagnostic embarqué. En s'appuyant sur le concept de surveillance active, la seconde partie de ce chapitre a présenté deux approches pertinentes pour modéliser les entités qui composent cette architecture : les systèmes multi-agents et les systèmes holoniques. Une étude comparative de ces deux approches a favorisée l'adoption d'une approche holonique pour la surveillance de système cibles mobiles.

Le chapitre suivant propose une architecture générique de surveillance "active" de type DEDC reposant sur une approche holonique. Cette architecture représente la composante surveillance d'un système de maintenance associé à un système cible.

Chapitre III

Proposition d'une architecture holonique pour la surveillance active de systèmes cibles mobiles

Introduction

Dans le précédent chapitre, une étude comparative des architectures de diagnostic pour la surveillance de systèmes cibles mobiles a été présentée. Cette analyse a permis de retenir une catégorie d'architecture dans laquelle le diagnostic est implanté à bord du système cible et élaboré de manière décentralisée et coopérative (architecture DEDC).

Ce chapitre est consacré à la proposition d'une architecture de type DEDC pour la surveillance de systèmes cibles mobiles. L'approche proposée s'appuie sur le concept de surveillance active décrit dans le précédent chapitre en adoptant une approche holonique. Dans ce chapitre, une architecture de surveillance active basée sur des entités de surveillance autonomes et coopératives est proposée. La première partie de ce chapitre présente tout d'abord une approche de modélisation d'un système cible à surveiller. Ce modèle permettra de caractériser les entités de surveillance qui lui sont associées et de proposer par la suite notre architecture holonique de surveillance. La deuxième partie se concentre sur la structure interne des entités qui composent cette architecture holonique de surveillance. Elle détaille notamment les fonctionnalités supportées par ces entités.

1. Système de surveillance holonique

L'adoption d'une architecture de type DEDC pour la surveillance de systèmes cibles mobiles nécessite une identification des sous-systèmes à surveiller qui composent un système cible. Un partitionnement du système cible en sous-systèmes est alors nécessaire avant de définir les entités de surveillance qui concourent à l'élaboration d'un diagnostic. Cette décomposition peut être abordée par une approche analytique, qui consiste à décomposer un problème en différents sous-problèmes indépendants. Cependant, cette approche connait certaines limitations, notamment en raison du niveau de complexité croissant des systèmes actuels et à la manière dont ces systèmes sont conçus (Fabre, 2007).

Une approche complémentaire, se focalisant non seulement sur une analyse séparée des sous-systèmes d'un système cible, mais considérant également un système cible dans son ensemble s'avère nécessaire. L'approche systémique adoptée au début de ce manuscrit (*cf.* chapitre I, section 1.1) peut ainsi remédier aux difficultés de compréhension d'un système cible. Elle permet de conserver une vision globale d'un système cible, constitué de sous-systèmes en interaction, eux-mêmes considérés comme des systèmes.

Dans cette partie, la décomposition d'un système cibles en sous-systèmes est abordée en adoptant une approche systémique. Après un rappel des principaux concepts de la systémique, la section 1.1 présente un modèle de système cible à surveiller basé sur cette approche. La section 1.2 spécifie les entités de surveillance associées à ce modèle suivant une approche holonique. La section 1.3 propose notre architecture holonique de surveillance basée sur ces entités de surveillance.

1.1 Approche de modélisation du système cible à surveiller

La systémique a pour objet l'étude et la compréhension de systèmes dont l'analyse par une approche analytique s'avère difficile ou impossible, et vise à établir une représentation du système étudié par le biais de modèles qualitatifs (Donnadieu et al., 2003). Elle s'appuie notamment sur les interactions entre les éléments constituant un système pour en comprendre son comportement global. Ces liens portent aussi bien sur de simples relations de causes à effets que sur des échanges d'informations, de flux de matière ou d'énergie. Ces interactions s'opèrent à des niveaux hiérarchiques différents, formant plusieurs niveaux organisationnels structurés de manière hiérarchisée. La systémique met en œuvre trois points de vue nécessaires à l'analyse et à la représentation d'un système (*c.f.* annexe A). Cette triangulation systémique regroupe les visions fonctionnelle, organique et historique du système étudié dans le but d'approfondir sa compréhension (Le Moigne, 1994). Outre ces différents axes d'analyse d'un système, la démarche systémique préconise une identification des sous-systèmes (sous-ensembles ou composants) d'un système, afin de mettre en évidence les relations que ces entités entretiennent entre elles et leurs rôles au sein d'un système. Contrairement à une approche analytique, ces entités ne coïncident pas nécessairement avec des composants élémentaires d'un système tangible à analyser. Aussi, cette décomposition soulève la question du partitionnement d'un système en sous-systèmes défini par le modélisateur. Bien que subjectif, l'établissement de frontières (ou périmètres) des sous-systèmes peut être dicté par certains critères, notamment à partir des différents points de vue d'un système abordés précédemment. La dimension fonctionnelle, s'intéressant aux fonctions assurées par le système, et la dimension organique, basée sur la structure des éléments constituant le système et leurs répétitivités, sont deux exemples de critères permettant de décomposer un système cible.

La systémique proposant un cadre méthodologique d'analyse et de conception de systèmes cibles, conformément à un fonctionnement désiré ou à une finalité du système, elle peut également être exploitée pour comprendre les dysfonctionnements survenant dans de tels systèmes (Belhadaoui, 2011). Cette section propose une modélisation d'un système cible basée sur une approche systémique. Ce modèle de système cible à surveiller servira par la suite de socle à l'élaboration d'une architecture de surveillance.

Le modèle proposé repose sur les hypothèses établies dans le premier chapitre et plus précisément :

- décomposition : le système cible à surveiller est supposé être organisé suivant une structure hiérarchisée de contrôle et décomposable structurellement en un ensemble fini de sous-systèmes. Le partitionnement du système cible en sous-systèmes est supposé être établi sans chevauchements entre les frontières des sous-systèmes. Chaque sous-système est supposé être contrôlé par un système de niveau supérieur,

- hétérogénéité technologique : le système cible et ses sous-systèmes sont supposés être conçus avec des technologies hétérogènes. Néanmoins, le système cible comme chaque sous-système présente une structure interne identique. Il se compose d'une partie contrôle (partie décisionnelle) et d'une partie sous-contrôle (partie opérante),

- environnement : les parties contrôle et sous-contrôle sont supposées interagir dans un environnement spécifique. Un contexte opérationnel de nature physique (température, champ électrique, par exemple) ou informationnelle (mode de fonctionnement, identifiant de l'utilisateur du système, par exemple) est supposé être une propriété intrinsèque du système cible comme de chaque sous-système constituant ce système.

La figure 3.1 présente un modèle organique du système cible considéré. Ce modèle est inspiré de l'architecture de systèmes constitués d'une partie contrôle et d'une partie sous-contrôle (Isermann, 2008). En accord avec nos hypothèses, ce modèle présente les propriétés suivantes :

- une partie contrôle : entité de traitement de l'information et de commande, en charge d'assurer un comportement cohérent de la partie sous-contrôle, conformément aux spécifications fonctionnelles imposées par le concepteur,
- une partie sous-contrôle : ensemble d'éléments soumis aux commandes de la partie contrôle, comprenant notamment des éléments matériels (capteurs, transducteurs, par exemple) ou informationnels (estimateurs, observateurs logiciels, par exemple) rendant compte de l'état de cette partie,
- des signaux de commandes (U) : ensemble d'ordres délivrés par la partie contrôle dans le but de commander la partie sous-contrôle,
- des signaux de mesures (Y) : ensemble de signaux fournis par des dispositifs de la partie sous-contrôle et exploités par la partie contrôle pour l'élaboration des signaux de commandes,
- un contexte opérationnel (C) : ensemble d'informations relatif à l'environnement du système et/ou au système lui-même caractérisant une situation de fonctionnement.

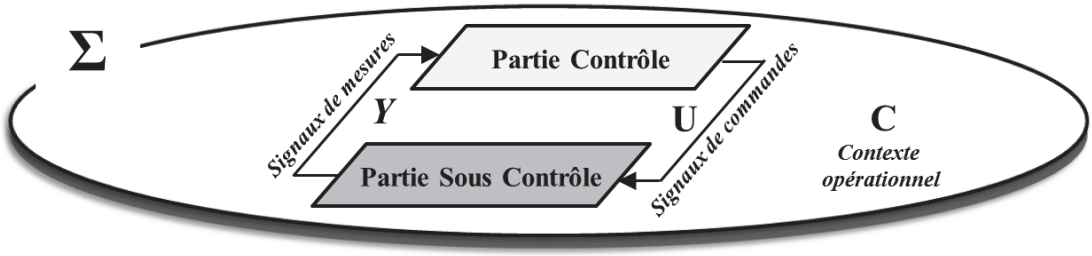

Figure 3.1. Modèle organique du système cible considéré.

Le système cible à surveiller, représenté suivant une vision systémique et noté Σ sur la figure 3.1, comporte deux parties en interaction opérant dans un contexte opérationnel. Ce modèle organique correspond à une vue globale du système cible considéré, au plus haut niveau de sa structure hiérarchique. De par ses propriétés récursives, ce système englobe un ensemble fini de sous-systèmes de niveaux inférieurs ayant une structure interne identique. Les sous-systèmes sont organisés en niveaux et contrôlés par un système de niveau supérieur. Dans le modèle proposé, chaque système est caractérisé par la position qu'il occupe dans la composition d'un système plus vaste. Par la suite, le sous-système j du système Σ_i sera noté $\Sigma_{i/j}$, et le $k^{\text{ème}}$ sous-système du sous-système $\Sigma_{i/j}$ sera noté $\Sigma_{i/j/k}$. La figure 3.2 ci-dessous propose un modèle organique arborescent du système cible représenté sur la figure 3.1. Suivant ce modèle, le système global, noté Σ_i sur la figure 3.2, est décomposé en une succession de sous-systèmes situés à différents niveaux. Chaque sous-système est décomposé en sous-systèmes de niveau inférieur, eux-mêmes décomposés en sous-systèmes.

Par exemple sur la figure 3.2, le système Σ_i de niveau n est décomposé en trois sous-systèmes ($\Sigma_{i/j}$, $\Sigma_{i/j'}$ et $\Sigma_{i/j''}$) de niveau *n-1*, et le sous-système $\Sigma_{i/j}$ est lui-même décomposé en deux sous-systèmes de niveau *n-2* ($\Sigma_{i/j/k}$ et $\Sigma_{i/j/k'}$). Le sous-système $\Sigma_{i/j/k'}$ est décomposé de nouveau en sous-systèmes élémentaires de niveau inférieur ($\Sigma_{i/j/k'/l}$ et $\Sigma_{i/j/k'/l'}$).

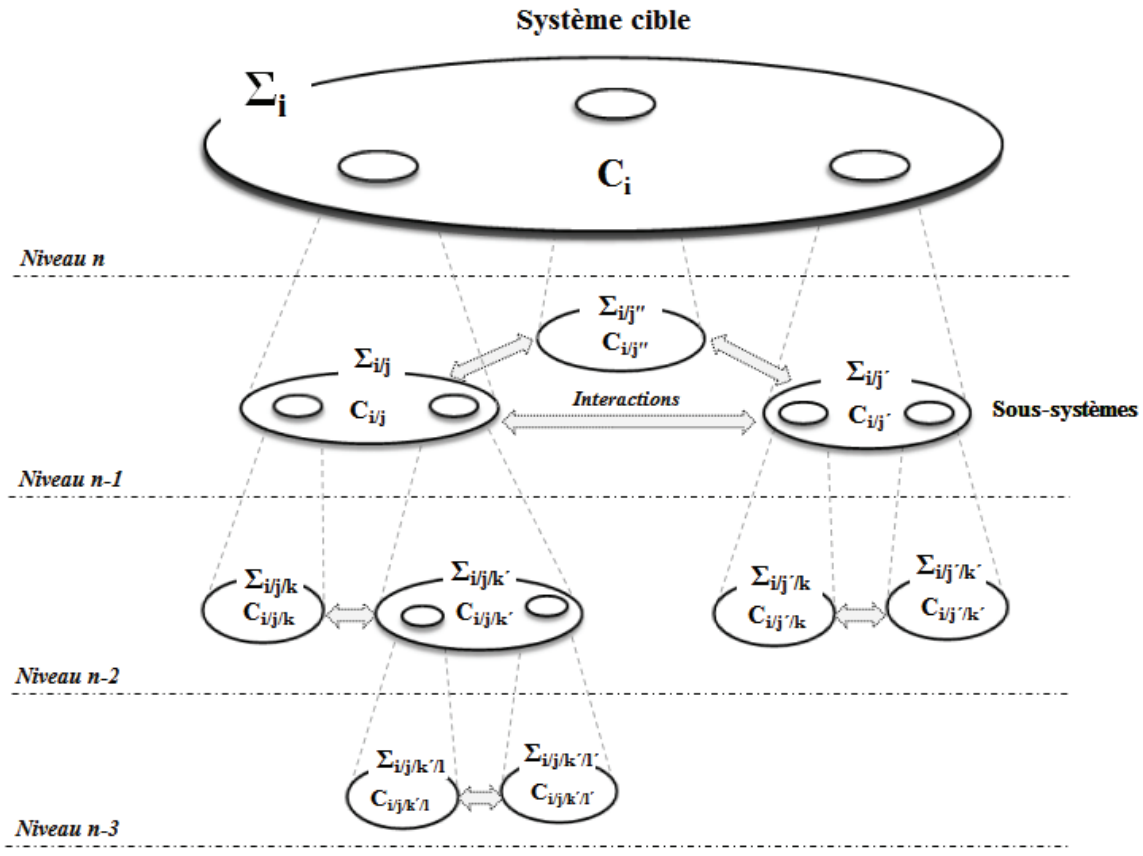

Figure 3.2. Modèle organique arborescent du système cible considéré.

Comme évoqué précédemment chaque système est immergé dans un contexte opérationnel. Le contexte du système Σ_i et de son sous-système $\Sigma_{i/j}$ seront par la suite notés respectivement C_i et $C_{i/j}$. Une propriété importante de l'approche systémique proposée est l'agrégation des informations de contexte entre les différents sous-systèmes constituant un système cible. En effet, le contexte opérationnel d'un sous-système caractérise la situation de fonctionnement de ce sous-système et également de ses sous-systèmes de niveaux inférieurs. Par ailleurs, les sous-systèmes sont soumis à un même contexte relatif au niveau hiérarchique le plus élevé. Par exemple, sur le modèle de la figure 3.2, les sous-systèmes $\Sigma_{i/j/k'/l}$ et $\Sigma_{i/j/k'/l'}$ opèrent tous deux dans le contexte $C_{i/j/k'}$, relatif au sous-système $\Sigma_{i/j/k'}$, et également dans $C_{i/j}$, contexte commun aux sous-systèmes $\Sigma_{i/j/k}$ et $\Sigma_{i/j/k'}$. Les différents sous-systèmes du système Σ_i sont soumis au même contexte C_i.

La structure hiérarchisée de contrôle du système cible considéré implique nécessairement des interactions "verticales" de nature informationnelle entre les différents niveaux de décomposition du système cible (relations d'autorités entre un sous-système et son sous-système de niveau inférieur, par exemple). De plus, des interactions "horizontales" de nature informationnelle et physique sont envisageables au sein d'un même niveau (échanges d'informations ou influences physiques entre sous-systèmes, par exemple).

La section suivante définit une approche holonique pour la surveillance de systèmes cibles décrits par ce modèle.

1.2 Spécification d'une architecture holonique de surveillance

L'approche holonique abordée dans le précédent chapitre (*cf.* chapitre II, section 2.1) permet de préciser et de caractériser la vision systémique du système cible à surveiller. En effet, le système cible considéré est organisé en niveaux et présente une structure hiérarchisée de contrôle. Comme l'illustre la figure 3.3 ci-dessous, la partie sous-contrôle d'un système Σ_i est composée d'un ensemble de sous-systèmes, constitués eux-mêmes de sous-systèmes. En considérant l'architecture interne d'un holon détaillée dans le chapitre précédent, chaque sous-système s'apparente à une partie "traitement physique" d'un holon, comprenant un module de contrôle physique (partie contrôle) et un module de traitement physique (partie sous-contrôle). Lorsqu'une partie sous-contrôle regroupe plusieurs parties "traitement physique", ce holon est alors lui-même composé d'autres holons. Par ailleurs, chaque sous-système peut être vu comme un tout, considéré comme un système autonome, et également comme une partie constitutive d'un tout, c'est-à-dire comme un organe appartenant à un système de plus grande dimension formant un ensemble cohérent.

La figure 3.3 décrit également, en conformité avec les principes holoniques, les interactions "verticales" inter-niveaux, représentant les liens de subordinations entre les sous-systèmes, et des interactions "horizontales" intra-niveaux se produisant dans le domaine informationnel (I_{inf}) comme dans le domaine physique (I_{phy}). Par exemple sur la figure 3.3, les sous-systèmes $\Sigma_{i/j/k}$ et $\Sigma_{i/j/k'}$ sont contrôlés par le sous-système $\Sigma_{i/j}$, et les sous-systèmes $\Sigma_{i/j}$ et $\Sigma_{i/j'}$ sont contrôlés par le système Σ_i.

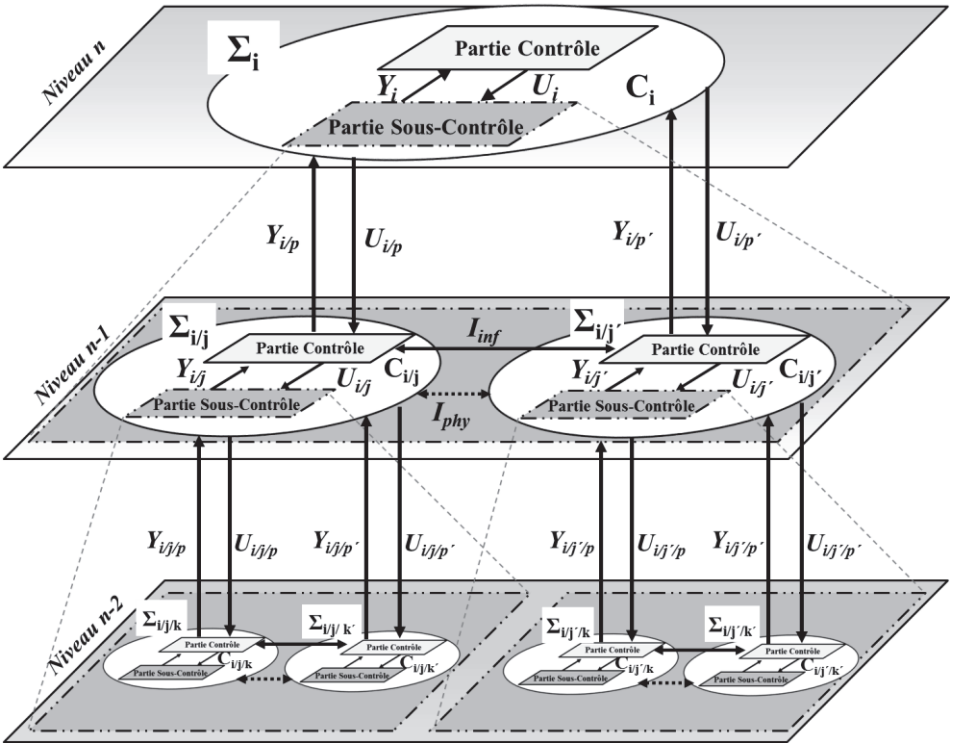

Figure 3.3. Structure hiérarchisée de contrôle du système cible considéré.

Suivant ce concept de holon associant une partie informationnelle à un système physique, des capacités d'auto-surveillance peuvent alors résider dans la partie informationnelle d'un holon, dans le but de diagnostiquer la partie physique de ce holon. De ce fait, un holon prend en charge la surveillance de sa partie "traitement physique", désignée par la suite *système sous surveillance*.

Les propriétés d'un holon de surveillance et la fonction de surveillance associée à cette entité sont définies dans les sous-sections suivantes.

1.2.1 Définition d'un holon de surveillance

Un holon de surveillance possède une autonomie décisionnelle lui permettant d'établir un diagnostic de manière autonome, à partir de ses observations et de ses connaissances du système sous surveillance, et de manière coopérative, suivant des mécanismes d'interactions avec d'autres holons. Par conséquent, un holon de surveillance est caractérisé par deux principales propriétés :

- Autonomie : un holon élabore le diagnostic du système sous surveillance auquel il est associé de manière autonome grâce à ses capacités d'auto-surveillance.
- Coopération : un holon peut interagir avec d'autres holons du système de surveillance dans le but d'affiner son diagnostic.

Un holon associé à un système sous surveillance représente à la fois une partie d'un tout, formée par des holons de niveaux supérieurs, et un tout, composé d'autres holons de niveaux inférieurs. Par conséquent, un holon associé à un système décomposable en sous-systèmes englobe des holons de niveaux inférieurs assurant la surveillance des sous-systèmes de ce système.

Un modèle générique de holon de surveillance est proposé ci-dessous sous forme de diagramme de classes UML (figure 3.4). Un holon de surveillance résulte de l'association d'un système sous surveillance et d'une fonction de surveillance. Ce modèle fait apparaitre une relation structurelle entre un système sous surveillance et une fonction de surveillance, dont l'association forme un holon de surveillance, et une relation de composition, exprimant la récursivité d'un système sous surveillance (un système peut être composé de systèmes), et par conséquent, la récursivité d'une fonction de surveillance et du holon de surveillance formé par cette association.

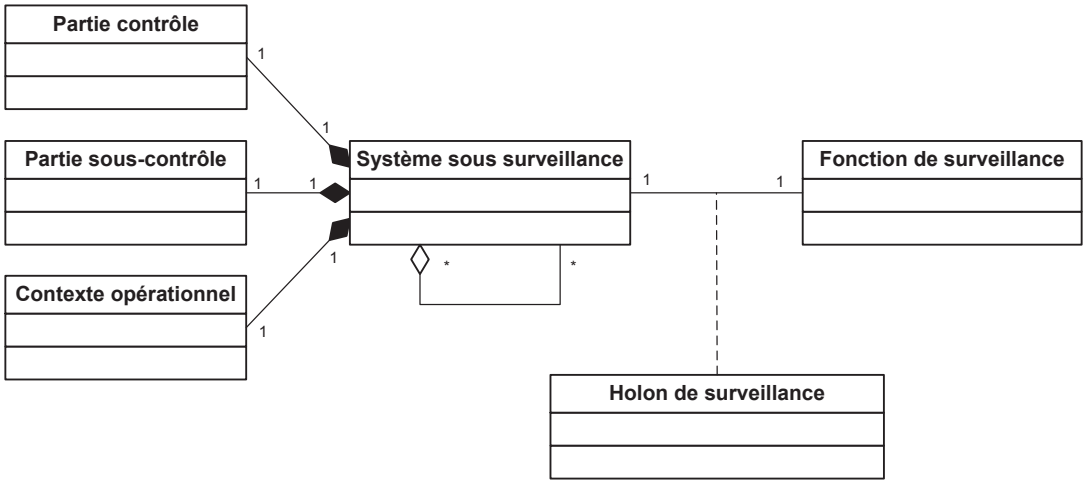

Figure 3.4. Diagramme de classes UML d'un holon de surveillance.

Un système sous surveillance est associé à trois relations de composition et comporte une partie contrôle, une partie sous-contrôle et un contexte opérationnel. La fonction de surveillance, présentée dans la sous-section suivante, observe en permanence le comportement du système sous surveillance et élabore un diagnostic de ce système. Conformément aux exigences d'un système de surveillance considérées dans nos travaux, un holon de surveillance doit être situé au plus près du système sous surveillance, ce qui suppose une implémentation embarquée de cette entité. Par ailleurs, la fonction de surveillance d'un holon est non intrusive vis-à-vis du système sous surveillance et n'interfère en aucun cas avec le fonctionnement de ce système, qu'il soit perçu comme normal ou anormal par cette fonction.

1.2.2 Définition d'une fonction de surveillance d'un holon

La fonction de surveillance d'un holon vise à observer le comportement du système sous surveillance, à détecter des anomalies de fonctionnement survenant dans ce système (symptômes), et à déterminer les causes probables des dysfonctionnements observés en incriminant des éléments potentiellement défectueux. Pour établir le diagnostic d'un système sous surveillance, une fonction de surveillance observe le comportement de la partie sous-contrôle de ce système. De plus, cette fonction exploite le contexte opérationnel du système sous surveillance dans le but d'accroître la robustesse du diagnostic. Dans les travaux présentés dans ce manuscrit, une fonction de surveillance n'a pas vocation à diagnostiquer la partie contrôle d'un système sous surveillance.

Une fonction de surveillance, notée f_i, constitue la partie informationnelle du holon de surveillance H_i associé au système sous surveillance Σ_i. Ainsi, un holon de surveillance H_i est formé par le couple (Σ_i, f_i). La fonction de surveillance du holon H_i élabore un diagnostic du système Σ_i, noté D_i (figure 3.5).

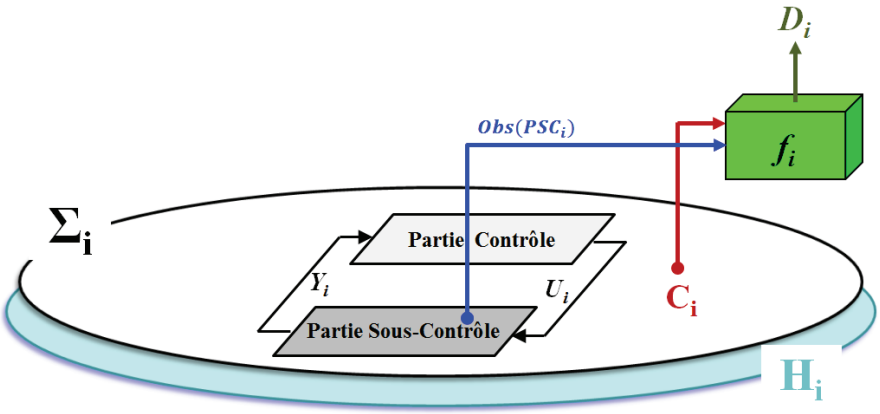

Figure 3.5. Système sous surveillance et fonction de surveillance d'un holon.

Le comportement de la partie sous-contrôle du système Σ_i est examiné par la fonction de surveillance à partir de données brutes, notées $Obs(PSC_i)$, recueillies sur cette partie. Une fonction f_i a donc pour rôle d'analyser ces données et de les convertir en informations pertinentes de diagnostic. Le diagnostic D_i élaboré par la fonction f_i est établi suivant les observations datées $Obs(PSC_i)$ de la partie sous-contrôle ainsi que des informations de contexte C_i du système Σ_i pendant une fenêtre temporelle notée T_{diag} (équation 3.1).

$$D_i = f_i\ (Obs(PSC_i), C_i, T_{diag}) \qquad (3.1)$$

Il est à noter que le holon de surveillance H_i représenté sur la figure 3.5 est associé à un système physique Σ_i considéré comme élémentaire et isolé d'un système cible plus vaste. Cependant, lorsque ce système est un élément constitutif d'un système cible et décomposable en sous-systèmes, il est contrôlé par un sous-système de niveau supérieur et englobe des sous-systèmes de niveaux inférieurs. Dans ce cas, un holon de surveillance associée à ce système peut être amené à interagir avec des holons du système de surveillance situés au même niveau, à un niveau inférieur ou à un niveau supérieur.

La section suivante caractérise les différentes interactions entre holons et propose une architecture holonique basée sur des holons de surveillance.

1.3 Architecture de surveillance proposée

Cette section présente l'architecture de surveillance proposée permettant de réaliser le diagnostic embarqué de systèmes cibles mobiles suivant une approche DEDC. L'architecture de surveillance proposée, nommée EMH (*Embedded Monitoring Holarchy*), est constituée de holons de surveillance en interaction. Ces holons forment une holarchie, dont l'objectif est de diagnostiquer un système cible mobile.

Notre contribution est double, en proposant, d'une part, une surveillance récursive, résultant d'une propagation successive d'informations de diagnostic à travers les niveaux hiérarchiques d'un système cible, basée sur des interactions "verticales", et d'autre part, une surveillance coopérative, reposant sur des interactions "horizontales" entre holons au sein d'un même niveau. Avant de présenter l'architecture de surveillance EMH, ces deux aspects sont présentés respectivement dans les sous-sections suivantes.

1.3.1 Surveillance récursive

Dès lors qu'un holon de surveillance est associé à un sous-système décomposable de nouveau en sous-systèmes, il est composé de holons de niveaux inférieurs associés aux sous-systèmes du sous-système auquel il est associé. Autrement dit, un holon de surveillance est associé à chaque sous-système d'un système cible, que ce sous-système soit décomposable ou élémentaire. Par conséquent, un holon composé d'autres holons se doit de collecter les résultats de diagnostic élaborés par les holons situés immédiatement au niveau inférieur (figure 3.6).

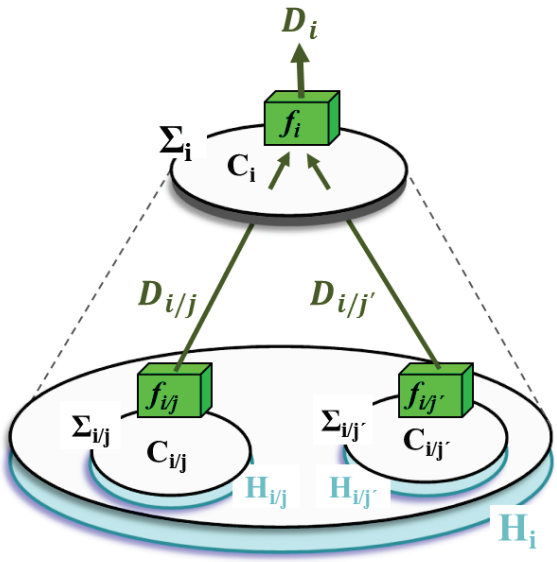

Figure 3.6. Récursivité d'un holon de surveillance.

Deux cas sont à considérer concernant un holon composé d'autres holons :

- Le holon élabore un diagnostic d'un système sous surveillance auquel il est associé grâce à ses capacités d'auto-surveillance et ses observations.
- Le holon reçoit des informations de diagnostic des autres holons situés immédiatement au niveau inférieur.

Par exemple, sur la figure 3.6, le système Σ_i est décomposé en deux sous-systèmes élémentaires ($\Sigma_{i/j}$ et $\Sigma_{i/j'}$). Le holon de surveillance H_i est associé au système Σ_i et les holons $H_{i/j}$ et $H_{i/j'}$ sont associés respectivement aux sous-systèmes $\Sigma_{i/j}$ et $\Sigma_{i/j'}$. La fonction de surveillance f_i élabore un diagnostic D_i et les fonctions de surveillance $f_{i/j}$ et $f_{i/j'}$ élaborent respectivement les diagnostics $D_{i/j}$ et $D_{i/j'}$. Le diagnostic D_i repose sur les observations du système Σ_i et sur les diagnostics $D_{i/j}$ et $D_{i/j'}$ des sous-systèmes de niveau inférieur. Les fonctions élémentaires de surveillance $f_{i/j}$ et $f_{i/j'}$ élaborent un diagnostic conformément à l'équation 3.1 présentée précédemment. La fonction f_i du holon composé H_i élabore un diagnostic D_i exprimé par l'équation 3.2 ci-dessous. Par souci de clarté, le terme D_i^* désigne les diagnostics émis par les holons de niveau inférieur collectés par la fonction f_i, soit les diagnostics $D_{i/j}$ et $D_{i/j'}$ sur la figure 3.6.

$$D_i = f_i\ (Obs(PSC_i), C_i, D_i^*, T_{diag}) \tag{3.2}$$

Le système Σ_i étant décomposable en un ensemble fini de sous-systèmes, le terme D_i^* est donné par l'équation 3.3 ci-dessous.

$$D_i^* = \{D_{i/j}, j \in [1; card(\Sigma_i)], j \in N\} \tag{3.3}$$

Cette équation traduit l'ensemble des diagnostics générés récursivement par les holons associés aux différents sous-systèmes du système Σ_i. Le terme $card(\Sigma_i)$ désigne le nombre total de sous-systèmes composant le système Σ_i.

Suivant ce principe de surveillance récursive, le diagnostic global d'un système cible sous surveillance est élaboré par le holon situé au niveau hiérarchique le plus élevé.

1.3.2 Surveillance coopérative

En complément d'une diffusion ascendante d'informations de diagnostic élaborées récursivement à travers les niveaux hiérarchiques d'un système cible, les échanges d'informations entre holons au sein d'un même niveau s'avèrent également pertinents pour accroitre la robustesse du diagnostic. En effet, les systèmes sous surveillance interagissent avec leur environnement et peuvent s'influencer mutuellement. L'environnement représente l'ensemble des entités externes à un système sous surveillance ayant potentiellement des interactions avec ce système.

Des interactions de nature informationnelle et physique peuvent ainsi exister entre un système et différentes entités situées dans son environnement (systèmes adjacents, utilisateurs du système, sources énergétiques, par exemple). Chaque interaction est caractérisée par sa nature et son sens (figure 3.7).

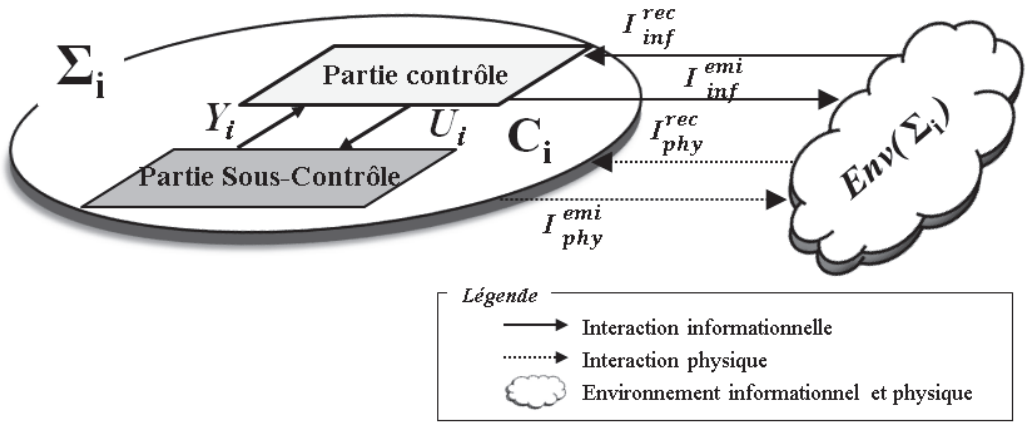

Figure 3.7. Exemples d'interactions entre un système et son environnement.

Une interaction est notée I_n^s où *n* représente la nature de l'interaction (nature informationnelle ou physique) et *s* désigne son sens par rapport au système sous surveillance (sens émission ou réception). Une interaction dans le sens réception indique un flux d'information entrant consommé par ce système dans le domaine informationnelle, ou une influence physique exercée par son environnement. Par ailleurs, une interaction dans le sens émission représente un flux d'information produit par ce système dans le domaine informationnel ou un phénomène physique généré par ce système.

Ces interactions, pour la plupart essentielles au bon fonctionnement du système et définies lors de sa conception, sont également le siège de dysfonctionnements difficiles à diagnostiquer, liés notamment aux interférences entre un système et son environnement. L'occurrence d'une faute primaire au niveau d'un sous-système engendrant de multiples fautes secondaires au niveau des sous-systèmes avec lesquels il interagit en est un exemple. Ce phénomène de propagation de fautes entre sous-systèmes peut ainsi être canalisé grâce aux échanges d'informations entre holons de même niveau. Ces échanges permettent de limiter les erreurs de diagnostics et évitent de propager des fausses alarmes. Par exemple, un équipement affecté d'une faute peut provoquer le dysfonctionnement d'un autre équipement en lui transmettant des informations erronées (interactions informationnelles). Dans ce cas, les

holons associés à ces équipements peuvent confronter mutuellement leurs observations afin d'élaborer un seul diagnostic pertinent.

L'échange d'informations entre holons permet à chaque holon d'enrichir ses informations de contexte grâce aux informations détenues par d'autres holons de même niveau. Ces informations complémentaires, non perceptibles localement par un holon, peuvent justifier des dysfonctionnements provoqués par l'environnement du système sous surveillance. Par exemple, un holon peut exploiter certaines grandeurs physiques ou disposer d'informations de géolocalisation permettant à d'autres holons de rattacher leurs observations à des phénomènes physiques externes.

La figure 3.8 illustre la surveillance de deux sous-systèmes élémentaires ($\Sigma_{i/j}$ et $\Sigma_{i/j'}$) en interaction dans le domaine physique et dans le domaine informationnel. Les holons $H_{i/j}$ et $H_{i/j'}$ associés à ces sous-systèmes coopèrent entre eux, en autorisant leurs fonctions respectives de surveillance $f_{i/j}$ et $f_{i/j'}$ à dialoguer entre elles. Le vocable "coopération", employé selon (Bussman, 1998), désigne l'ensemble des activités communicationnelles entre holons de même niveau. Ces activités ponctuelles peuvent influencer les résultats de diagnostic de chaque holon.

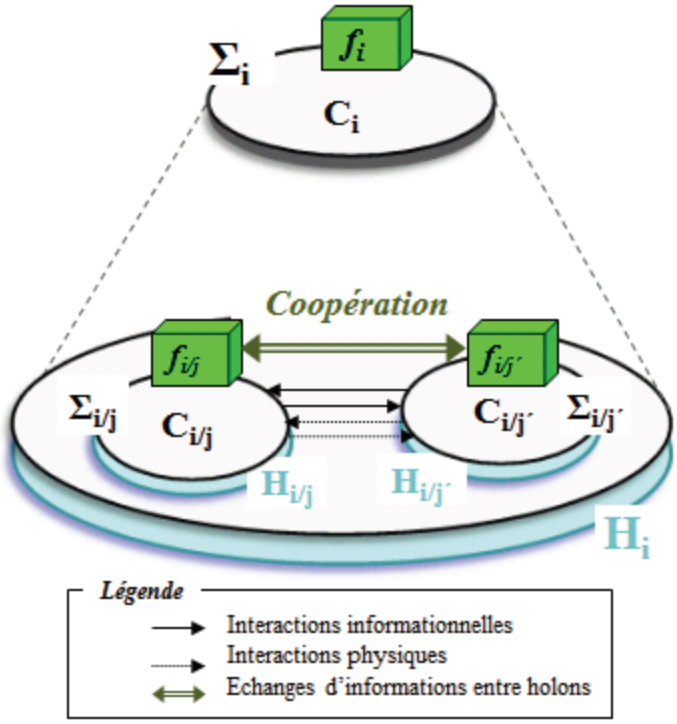

Figure 3.8. Coopération entre holons de surveillance.

La coopération entre holons de même niveau garantit la robustesse du diagnostic élaboré à chaque niveau avant la diffusion des résultats de diagnostic aux holons de niveau supérieur. Chaque holon peut considérer le contexte opérationnel et l'état des autres sous-systèmes situés au même niveau que lui avant de propager un diagnostic (analyse des informations de contexte des sous-systèmes situés à proximité ou des symptômes détectés par d'autres holons, par exemple). Les activités communicationnelles intra-niveaux supportent aussi bien des échanges élémentaires d'informations entre holons que des stratégies de diagnostic plus avancées, comme l'élaboration d'une décision collective de diagnostic, par exemple.

1.3.3 Architecture holonique de surveillance

L'architecture holonique de surveillance proposée, nommée EMH, repose sur la structure arborescente du système cible à surveiller considéré au début de ce chapitre (figure 3.3). Suivant une vision systémique, le système cible à surveiller est décomposé récursivement en sous-systèmes de niveaux inférieurs ayant la même structure. L'architecture de surveillance EMH proposée est illustrée par la figure 3.9. Un holon de surveillance est affecté à chaque sous-système du système cible à surveiller. Chaque holon hérite ainsi du niveau hiérarchique du sous-système auquel il est associé. De plus, un holon de surveillance est associé au système cible dans sa globalité. Celui-ci est situé au sommet de la structure holonique. Par conséquent, la structure holonique de surveillance reflète la structure arborescente du système cible à surveiller obtenue par décomposition systémique.

La fonction de surveillance d'un holon analyse le comportement d'un système physique et exploite son contexte pour élaborer un diagnostic.

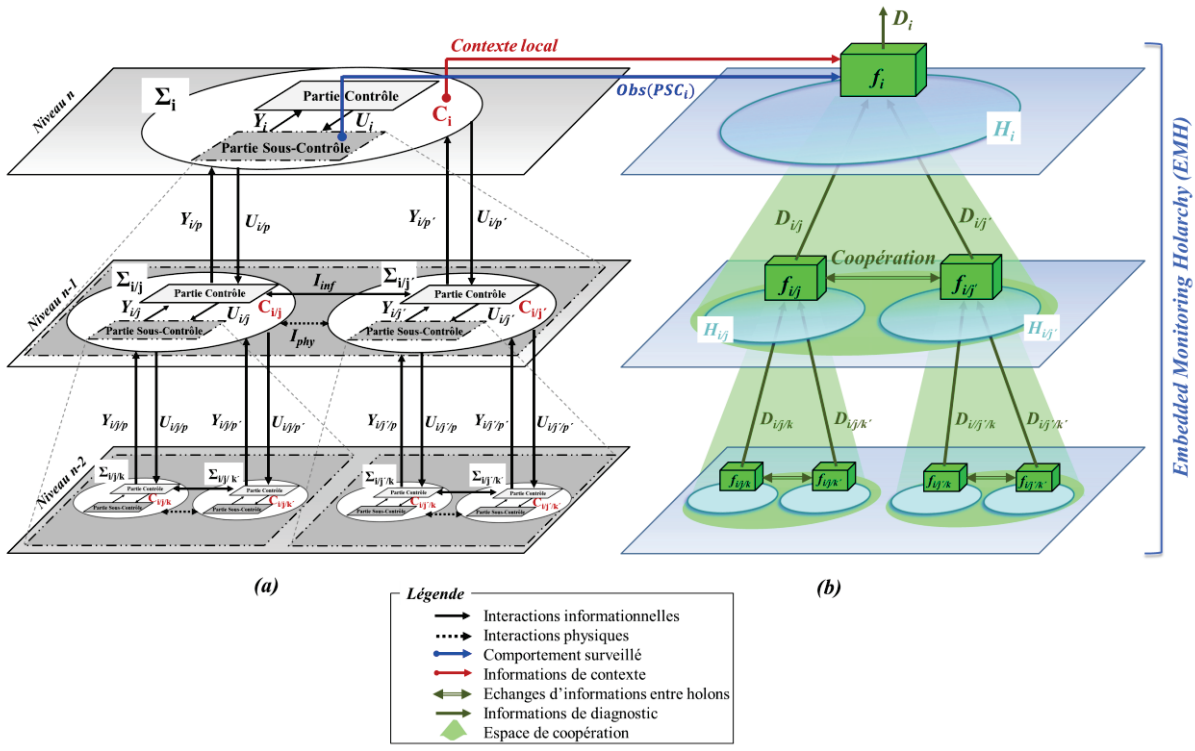

Figure 3.9. Système cible sous surveillance (a) et architecture de surveillance EMH (b).

Chaque holon de surveillance interagit avec les holons situés au niveau inférieur, dans le but de collecter les diagnostics élaborés récursivement par les holons qu'il regroupe, et interagit avec un holon hiérarchiquement supérieur, afin de propager ses résultats de diagnostic vers le niveau supérieur. De plus, chaque holon peut coopérer avec des holons situés au même niveau pour affiner ses résultats de diagnostic. Les interactions inter-niveaux et intra-niveaux initiés par les holons de surveillance ont lieu dans des espaces de coopération. Un espace de coopération véhicule l'ensemble des échanges entre holons d'un même groupe pour élaborer le diagnostic d'un système sous surveillance.

Suivant cette approche, chaque holon élabore un diagnostic précis à son niveau, et participe ainsi à l'élaboration d'un diagnostic robuste au niveau du système cible dans sa globalité. Dans l'architecture EMH proposée (figure 3.9), le diagnostic global du système cible (D_i) est établi par le holon situé au plus haut niveau de la structure holonique (H_i). Ce diagnostic

repose sur les fautes détectées localement par sa fonction de surveillance (f_i) et sur les diagnostics récursifs émis par les holons de niveaux inférieurs.

Dans l'approche proposée, le contexte du système est considéré à chaque niveau de l'architecture holonique, de telle sorte que chaque fonction de surveillance puisse interpréter des diagnostics élaborés localement, comme des diagnostics des sous-systèmes de niveaux inférieurs, suivant le contexte opérationnel local du système et suivant un contexte enrichi, établi grâce aux informations fournies par des holons situés au même niveau.

La partie suivante spécifie la structure interne des holons d'une architecture de surveillance EMH.

2. Structure interne d'un holon de surveillance

Après avoir proposé une architecture holonique de surveillance, reposant sur des holons de surveillance autonomes et coopératifs, cette partie se focalise sur la structure interne d'un holon de surveillance. Cette partie est organisée de la manière suivante. La section 2.1 présente les données consommées par une fonction de surveillance d'un holon pour élaborer un diagnostic. La section 2.2 précise les hypothèses considérées dans nos travaux lorsqu'une fonction élabore un diagnostic. La section 2.3 détaille les modules internes d'une fonction de surveillance d'un holon. Enfin, la section 2.4 spécifie les niveaux de diagnostic internes à une fonction de surveillance.

2.1 Données exploitées par la fonction de surveillance d'un holon

Comme abordé dans la partie précédente, la fonction de surveillance d'un holon observe le comportement de la partie sous-contrôle d'un système sous surveillance pour élaborer un diagnostic. Pour ce faire, une fonction de surveillance exploite en permanence les données d'entrée et de sortie de cette partie (figure 3.10).

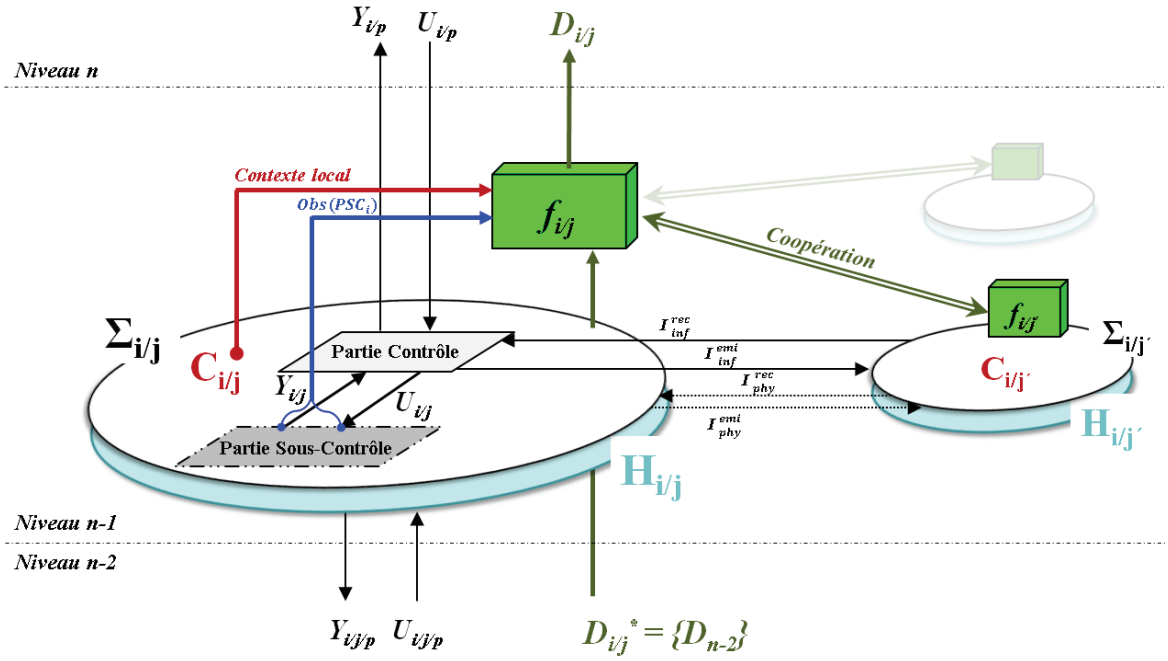

Figure 3.10. Données exploitées par une fonction de surveillance.

Les données permettant d'analyser le comportement de la partie sous-contrôle d'un système sous surveillance ($\Sigma_{i/j}$) sont les signaux de commandes délivrés par la partie contrôle ($U_{i/j}$) et de mesures fournis par la partie sous-contrôle ($Y_{i/j}$). De plus, chaque fonction de surveillance exploite le contexte local du système sous surveillance ($C_{i/j}$) et des informations élaborées par des fonctions de surveillance de même niveau. Par exemple, les informations contextuelles locales représentent les signaux consommés et délivrés par la partie contrôle, c'est à dire les signaux de commandes émanant du niveau supérieur ($U_{i/p}$) et de mesures transmis au niveau supérieur ($Y_{i/p}$), ainsi que les différentes interactions présentes entre le système et son environnement converties dans le domaine informationnel (I_{inf}^{rec}, I_{inf}^{emi}, I_{phy}^{rec}, I_{phy}^{emi}). Enfin, une fonction de surveillance réceptionne des informations de diagnostic fournies par les holons du niveau inférieur ($D_{i/j}{}^{*}$).

2.2 Hypothèses pour l'élaboration d'un diagnostic

Le diagnostic d'un système sous surveillance élaboré par une fonction de surveillance repose sur un ensemble d'hypothèses précisées ci-après :

- Fiabilité des observations : les dispositifs permettant de recueillir les données brutes nécessaires à l'élaboration d'un diagnostic sont supposés être non défaillants. De plus, ces données sont supposées être synchronisées par une même référence temporelle. Cette référence temporelle est supposée être fiable.
- Fiabilité des informations de contexte : le contexte opérationnel du système exploité par la fonction de surveillance d'un holon est supposé être fiable.
- Fiabilité de la fonction de surveillance : une fonction de surveillance est supposée être intrinsèquement fiable. Les traitements associés à cette fonction sont supposés être réalisés sans erreurs.
- Fiabilité des échanges entre holons : l'espace de coopération véhiculant l'ensemble des informations échangées entre holons est supposé être fiable, écartant toutes pertes d'informations ou erreurs de transmissions lors des échanges entre holons.
- Délais de transmissions : les informations détenues par un holon sont supposées être accessibles à un autre holon dès lors qu'il en fait la demande. Le délai de transmission entre holons est un aspect important, mais ne s'inscrit pas directement dans le cadre de nos travaux. Ainsi, les temps de communications et la quantité d'informations échangées entre holons sont supposés sans incidence sur le diagnostic.
- Panne unique du système sous surveillance : un seul mode de défaillance est supposé être présent lors de l'élaboration d'un diagnostic par une fonction de surveillance.

2.3 Modules internes d'une fonction de surveillance

La fonction de surveillance d'un holon repose sur les couches #1 à #4 du modèle prescrit par la norme ISO 13374. Dans le modèle proposé, le contexte opérationnel local du système est considéré au niveau de la quatrième couche pour élaborer le diagnostic du système sous surveillance. Le modèle proposé étant générique, chaque module peut implémenter différents algorithmes de traitement (figure 3.11).

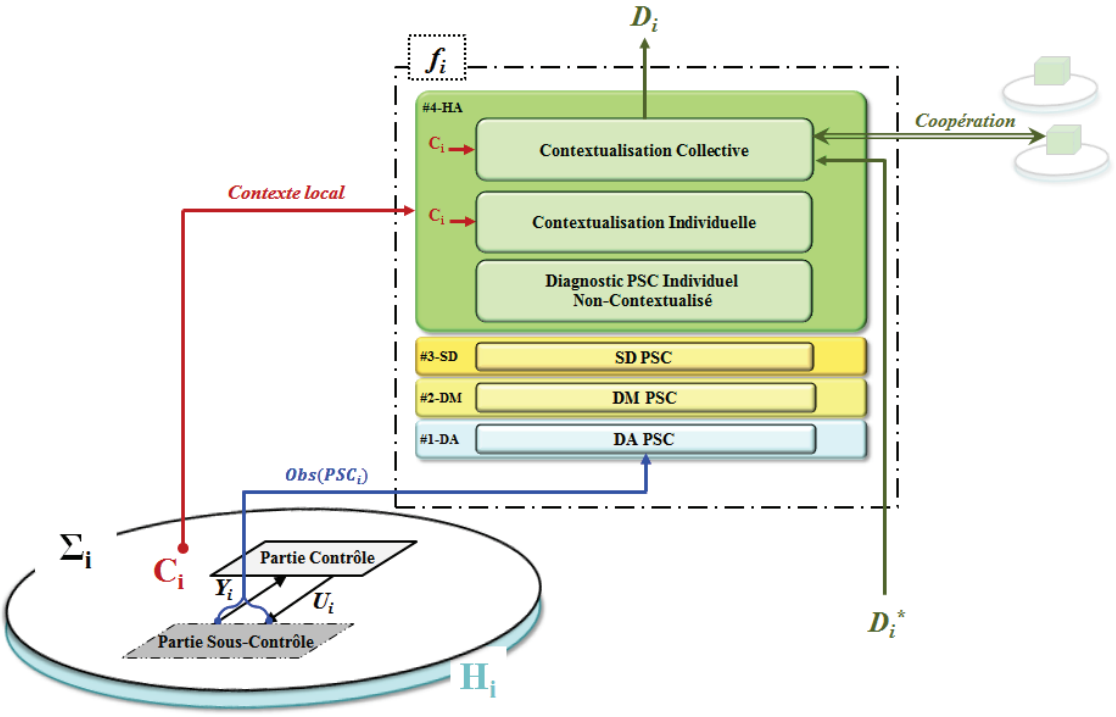

Figure 3.11. Modules internes d'une fonction de surveillance.

Les couches repérées #1 à #3 ont pour rôle d'observer le comportement de la Partie Sous-Contrôle (PSC) du système sous surveillance et de détecter des dysfonctionnements survenant dans cette partie.

La couche #4 de diagnostic permet d'analyser les dysfonctionnements détectés par la couche de niveau inférieur et réalise dans un premier temps le diagnostic de la partie PSC de manière individuelle grâce à un module de diagnostic. Ce module nommé *Diagnostic PSC Individuel Non-Contextualisé* génère des événements de diagnostic sans considérer le contexte opérationnel du système. Un événement de diagnostic représente un symptôme de dysfonctionnement associé à un ensemble de causes probables. Par exemple, un symptôme correspond à la détection d'un écart du comportement de la partie sous-contrôle par rapport à son comportement nominal et une cause probable est la défaillance d'un composant physique de cette partie.

Comme suggéré par la norme ISO 13374, le contexte opérationnel du système est exploité au niveau de la couche #4 dans le modèle proposé afin d'accroitre la robustesse du diagnostic. De ce fait, les informations fournies par le module de diagnostic individuel sont ensuite contextualisées par un premier module de contextualisation. Ce module, nommé *Contextualisation Individuelle* analyse des événements de diagnostic non contextualisés en exploitant le contexte local du système sous surveillance. Les événements de diagnostic générés par ce module sont ensuite traités par un second module de contextualisation nommé *Contextualisation Collective*. Ce module a pour objectif de délivrer un diagnostic du système sous surveillance à partir des informations de diagnostic contextualisées et des informations fournies par les holons de même niveau (contextes et états des autres sous-systèmes). Par ailleurs, ce module collecte et analyse des informations de diagnostic du niveau inférieur en fonction du contexte local du système et suivant les contextes et états transmis par le collectif de holons de même niveau. Ce module peut également être amené à partager des informations sur l'état et le contexte du système sous surveillance avec des holons de même niveau. Les

informations de diagnostic élaborées par ce dernier module sont propagées au holon de niveau supérieur.

2.4 Niveaux de diagnostic d'une fonction de surveillance

Le diagnostic d'un système sous surveillance est élaboré suivant trois niveaux de diagnostic. Le premier niveau permet d'établir individuellement le diagnostic de la partie PSC du système grâce aux modules associés aux couches #1 à #3 (figure 3.12). Le module de la couche #1 assure l'acquisition de signaux bruts de la partie PSC du système (DA PSC). Ces données sont consommées par le module de la couche #2 qui réalise des traitements sur ces signaux (DM PSC). Les signaux traités sont ensuite analysés par le module de la couche #3 afin de détecter des fautes survenant dans la partie PSC du système (SD PSC).

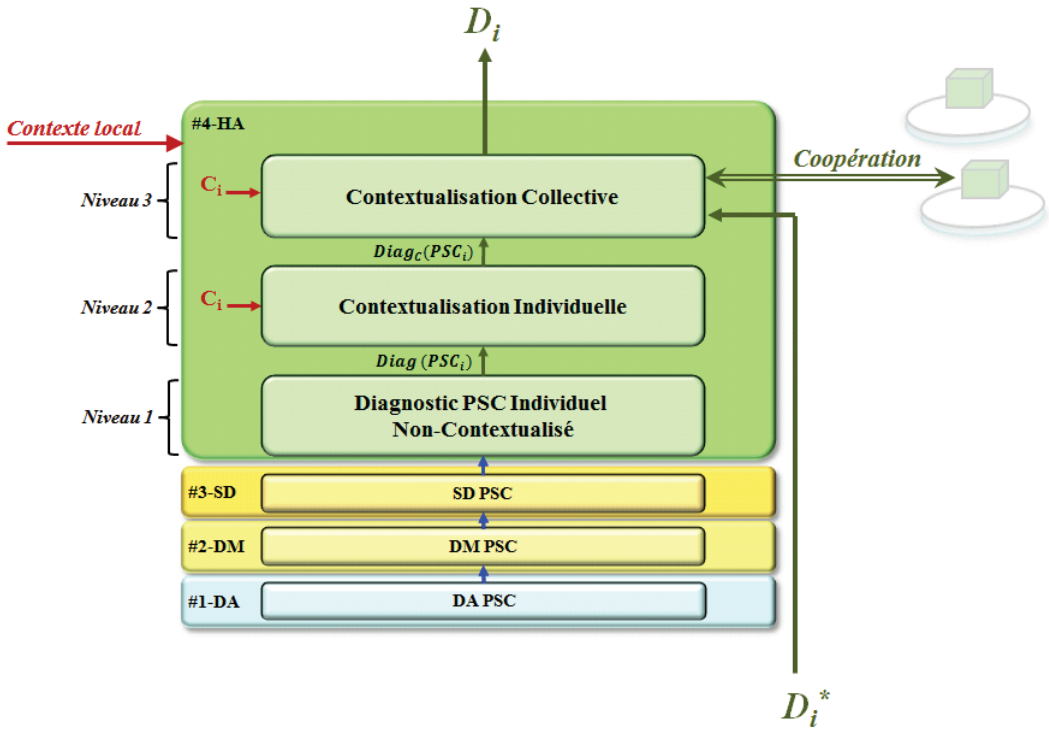

Figure 3.12. Niveaux de diagnostic d'une fonction de surveillance.

Le module de diagnostic de niveau 1 interne à la couche #4 procède à une identification des causes probables des fautes détectées et génèrent des événements de diagnostic relatifs à la partie sous-contrôle notés $Diag(PSC_i)$. Ces événements sont traités le module de diagnostic de niveau 2, qui permet d'évaluer leurs pertinences par rapport au contexte opérationnel local du système. Les événements de diagnostic contextualisés générés par ce modules, notés $Diag_C(PSC_i)$, sont ensuite analysés par le module de diagnostic de niveau 3.

2.4.1 Diagnostic de niveau 1

Le diagnostic de niveau 1 vise à déceler des fautes internes à la partie sous-contrôle survenant de manière soudaine ou progressive et à identifier leurs causes racines, sans exploiter le contexte local du système sous surveillance. Le diagnostic de niveau 1 peut être élaboré, par exemple, à partir d'une méthode de diagnostic exploitant une connaissance du comportement nominal ou dégradé de la partie sous-contrôle. Dans ce cas, le module de détection (SD PSC)

a pour rôle de confronter le comportement observé de la partie PSC avec son comportement nominal ou dégradé, afin de détecter des fautes survenant dans cette partie. Le module de diagnostic individuel non-contextualisé émet des hypothèses concernant les causes des fautes détectées par le module de détection et génère des événements de diagnostic, notés $Diag(PSC_i)$. Les éléments incriminés par le module de diagnostic PSC individuel non-contextualisé représentent des organes potentiellement défaillants de la partie PSC. Notons que ce module fait abstraction du fonctionnement de la partie contrôle du système sous surveillance.

2.4.2 Diagnostic de niveau 2

Le module de contextualisation de niveau 2 (Contextualisation Individuelle) analyse les événements de diagnostic élaborés par le module de diagnostic individuel de niveau 1. Ce module de contextualisation raisonne sur les conditions d'apparitions des événements de diagnostic de la partie PSC en fonction du contexte opérationnel local du système sous surveillance (C_i). En effet, la connaissance exploitée pour élaborer un diagnostic est généralement associée à un domaine de validité. Un résultat de diagnostic de niveau 1 peut alors être pertinent uniquement dans certains modes de fonctionnement du système ou lorsque certains critères de validité sont respectés. Ainsi, le module de contextualisation de niveau 2 vérifie la pertinence des résultats de diagnostic de niveau 1 en considérant la situation de fonctionnement du système sous surveillance, déterminée à partir de son contexte opérationnel. Par conséquent, les événements de diagnostic délivrés par ce module, notés $Diag_C(PSC_i)$, représentent les diagnostics pertinents de la partie sous-contrôle pour un contexte opérationnel donné.

2.4.3 Diagnostic de niveau 3

Le module de contextualisation de niveau 3 (Contextualisation Collective) raisonne à partir des événements de diagnostic contextualisés $Diag_C(PSC_i)$ de niveau 2 et des événements de diagnostic transmis par les holons du niveau inférieur (D_i^*) pour élaborer un diagnostic du système sous surveillance. Ce module accède également aux informations de contexte locales du système (C_i). Ainsi, les diagnostics des sous-systèmes de niveaux inférieurs réceptionnés par ce module peuvent être interprétés en fonction des événements de diagnostic apparus localement et du contexte local du système. Lorsque ce module élabore un diagnostic, les différents événements de diagnostic apparus dans un même intervalle de temps (T_{diag}), sont supposés être déclenchés par une seule et même cause, d'origine interne ou externe au système sous surveillance. A la différence du diagnostic contextualisé de niveau 2, le module de contextualisation de niveau 3 peut avoir accès aux informations de contexte et états des autres sous-systèmes de même niveau, fournies par leurs holons respectifs. Ce module joue donc un rôle actif, en sollicitant le collectif de holons situés à son niveau pour évaluer la pertinence des événements de diagnostic qu'il reçoit. Les événements de diagnostic analysés par ce module peuvent être une conséquence d'une faute survenue dans un sous-système avoisinant, ou un effet occasionné par l'environnement du système et non perceptible par le contexte local. Aussi, ce module joue un rôle participatif vis-à-vis du collectif de holons de même niveau, dans la mesure où il est capable de transmettre des diagnostics pertinents comme des informations de contexte du système sous surveillance aux autres holons.

3. Conclusion

Dans ce chapitre, une architecture holonique de surveillance nommée EMH a été proposée. Cette architecture permet d'élaborer un diagnostic embarqué robuste de manière décentralisée et coopérative à bord de systèmes cibles mobiles. Les travaux présentés dans ce chapitre ont fait l'objet de deux publications (Le Mortellec et al., 2012) et (Le Mortellec et al., 2013).

L'architecture holonique de surveillance proposée reflète la structure arborescente du système cible à surveiller obtenue par décomposition systémique, et repose sur des entités de surveillance autonomes, récursives et coopératives nommées holons. A chaque niveau de décomposition du système cible, un holon de surveillance est formé par un système tangible sous surveillance et une fonction de surveillance en charge d'établir un diagnostic de ce système. Chaque système sous surveillance est immergé dans un contexte opérationnel et comporte une partie contrôle et une partie sous-contrôle. Dans le but d'accroitre la pertinence du diagnostic et de limiter le nombre de fausses alarmes, un holon de surveillance considère le contexte opérationnel du système sous surveillance. De plus, un holon interprète des diagnostics élaborés par des holons de niveau inférieur, échange des informations avec des holons situés au même niveau, et propage ses résultats de diagnostic au holon de niveau supérieur.

Le diagnostic d'un système sous surveillance est élaboré suivant trois niveaux de diagnostic, dont le premier permet d'établir individuellement le diagnostic de la partie sous-contrôle du système sous surveillance. Le second et le troisième niveau représentent les deux aspects novateurs de notre proposition. Le second niveau de diagnostic réceptionne les résultats de diagnostic établis par le premier niveau et évalue la pertinence de ces résultats au regard du contexte opérationnel local du système sous surveillance. Enfin, le troisième niveau exploite des informations détenues par le collectif de holons situés au même niveau et affine le diagnostic du système avant de le diffuser au holon de niveau supérieur.

La mise en œuvre de cette architecture holonique de surveillance active est présentée dans le chapitre suivant.

Chapitre IV

Mise en œuvre de l'architecture holonique de surveillance active

Introduction

Ce chapitre présente une mise en œuvre de l'architecture de surveillance EMH (*Embedded Monitoring Holarchy*) présentée dans le chapitre précédent. L'architecture de surveillance proposée implémente une méthode de diagnostic à base de modèle ou MBD (*Model Based Diagnosis*) de type qualitatif. Ce choix résulte d'une étude comparative de méthodes de diagnostic annexée à ce manuscrit (*c.f.* annexe B). Cette méthode de diagnostic permet d'établir un diagnostic individuel non-contextualisé de la partie sous-contrôle (PSC) d'un système sous surveillance. Comme illustré sur la figure 4.1, cette méthode est implémentée par le module de diagnostic de niveau 1 relatif à une fonction de surveillance de l'architecture EMH (Diagnostic PSC Individuel Non-Contextualisé).

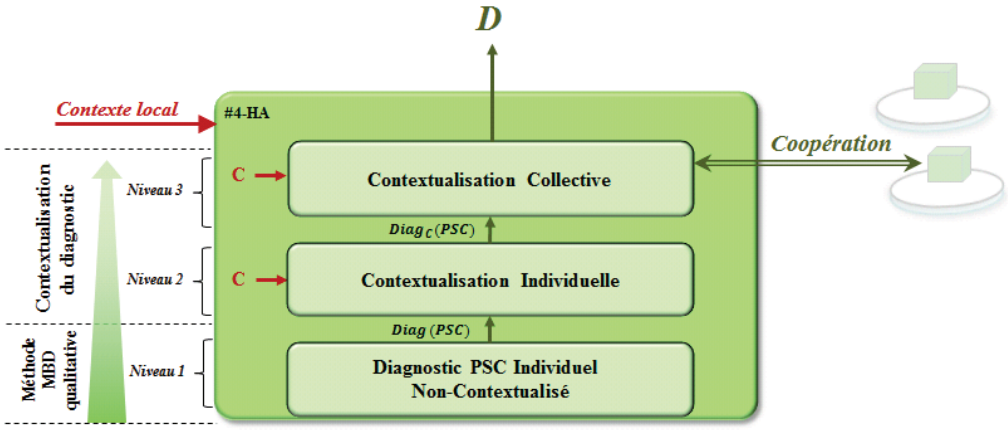

Figure 4.1. Modules de diagnostic internes à une fonction de surveillance.

La première partie de ce chapitre décrit la nature des systèmes diagnostiqués par cette méthode et présente les principales étapes de la procédure de diagnostic. La seconde partie détaille les modules proposés pour contextualiser les informations de diagnostic de manière individuelle (Contextualisation Individuelle) et collective (Contextualisation collective). Ces modules élaborent respectivement les diagnostics de niveau 2 et 3 (figure 4.1). La troisième partie expose les comportements attendus des holons au sein d'une architecture EMH. La quatrième partie présente l'approche proposée pour implémenter les holons qui composent cette architecture. Cette approche repose notamment sur le paradigme multi-agents présentée dans le chapitre 2.

1. Diagnostic PSC individuel non-contextualisé

Cette partie présente la méthode de diagnostic mise en œuvre pour élaborer le diagnostic individuel non contextualisé de niveau 1. Cette méthode est implémentée uniquement par les holons de surveillance situés au plus bas de l'architecture EMH. La méthode utilisée, nommée *Diagnostic Automatique*, a été développée par la société PROSYST et appliquée à des systèmes automatisés de production (Willaeys et Asse, 2011). Cette partie précise la nature des systèmes diagnostiqués par cette méthode (section 2.1) et décrit les principales étapes du processus de diagnostic (section 2.2). Une description plus détaillée de cette méthode MBD qualitative est donnée en annexes de ce manuscrit (*c.f.* annexe C).

1.1 Nature des systèmes diagnostiqués

La méthode de diagnostic utilisée vise à établir le diagnostic de manière autonome d'une classe de systèmes à événements discrets. Le système diagnostiqué est un système en boucle fermé constitué d'une partie contrôle (contrôleur) et d'une partie sous-contrôle (partie opérative), et caractérisé par des états discrets évoluant dans le temps suivant des événements asynchrones. La partie opérative de ce système est typiquement constituée d'éléments physiques (capteurs et actionneurs, par exemple), lesquels sont associés à des contraintes mécaniques et électriques. La partie contrôle exécute un programme indéfiniment de manière à assurer un fonctionnement attendu du système (exécution de lois de commandes, par exemple). Cette partie reçoit des signaux de mesures (entrées) et détermine les signaux de commandes appropriés (sorties) afin de commander la partie opérative conformément au programme qu'elle exécute. Les sorties sont généralement reliés à des actionneurs de la partie opérative. Ces actionneurs entrainent des mouvements mécaniques (rotation ou translation, par exemple).

Le comportement d'un tel système peut être analysé en observant les signaux échangés entre le contrôleur et la partie opérative. Ces signaux binaires représentent une séquence d'événements. Un événement est considéré lorsqu'un changement d'état survient dans un signal. L'occurrence et l'ordre d'apparition de ces événements sont régis par la dynamique du système. La dimension de cette séquence d'événements est intimement liée à la complexité de la partie opérative. Une partie opérative constituée de nombreux éléments implique généralement une quantité importante de signaux gérés par le contrôleur. Les signaux de mesures délivrés par la partie opérative sont exploités en premier lieu à des fins de commande du système par le contrôleur.

La section suivante présente une méthode de diagnostic visant à analyser cette séquence d'événements pendant le fonctionnement du système pour élaborer un diagnostic.

1.2 Méthode de diagnostic mise en œuvre

La méthode de diagnostic présentée ci-après repose sur une modèle structurel de la partie opérative d'un système sous surveillance, en faisant abstraction du fonctionnement de la partie contrôle. Elle se concentre sur le diagnostic de défaillances soudaines de la partie opérative et vise à incriminer des composants physiques de cette partie (actionneur bloqué ou capteur à contact mécanique collé, par exemple). Le modèle structurel exploité par cette méthode décrit le comportement correct des éléments de la partie opérative et de sa structure sous forme d'axes (Delépine et Caulier, 2005). Comme l'illustre la figure 4.2 ci-dessous, nous

considérons que cette méthode intègre les modules nécessaires pour élaborer le diagnostic de niveau 1 d'une fonction de surveillance. Par conséquent, cette méthode n'exploite pas le contexte opérationnel du système sous surveillance. Seuls les signaux discrets reçus et délivrés par la partie sous-contrôle (partie opérative) sont analysés pour établir un diagnostic du système.

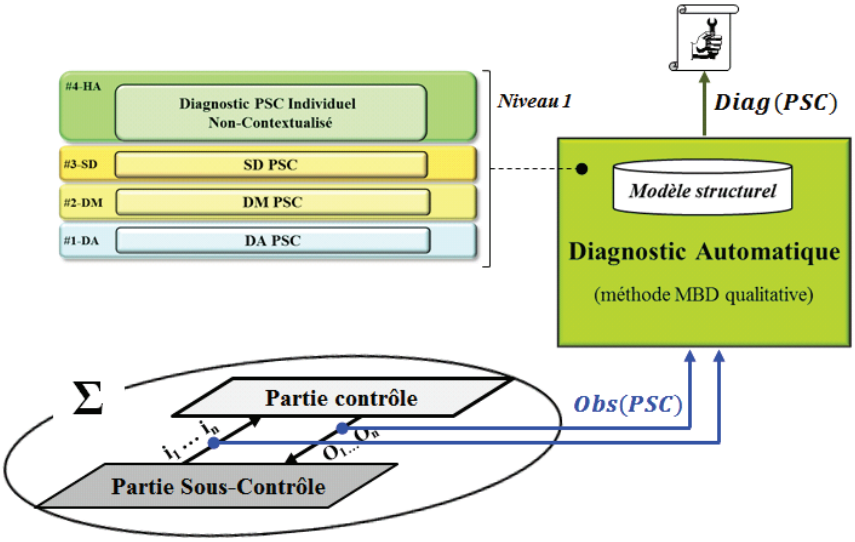

Figure 4.2. Schéma de principe de la méthode de diagnostic mise en œuvre.

Suivant les évolutions de ces signaux discrets, un algorithme de diagnostic permet de prédire les signaux qui seront émis par la partie sous-contrôle grâce à son modèle structurel, en faisant l'hypothèse que cette partie fonctionne normalement (modèle de bon fonctionnement). Une divergence entre le comportement prédit et le comportement observé de la partie sous-contrôle déclenche la génération d'une discordance, qui indique la présence d'une faute dans la partie sous-contrôle du système. L'algorithme de diagnostic analyse par la suite cette discordance en explorant le modèle structurel, afin de rechercher les composants suspects qui pourraient expliquer l'apparition de cette discordance. Cette méthode de diagnostic suppose qu'un seul composant de la partie sous-contrôle puisse être défaillant lors de l'élaboration d'un diagnostic. Le résultat d'un diagnostic délivré par cette méthode correspond à une liste de discordances, dans laquelle, à chaque discordance, est associée une liste de composants suspects. Ces informations sont exploitées par les opérateurs de maintenance pour réparer ou remplacer les composants défaillants du système.

Suivant cette méthode, un événement de diagnostic correspond à une discordance associée à une liste de composants potentiellement défaillants de la partie opérative d'un système sous surveillance. La section suivante souligne les limites de cette méthode de diagnostic.

1.3 Limitations de la méthode de diagnostic mise en œuvre

Comme toutes les méthodes de diagnostic à base de modèle, le modèle utilisé doit être fidèle au système à diagnostiquer. La méthode utilisée permet de diagnostiquer des défaillances internes à la partie sous-contrôle du système (partie opérative), en faisant l'hypothèse que la partie contrôle est infaillible. Ainsi, le modèle utilisé intègre une connaissance partielle du système, limitée à une connaissance de la structure et du comportement de la partie opérative. De plus, ce modèle admet des paramètres de modélisation associés à un domaine de validité. Le modèle peut alors s'avérer inexact dans certains contextes opérationnels (modèle non valable pour un mode de fonctionnement donné, par exemple). Enfin, les interactions

physiques et informationnelles entre le système et son environnement sont en dehors du périmètre de modélisation. Elles ne sont donc pas considérées lors de l'élaboration d'un diagnostic. Par exemple, une intervention humaine sur la partie opérative du système est susceptible d'entrainer la génération d'un diagnostic erroné par cette méthode. Ce diagnostic constitue une fausse alarme car aucune défaillance n'est apparue sur le système.

Les modules de contextualisation proposés dans la suite de ce chapitre visent à acquérir plus d'informations sur le système et son environnement pour enrichir le diagnostic et limiter notamment la génération de fausses alarmes.

2. Contextualisation du diagnostic

La fonction de surveillance d'un holon définie dans le précédent chapitre élabore le diagnostic d'un système sous surveillance suivant trois niveaux de diagnostic, dont les niveaux 2 et 3 contextualisent les résultats de diagnostic de manière individuelle et collective. Dans la mise en œuvre de l'architecture EMH proposée, le module de contextualisation individuelle est implémenté uniquement par les holons de surveillance qui élaborent un diagnostic de niveau 1, c'est à dire par les holons de surveillance situés en bas de l'architecture holonique. Le module de contextualisation collective proposé est implémenté par chaque holon de surveillance. En d'autres termes, une fonction de surveillance élémentaire située en bas de l'architecture holonique implémente les deux modules de contextualisation et une fonction de surveillance composée implémente uniquement le module de contextualisation collective (figure 4.3).

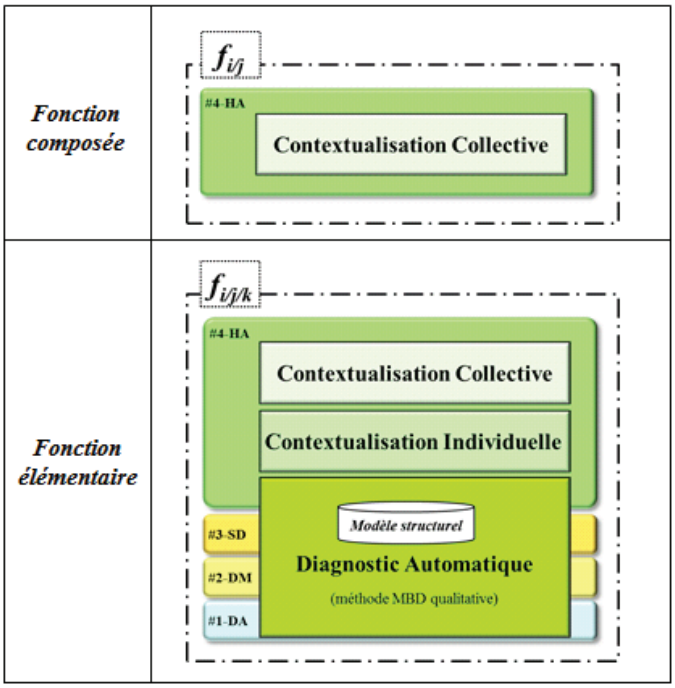

Figure 4.3. Modules implémentées par une fonction de surveillance composée et élémentaire.

Cette partie expose les modules proposés pour chaque niveau de contextualisation. Le module de contextualisation individuelle mis en œuvre est présenté dans la section 2.1 et le module de contextualisation collective est présenté dans la section 2.2.

2.1 Contextualisation individuelle

Le module de contextualisation individuelle proposé est générique et peut être mis en œuvre pour contextualiser le résultat d'un diagnostic quelle que soit la méthode de diagnostic de niveau 1 utilisée. Ce module limite la propagation de fausses alarmes en évaluant la pertinence des résultats de diagnostic suivant le contexte local du système diagnostiqué. Il agit comme un filtre, en autorisant la diffusion des résultats de diagnostic pertinents et en inhibant les diagnostics justifiés par le contexte local du système surveillé. Ce module de contextualisation individuelle repose sur un processus d'inhibition, qui analyse chaque événement de diagnostic reçu (discordance associée à une liste de composants suspects, par exemple), noté E, et exploite une information locale de contexte notée C pour délivrer un diagnostic contextualisé noté $Diag_C$ (figure 4.4).

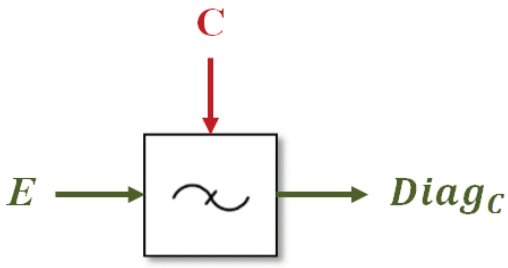

Figure 4.4. Schéma de principe du processus d'inhibition.

Un diagnostic contextualisé représente un diagnostic pertinent pour un contexte opérationnel donné. Lorsqu'un événement de diagnostic reçu en entrée de ce module n'est pas expliqué par le contexte local du système, le diagnostic contextualisé correspondant est l'image de cet événement. Dans ce cas, le contenu informationnel de l'événement de diagnostic analysé reste inchangé. Dans le cas contraire, c'est-à-dire lorsque le contexte du système permet de fournir une explication plausible sur l'apparition d'un événement de diagnostic, cet événement est filtré et aucun diagnostic contextualisé n'est généré. L'équation 4.1 ci-après exprime le diagnostic contextualisé $Diag_C$ en fonction d'un événement de diagnostic analysé E et d'une variable de contexte pertinente C pouvant justifier l'apparition de cet événement.

$$Diag_C = \begin{cases} \emptyset & \text{si C justifie } E \\ E & \text{sinon} \end{cases} \quad (4.1)$$

L'apparition d'un événement de diagnostic E provoque le déclenchement du processus d'inhibition. Cet événement correspond à un résultat de diagnostic de niveau 1, soit $Diag(PSC)$. Le processus d'inhibition illustré sur la figure 4.5 ci-après exécute des règles de filtrage sur les événements de diagnostic reçus. Les règles de filtrage sont contenues dans une base de règles regroupant les événements de diagnostic candidats et les variables de contexte susceptibles d'inhiber ces événements. Chaque règle utilise une variable de contexte qui peut être de nature continue (température du système, par exemple) ou discrète (état du système, par exemple). Une même variable de contexte peut permettre d'inhiber plusieurs événements de diagnostic.

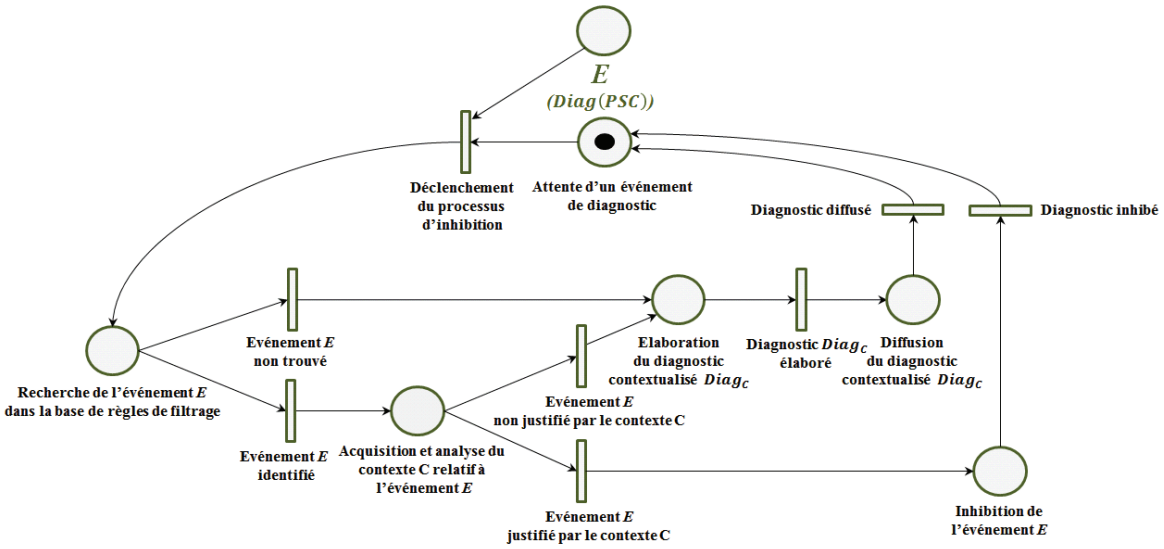

Figure 4.5. Processus d'inhibition sous forme de réseau de Petri.

La base de règles de filtrage peut être établie à partir des restrictions connues de la méthode de diagnostic utilisée (aspects non modélisés du système, domaine de validité du modèle, par exemple). Par ailleurs, des règles complémentaires peuvent être élaborées à partir d'une expertise humaine. Néanmoins, une compréhension des résultats fournis par la méthode de diagnostic de niveau 1 utilisée est nécessaire pour établir cette base de règles. La base de règle proposée se présente sous la forme d'une table contenant les événements de diagnostic candidats et les variables de contexte locales du système sous surveillance pertinentes pour ces événements. Le format de cette table est donné par la figure 4.6 ci-dessous. Dans cette table, une valeur "VRAI" indique que le processus doit inhiber un événement de diagnostic pour un contexte particulier. Par exemple, la première ligne de la table 4.6 peut être formulée de la manière suivante en langage naturel : **si** l'événement de diagnostic "E_1" est présent **et** que le système sous surveillance opère dans le contexte "C_1" **alors** cet événement de diagnostic est inhibé, **sinon** cet événement est diffusé.

		Contexte local C			
		Contexte C_1	Contexte C_2	…	Contexte C_k
Evénement diagnostic E	Evénement E_1	VRAI	FAUX		FAUX
$Diag(PSC)$	Evénement E_2	FAUX	FAUX		VRAI
	⋮	⋮	⋮		⋮
	Evénement E_n	FAUX	VRAI		VRAI

Figure 4.6. Format de la base de règles exploitée par le processus d'inhibition.

Le processus d'inhibition proposé priorise la diffusion des événements de diagnostic par rapport à l'inhibition d'événements. De ce fait, lorsqu'un événement de diagnostic ne figure pas dans la base de règles, cet événement n'est pas inhibé. Ce processus adopte un comportement plutôt réactif face aux événements qu'il reçoit (résultat d'un diagnostic, évolution d'une information de contexte) et présente certaines caractéristiques similaires à celles d'une architecture de *subsumption* (Brooks, 1986). Les événements non-inhibés par ce processus sont transmis au module de contextualisation collective décrit dans la section suivante.

2.2 Contextualisation collective

Le module de contextualisation collective considère les résultats de diagnostic issus du module de contextualisation individuelle (diagnostic de niveau 2) et bénéficie d'un apport d'informations du collectif de holons pour élaborer un diagnostic. Le module proposé est une extension du principe d'inhibition présenté auparavant. Il intègre des informations de contextes supplémentaires fournies par d'autres holons pour établir un contexte enrichi. Ce contexte enrichi peut potentiellement justifier l'apparition de certains événements de diagnostic. Ainsi, le module de contextualisation collective proposé repose sur un processus d'inhibition étendue, qui évalue la pertinence de chaque événement de diagnostic en dialoguant avec d'autres holons. Ce processus d'inhibition étendue permet de délivrer un diagnostic contextualisé noté D à partir d'un événement de diagnostic noté E et d'un contexte enrichi noté C_E (figure 4.7).

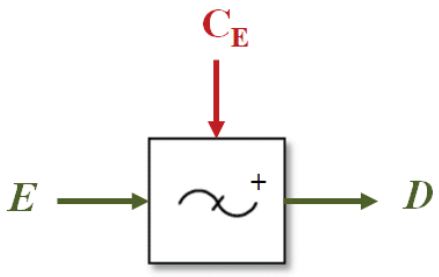

Figure 4.7. Schéma de principe du processus d'inhibition étendue.

Le contexte enrichi C_E représente un ensemble de variables de contexte pertinentes obtenues grâces aux informations locales de contexte et aux informations transmises par les holons de même niveau (interactions "horizontales"). L'expression d'un diagnostic contextualisé D en fonction d'un événement de diagnostic analysé E et d'un contexte enrichi C_E est donnée par l'équation ci-dessous (équation 4.2).

$$D = \begin{cases} \emptyset & \text{si } C_E \text{ justifie } E \\ E & \text{sinon} \end{cases} \quad (4.2)$$

Suivant le principe de surveillance récursive explicité dans le précédent chapitre, deux types d'événements de diagnostic peuvent être analysés dans le cas d'une fonction de surveillance composée :

- Les événements de diagnostic internes : ces événements sont les résultats de diagnostic de niveau 1 qui n'ont pas été inhibés par le module de contextualisation individuelle.
- Les événements de diagnostic externes : ces événements sont les résultats de diagnostic transmis par les holons de niveau inférieur.

Pour une fonction de surveillance élémentaire, les événements de diagnostic externes sont inexistants et seuls les événements de diagnostic internes sont analysés par ce processus.

Ainsi, le déclenchement du processus d'inhibition étendue est causé par un événement de diagnostic E de type interne ou externe (figure 4.8). Lorsque l'événement E est un événement de diagnostic interne, cet événement correspond à un événement de diagnostic de niveau 2, soit $Diag_C(PSC)$. Dans le deuxième cas, l'événement E représente un événement de diagnostic externe de niveau inférieur noté D^*.

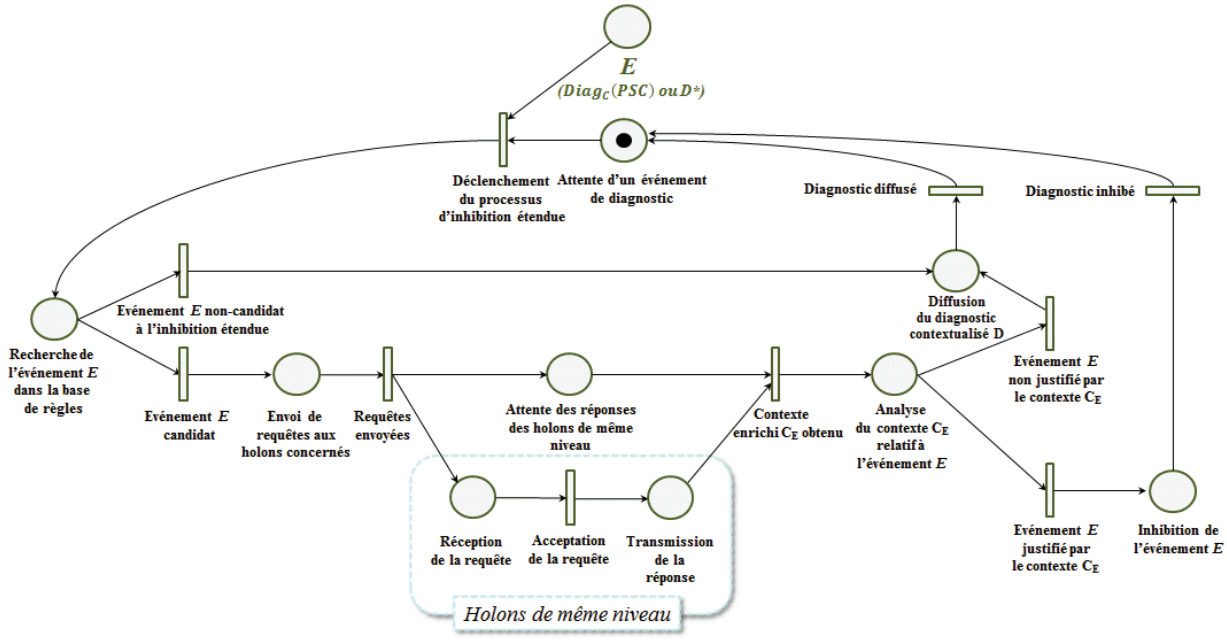

Figure 4.8. Processus d'inhibition étendue sous forme de réseau de Petri.

Le processus d'inhibition étendue exécute des règles de filtrage sur les événements de diagnostic déclencheurs ($Diag_C(PSC)$ ou D^*). Un évènement de diagnostic peut faire l'objet d'une inhibition étendue lorsqu'une règle lui est associée dans la base de règles. En l'absence de règles associées à un événement, celui-ci est diffusé à un holon de niveau supérieur et aucun échange n'a lieu avec d'autres holons de même niveau. Les règles définissent les événements de diagnostic candidats et les variables de contexte qui constituent le contexte enrichi. Ces variables peuvent potentiellement justifier l'apparition des événements de diagnostic. De ce fait, les interactions entre les holons sont implicitement définies par ces règles de filtrage. Lorsqu'un événement de diagnostic est défini dans la base de règles, les variables de contexte pertinentes pour cet événement sont obtenues par le biais de requêtes envoyées vers les holons qui détiennent ces variables. Ainsi, ce processus initie une phase de coopération pendant laquelle des holons sont sollicités pour obtenir des informations de contexte. Après cette phase, le contexte enrichi est évalué par ce processus. Lorsque le contexte enrichi permet de justifier l'apparition d'un événement de diagnostic, alors cet événement est inhibé. Dans le cas contraire, cet événement est diffusé à un holon de niveau supérieur. Le format de la base de règles exploitée par ce processus est présenté ci-dessous (figure 4.9).

		Contexte enrichi C_E			
		Contexte C_{L1} *(contexte local)*	Contexte C_{D1} *(contexte distant : Holon $H_{i/j}$)*	…	Contexte C_{Dk} *(contexte distant : Holon $H_{i/j}$)*
Evénement diagnostic E	Evénement E_1	VRAI	VRAI		FAUX
$Diag(PSC)$ *ou D**	Evénement E_2	FAUX	VRAI		VRAI
	⋮	⋮	⋮		⋮
	Evénement E_n	FAUX	FAUX		VRAI

Figure 4.9. Format de la base de règles exploitée par le processus d'inhibition étendue.

Une valeur "VRAI" inscrite dans cette base de règles signifie que l'événement de diagnostic candidat est inhibé à partir du contexte enrichi C_E. Dans ce cas, au moins une variable de contexte locale ou une variable de contexte distante, c'est-à-dire une variable détenue par un autre holon, justifie la présence de cet événement. Ces règles peuvent être définies, par exemple, à partir des influences externes exercées par l'environnement du système sous surveillance et non prises en compte par la méthode de diagnostic mise en œuvre (interactions entre sous-systèmes, défaillances de causes communes, par exemple). Une expertise humaine peut également permettre de définir ces règles de filtrage. Dans la table proposée ci-dessus (figure 4.9), la variable locale de contexte C_{L1} permet par exemple de justifier l'événement de diagnostic E_1. Ce même événement peut également être justifié par la variable distante de contexte C_{D1} fournie par un holon spécifique. Dans cet exemple, l'apparition de l'événement E_1 implique alors une coopération avec un holon de même niveau pour obtenir cette variable distante de contexte.

Le processus d'inhibition étendue proposée n'est pas associé à un protocole spécifique de communication. Néanmoins, un mode de communication par échanges de messages est adopté. Une requête est donc transmise à chaque holon concerné pendant la phase de coopération. Cette requête peut être acceptée ou refusée par les holons sollicités. Le comportement de ce processus peut être qualifié de proactif puisqu'il interroge des holons spécifiques selon les événements de diagnostic qu'il reçoit.

3. Description comportementale des holons de surveillance

Cette partie expose les différents comportements attendus des holons de surveillance au sein de l'architecture EMH en considérant les modules de contextualisation individuelle et collective proposés précédemment. Chaque comportement met en évidence l'activité interne d'un holon de l'architecture ainsi que les interactions entre ces entités. Dans la suite de nos travaux, les holons de surveillance seront donc programmés de manière à satisfaire ces différents comportements. Afin de décrire les comportements de ces holons, une architecture EMH associée à un système cible noté Σ_1 est définie. Cette architecture, illustrée sur la figure 4.10, comporte trois holons de surveillance composés (H_1, $H_{1/1}$, $H_{1/2}$) et quatre holons de surveillance élémentaires ($H_{1/1/1}$, $H_{1/1/2}$, $H_{1/2/1}$, $H_{1/2/2}$). Par conséquent, les fonctions de surveillance $f_{1/1/1}$, $f_{1/1/2}$, $f_{1/2/1}$ et $f_{1/2/2}$ implémentent les modules de contextualisation individuelle et collective détaillés dans la section précédente. Les fonctions f_1, $f_{1/1}$ et $f_{1/2}$ implémentent uniquement le module de contextualisation collective.

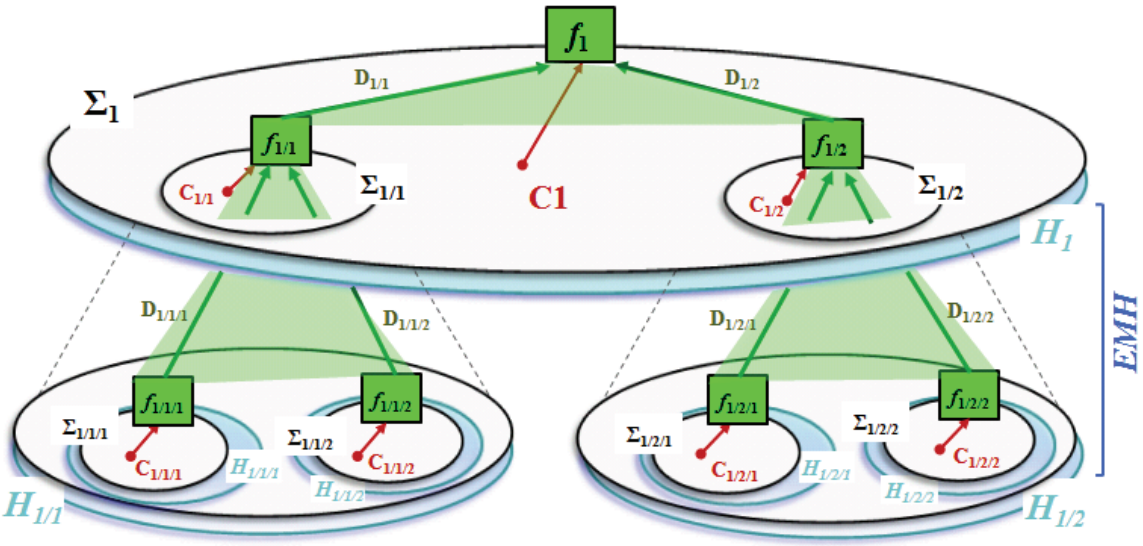

Figure 4.10. Architecture EMH considérée pour la description comportementale des holons.

Le formalisme utilisé pour décrire les comportements des holons est le diagramme de séquences. Un tel digramme permet de représenter graphiquement les activités internes des holons et leurs interactions en y intégrant une dimension temporelle. La figure 4.11 présente un exemple de diagramme de séquences représentant un échange de messages entre les holons $H_{1/1/1}$ et $H_{1/1/2}$.

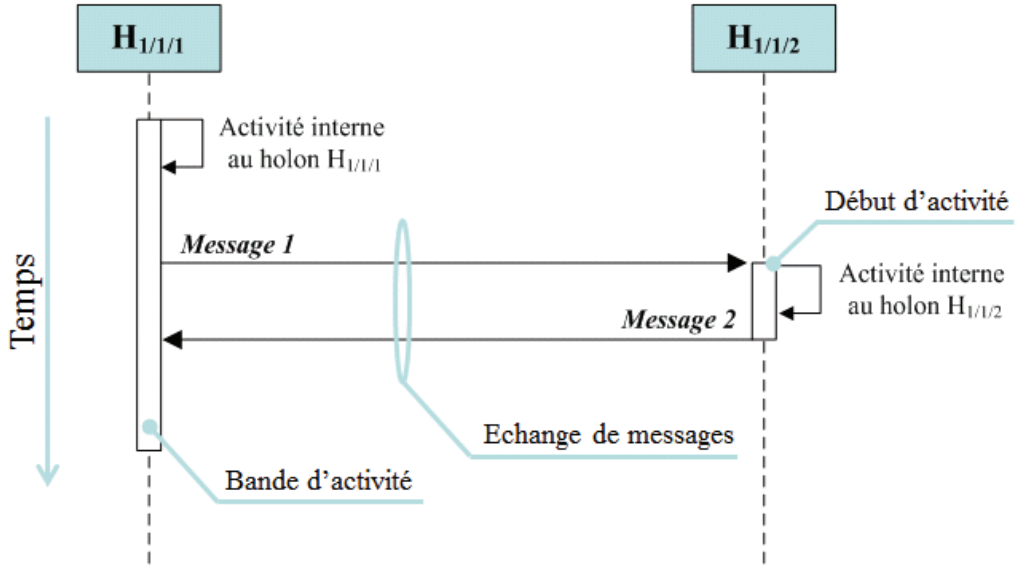

Figure 4.11. Exemple de diagramme de séquences entre les holons $H_{1/1/1}$ et $H_{1/1/2}$.

Les comportements des holons élémentaires et composés sont spécifiés dans les sous-sections suivantes. Pour chaque comportement, les interactions "verticales" et "horizontales" entre les holons seront représentées sur un même diagramme de séquences.

3.1 Comportements des holons élémentaires

Cette section présente les comportements des holons élémentaires $H_{1/1/1}$, $H_{1/1/2}$, $H_{1/2/1}$ et $H_{1/2/2}$ illustrés sur la figure 4.10. Le holon $H_{1/1/1}$ est choisi arbitrairement pour décrire les comportements attendus de ces holons.

3.1.1 Diffusion d'un diagnostic contextualisé

Dans ce comportement, le holon élabore un diagnostic et le diffuse à un holon de niveau supérieur après une contextualisation individuelle et collective. Pour définir ce comportement, nous supposons que la méthode de diagnostic de niveau 1 du holon $H_{1/1/1}$ génère un diagnostic avéré suite à l'occurrence d'une défaillance dans la partie $PSC_{1/1/1}$ du sous-système $\Sigma_{1/1/1}$. Ainsi, le holon $H_{1/1/1}$ élabore un diagnostic $Diag(PSC_{1/1/1})$ de niveau 1 qui déclenche le processus d'inhibition interne à ce holon (diagnostic de niveau 2). Le contexte $C_{1/1/1}$ ne permettant pas de fournir une explication plausible sur l'apparition de cet événement de diagnostic, le processus d'inhibition transmet cet événement au module de contextualisation collective (diagnostic de niveau 3), qui déclenche alors le processus d'inhibition étendue. Etant donné que cet événement n'est pas candidat à une inhibition étendue (événement non présent dans la base de règles associée à ce processus, par exemple), ce diagnostic est diffusé au holon $H_{1/1}$ situé immédiatement au niveau supérieur. Ce comportement est illustré par le diagramme de séquences ci-dessous (figure 4.12).

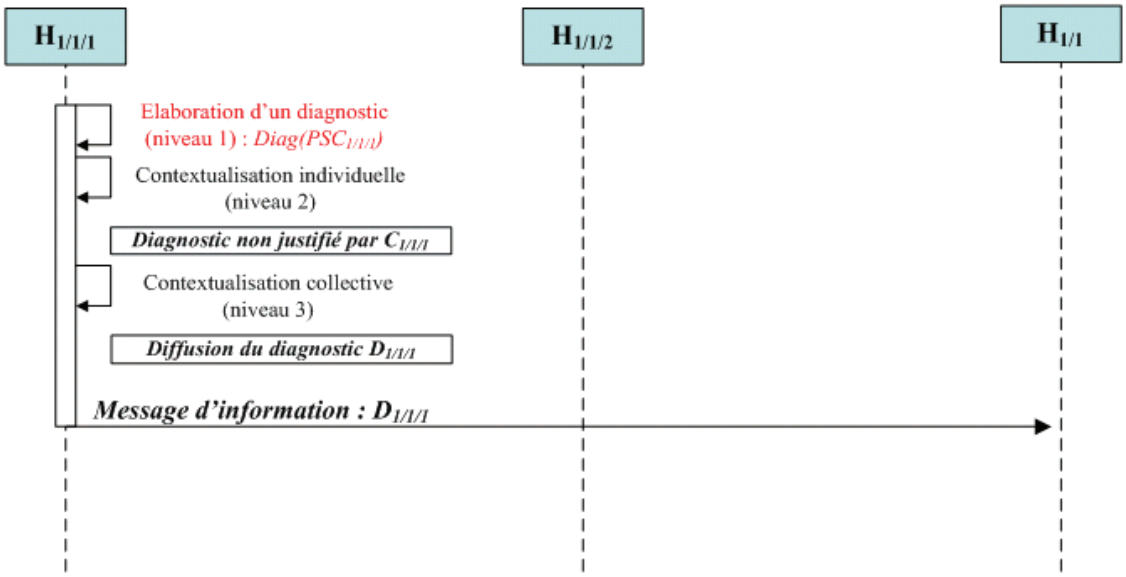

Figure 4.12. Diffusion d'un diagnostic contextualisé.

Le module de contextualisation collective assure ici la transmission d'un événement de diagnostic non-inhibé par le module de contextualisation individuelle à un holon de niveau supérieur. Les interactions "horizontales" entre holons sont inexistantes. Ainsi, le holon $H_{1/1/1}$ transmet un message d'information contenant le diagnostic $D_{1/1/1}$ au holon $H_{1/1}$ de niveau supérieur.

3.1.2 Inhibition individuelle d'un diagnostic

Ce comportement permet au holon de ne pas diffuser un diagnostic erroné à un holon de niveau supérieur à partir de ses observations locales. Dans ce comportement, le module de contextualisation individuelle inhibe un événement de diagnostic de niveau 1. Nous supposons que la méthode de diagnostic de niveau 1 du holon $H_{1/1/1}$ génère un diagnostic erroné. Ainsi, le holon $H_{1/1/1}$ élabore un diagnostic $Diag(PSC_{1/1/1})$ de niveau 1 et déclenche son processus d'inhibition (diagnostic de niveau 2). Ce processus analyse le contexte $C_{1/1/1}$ qui apporte une explication plausible à l'apparition de cet événement de diagnostic. Par conséquent, le diagnostic $Diag(PSC_{1/1/1})$ est inhibé par le module de contextualisation individuelle. Le processus d'inhibition étendue n'est alors pas déclenché et aucun diagnostic n'est diffusé au holon $H_{1/1}$ de niveau supérieur. La figure 4.13 présente le diagramme de séquences de ce comportement.

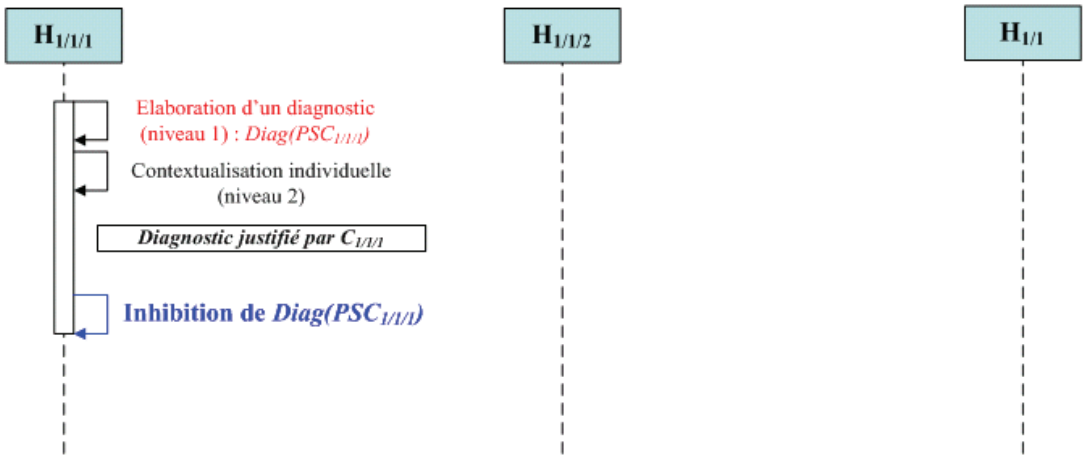

Figure 4.13. Inhibition individuelle d'un diagnostic.

Les interactions "horizontales" et "verticales" entre holons sont inexistantes pour ce comportement. En effet, le holon $H_{1/1/1}$ inhibe le diagnostic qu'il élabore uniquement grâce à son contexte $C_{1/1/1}$.

3.1.3 Inhibition collective d'un diagnostic

Ce comportement permet au holon d'inhiber un diagnostic qu'il élabore grâce à un contexte enrichi résultant d'une coopération avec des holons de même niveau. Dans ce comportement, le module de contextualisation collective inhibe un événement de diagnostic en sollicitant le collectif de holons situés au même niveau. Nous supposons que la méthode de diagnostic de niveau 1 du holon $H_{1/1/1}$ génère un diagnostic qui n'est pas expliqué par le contexte local $C_{1/1/1}$, mais qui est justifié par le contexte du holon $H_{1/1/2}$. De ce fait, le diagnostic $Diag(PSC_{1/1/1})$ n'est pas inhibé par le module de contextualisation individuelle du holon $H_{1/1/1}$ (diagnostic de niveau 2). Il est analysé par le module de contextualisation collective interne au holon $H_{1/1/1}$ et déclenche alors le processus d'inhibition étendue (diagnostic de niveau 3). Ce comportement est décrit par le diagramme de séquences ci-dessous (figure 4.14). Le processus d'inhibition étendue initie une coopération avec le holon $H_{1/1/2}$. Le holon

$H_{1/1/1}$ initiateur de la coopération envoie une requête au holon $H_{1/1/2}$ afin d'obtenir des informations de contexte complémentaires (message repéré 1 sur la figure 4.14). Cette requête est acceptée par le holon $H_{1/1/2}$ qui transmet par la suite des informations de contexte au holon $H_{1/1/1}$ (message repéré 2 sur la figure 4.14). Enfin, le holon $H_{1/1/1}$ analyse ces informations et détermine que le contexte enrichi apporte une explication plausible à l'apparition de l'événement de diagnostic analysé. Cet événement est alors inhibé par le holon $H_{1/1/1}$.

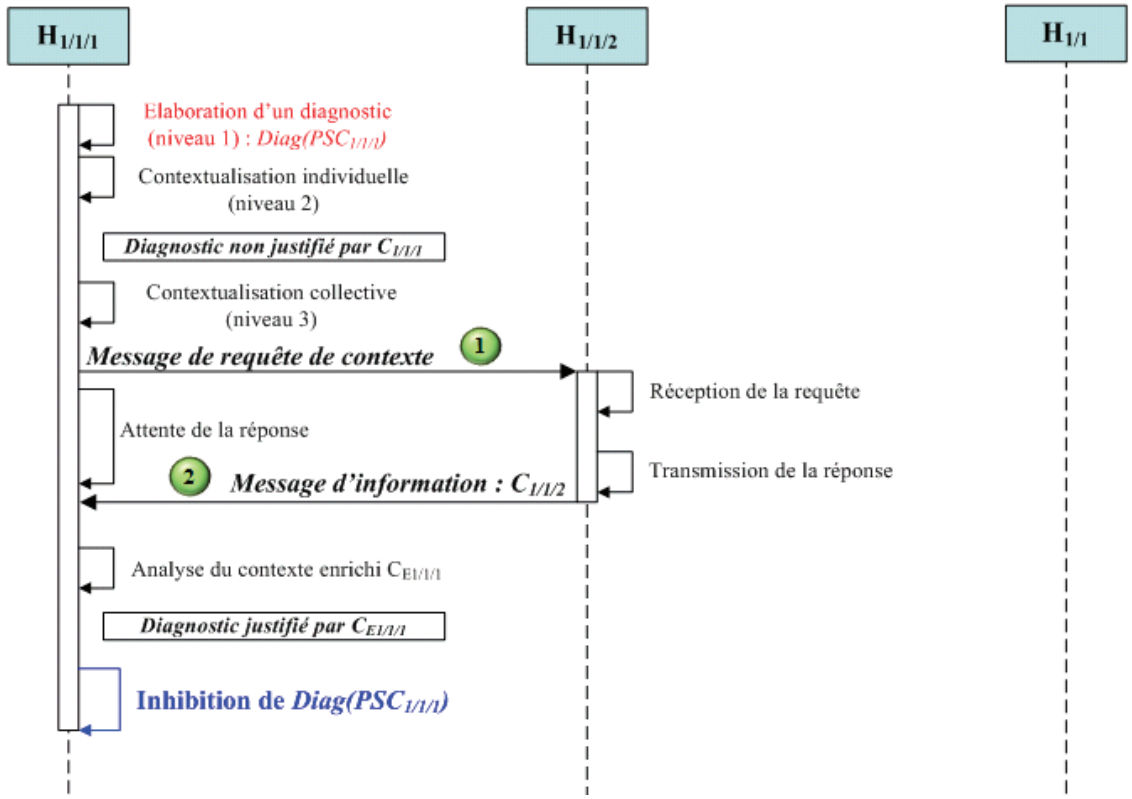

Figure 4.14. Inhibition collective d'un diagnostic.

Pour ce comportement, le module de contextualisation collective permet d'inhiber un événement de diagnostic à partir des informations fournies par des holons de même niveau. Les interactions "horizontales" entre holons permettent d'éviter la diffusion d'un diagnostic à un niveau supérieur.

3.2 Comportements des holons composés

Les holons composés, notés H_1, $H_{1/1}$, $H_{1/2}$ sur la figure 4.10, ne peuvent élaborer un diagnostic de niveau 1. Ces holons contextualisent de manière collective les événements de diagnostic de leurs holons "fils". Les deux comportements attendus de ces holons sont décrits dans les sous-sections suivantes. Le holon $H_{1/1}$ est retenu pour illustrer ces comportements.

3.2.1 Propagation d'un diagnostic contextualisé

Dans ce comportement, le holon reçoit un diagnostic contextualisé de l'un de ses holons "fils" de niveau inférieur et le propage à un autre holon de niveau supérieur. Le diagramme de séquences associé à ce comportement est présenté ci-après (figure 4.15). Un événement de diagnostic de niveau inférieur, noté D^*, est reçu par le holon $H_{1/1}$. Cet événement déclenche

le processus d'inhibition étendue interne à ce holon. En supposant que cet événement soit candidat à une inhibition étendue et que son analyse nécessite des informations détenues par la holon $H_{1/2}$, les holons $H_{1/1}$ et $H_{1/2}$ coopèrent. Ainsi, le holon $H_{1/1}$ envoie une requête au holon $H_{1/1}$ (message repéré 1 sur la figure 4.15) et analyse les informations transmises par ce holon pour obtenir un contexte enrichi (message repéré 2). Le contexte enrichi ne permettant pas de justifier l'événement D^*, le holon $H_{1/1}$ propage cet événement au holon situé immédiatement au niveau supérieur, c'est-à-dire au holon H_1.

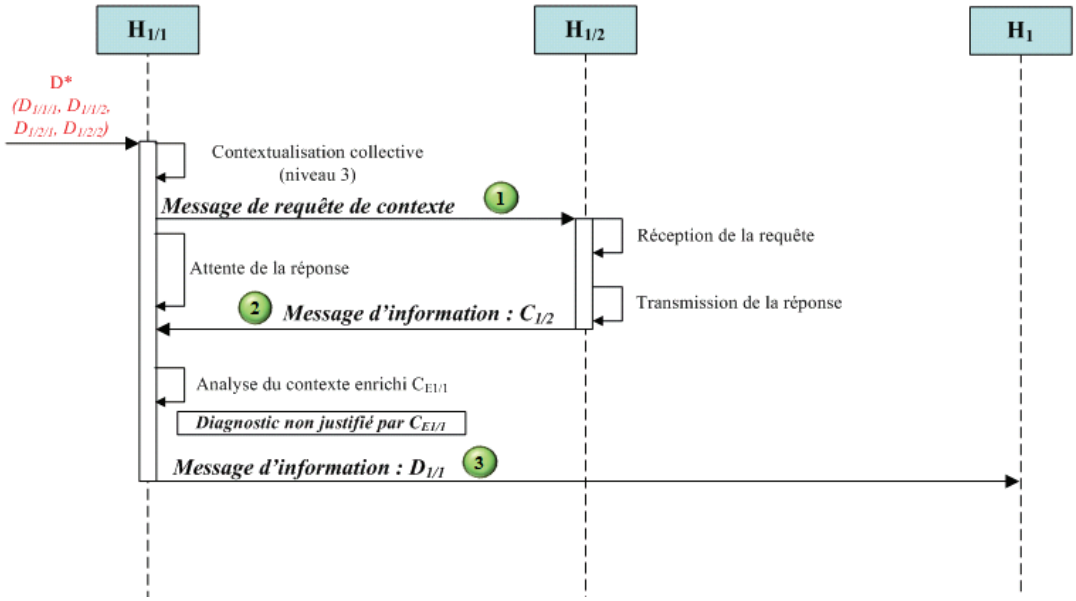

Figure 4.15. Propagation d'un diagnostic contextualisé.

Le holon $H_{1/1}$ confirme ainsi la pertinence de l'événement D^* à son niveau et le transmet au holon H_1 par un message d'information (message repéré 3 sur la figure 4.15). Notons que l'événement transmis par le holon $H_{1/1}$, noté alors $D_{1/1}$, provoque à son tour le déclenchement du processus d'inhibition étendue interne au holon H_1. L'activité associée à ce holon n'est pas représentée sur la figure 4.15.

3.2.2 Inhibition d'un diagnostic propagé

Ce comportement permet à un holon d'inhiber un diagnostic de niveau inférieur et, par conséquent, de ne pas le propager à un holon de niveau supérieur. Dans ce comportement, le contexte enrichi permet de justifier l'apparition d'un événement de diagnostic de niveau inférieur. Le diagramme de séquences correspondant à ce comportement est illustré ci-dessous (figure 4.16).

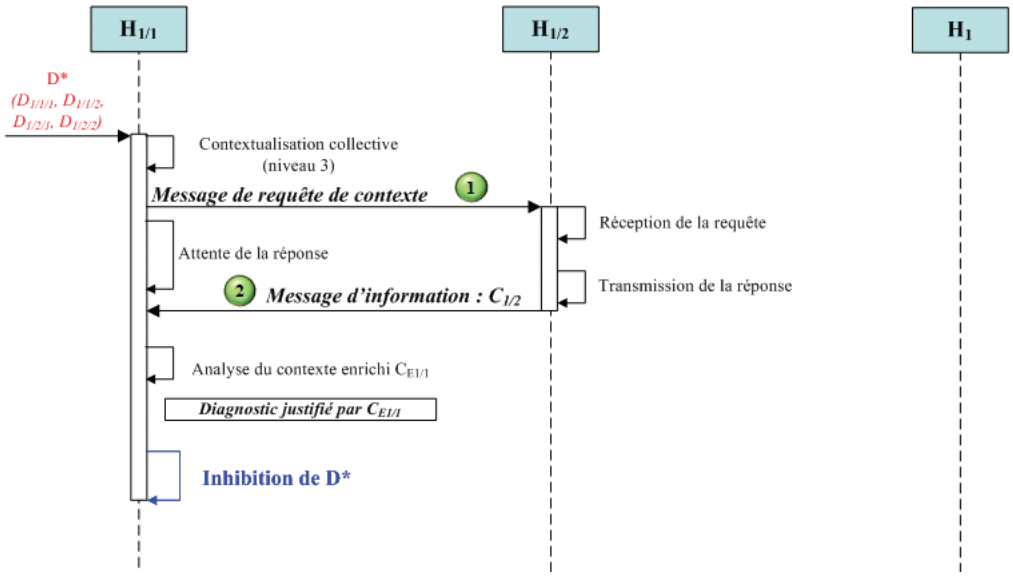

Figure 4.16. Inhibition d'un diagnostic propagé.

Le déclenchement du processus d'inhibition étendue interne au holon $H_{1/1}$ est provoqué par l'événement de diagnostic de niveau inférieur D^*. Il est supposé que cet événement peut potentiellement être expliqué par des informations de contexte détenues par le holon $H_{1/2}$. Le holon $H_{1/1}$ interroge alors le holon $H_{1/2}$ (messages de requête repéré 1 et de réponse repéré 2 sur la figure 4.16). Etant donné que les informations fournies par le holon $H_{1/2}$ permettent de justifier l'apparition de l'événement D^*, ce diagnostic est inhibé et aucun diagnostic n'est transmis au holon H_1 de niveau supérieur. Ainsi, plus les événements de diagnostic parviennent à un niveau élevé de la hiérarchie, plus ces événements sont pertinents.

Dans la partie suivante, une approche est proposée pour mettre en œuvre les holons de surveillance et implémenter les différents comportements définis précédemment.

4. Mise en œuvre des holons de surveillance

Cette partie propose une implémentation des holons de surveillance dans le but de concevoir une architecture de surveillance EMH opérationnelle. La section 4.1 présente l'approche proposée pour implémenter les parties informationnelle des holons de surveillance. La section 4.2 décrit la manière dont ces entités peuvent être supportées par des systèmes matériels.

4.1 Implémentation de la partie informationnelle d'un holon

Dans une architecture EMH, la fonction de surveillance d'un holon constitue sa partie informationnelle. Elle délivre des informations de diagnostic relatives à la partie physique de ce holon. De notre point de vue, une approche à base d'agents est appropriée pour implémenter la partie informationnelle d'un holon. La technologie agent offre notamment des environnements de développement dédiés (plateformes multi-agents) qui peuvent être utilisés pour implémenter les traitements informationnels associés aux comportements des holons et à leurs interactions. Une approche multi-agents est donc choisie pour implémenter les parties informationnelles des holons de surveillance sous formes d'agents logiciels. Dans l'implémentation proposée, les traitements relatifs à la méthode de diagnostic mise en œuvre (diagnostic de niveau 1) sont dissociés des traitements liés à la contextualisation du diagnostic (diagnostic de niveau 2 et 3). Par conséquent, dans le cas d'un holon élémentaire, un agent nommé Agent Diagnostic PSC (ADSPC) implémente le module de diagnostic PSC individuel non contextualisé et un agent nommé Agent Contextuel (AC) implémente les deux modules de contextualisation internes à une fonction de surveillance (figure 4.17).

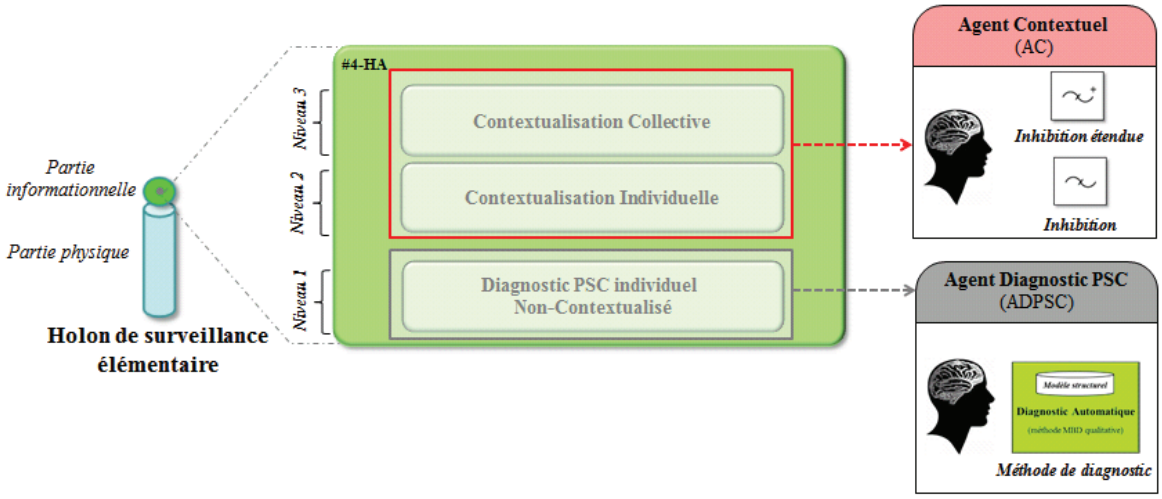

Figure 4.17. Implémentation de la partie informationnelle d'un holon élémentaire.

L'agent ADPSC prend en charge la méthode de diagnostic à base de modèle nommée *Diagnostic Automatique* présentée au début de ce chapitre. Cet agent élabore le diagnostic de la partie PSC d'un système sous surveillance et transmet les résultats de diagnostic à l'agent AC. Ce dernier exécute le processus d'inhibition (contextualisation individuelle) et le processus d'inhibition étendue (contextualisation collective) pour contextualiser les résultats de diagnostic. L'agent AC perçoit le contexte opérationnel du système sous surveillance et analyse les événements de diagnostic transmis par l'agent ADPSC.

La partie informationnelle d'un holon composé est implémentée par un agent AC. Cet agent analyse les informations de diagnostic transmis par les agents AC relatifs aux holons de

niveau inférieur. La figure 4.18 ci-après illustre l'approche proposée pour implémenter la structure holonique de l'architecture EMH. Un holon composé d'autres holons est implémenté par un système multi-agents dans lequel les agents sont organisés de manière hiérarchisée. Un agent AC reçoit des informations de diagnostic des agents AC de niveau inférieur (interactions inter-niveaux) et peut coopérer avec d'autres agents AC situés au même niveau (interactions intra-niveaux). Par ailleurs, l'agent AC transmet des informations de diagnostic à un agent AC de niveau supérieur.

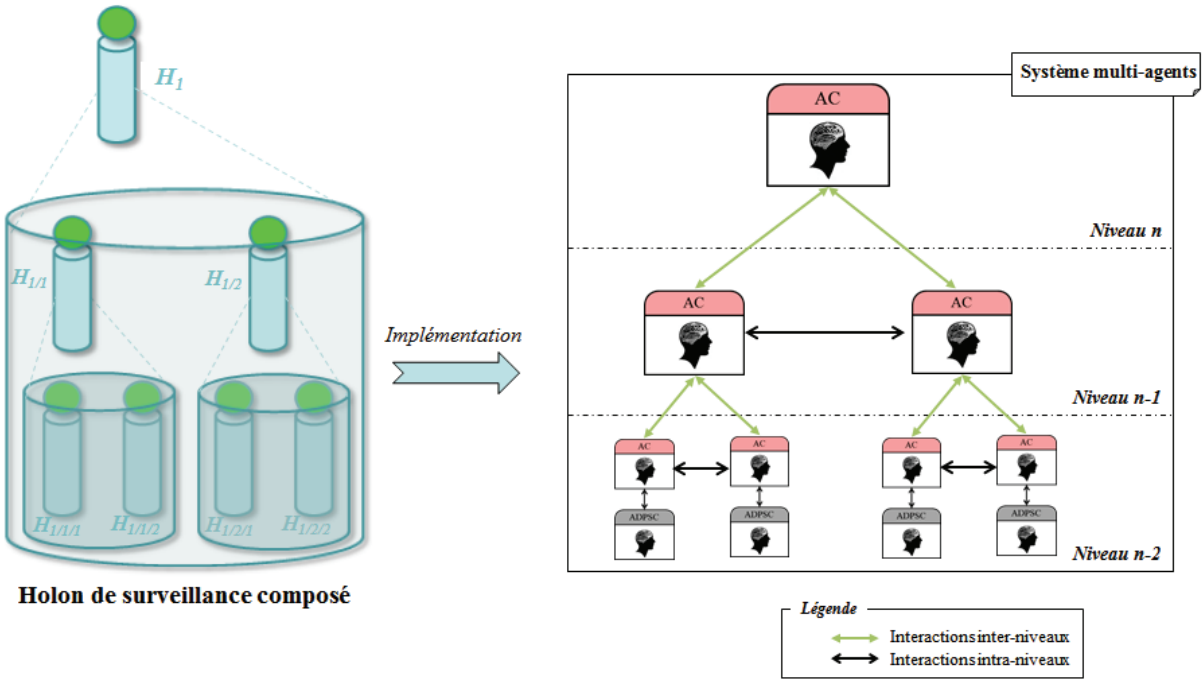

Figure 4.18. Exemple de holon composé implémenté par un système multi-agents.

Les interactions entre les agents du système sont imposées par les liens hiérarchiques entre les holons mis en œuvre. En effet, en considérant l'exemple présenté sur la figure 4.18, un agent AC de niveau *n-2* est en principe non autorisé à interagir avec un agent AC de niveau *n*. Celui-ci peut néanmoins communiquer avec un agent AC de même niveau et avec un agent AC de niveau supérieur.

La vision organique de cette implémentation est abordée dans la section suivante.

4.2 Implantation physique des holons de surveillance

Les agents logiciels nécessaires à l'implémentation des parties informationnelles des holons peuvent être supportés par un ou plusieurs dispositifs matériels. Par conséquent, la définition d'une architecture opérationnelle de surveillance passe par une allocation des différents agents sur des organes physiques de l'architecture (supports exécutifs des agents). Comme l'illustre la figure 4.19, chaque holon peut être supporté par un organe, désigné *système* par la suite. Dans cet exemple, chaque système exécute un ou plusieurs agents qui implémentent la partie informationnelle d'un holon. Dans le cas d'un holon composé, seul un agent AC réside sur le système supportant la partie informationnelle de ce holon. Les agents AC et ADPSC relatifs à un holon élémentaire sont implémentés sur un même système.

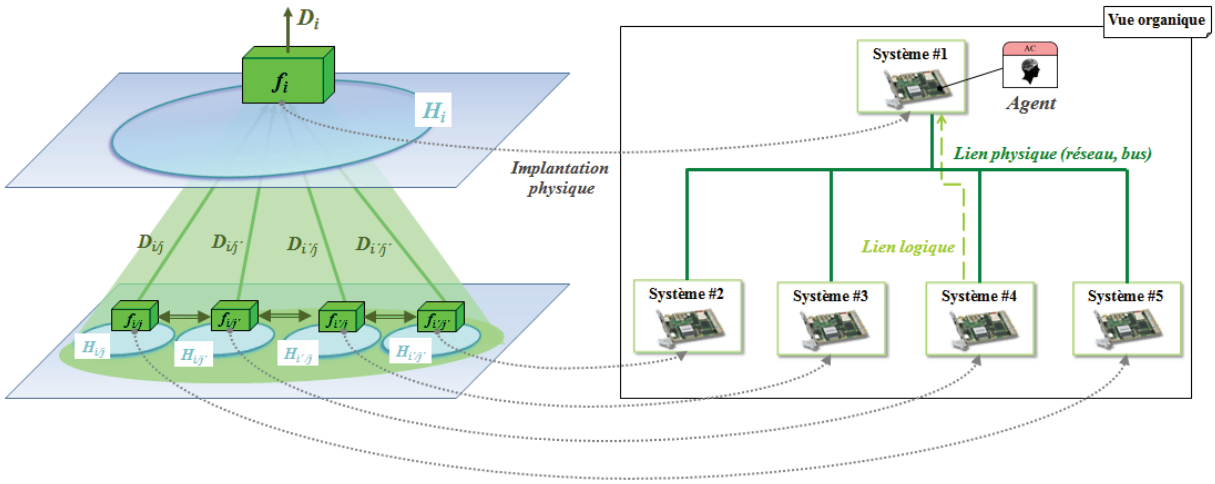

Figure 4.19. Exemple de projection de holons de surveillance sur des systèmes matériels.

D'un point de vue organique, les systèmes matériels sont interconnectés entre eux par le biais d'un lien physique (réseau ou bus de terrain, par exemple). Dans l'exemple présenté sur la figure 4.19, une architecture distribuée est formée par une projection des holons de surveillance sur des systèmes matériels (au sens informatique du terme). Les liens logiques entre ces systèmes matériels sont imposés par les interactions informationnelles entre les agents logiciels qu'ils supportent.

Afin de garantir la "non-intrusivité" de l'architecture de surveillance, nous supposons que les systèmes qui supportent les parties informationnelles des holons de surveillance diffèrent des parties physiques du système cible sous surveillance. Par ailleurs, nous suggérons une liaison physique dédiée entre ces systèmes (réseau de diagnostic spécifique, par exemple). De ce fait, les informations échangées entre ces systèmes n'interfèrent pas avec les interactions informationnelles des sous-systèmes du système cible.

5. Conclusion

Ce chapitre a présenté une mise en œuvre de l'architecture de surveillance EMH proposée dans le précédent chapitre. Dans la première partie de ce chapitre, la méthode de diagnostic implémentée par une fonction de surveillance pour élaborer un diagnostic de niveau 1 a été présentée. Cette méthode de diagnostic à base de modèle qualitatif fait abstraction des informations de contexte du système surveillé pour établir un diagnostic. Les limitations de cette méthode sont liées principalement à la modélisation partielle du système diagnostiqué. Dans la seconde partie, les modules de contextualisation individuelle et collective implémentés par une fonction de surveillance ont été proposés. Le premier repose sur un processus d'inhibition, visant à inhiber des événements de diagnostic à partir des informations de contexte locales observées par un holon. Le second est basé sur un processus d'inhibition étendue. Ce dernier consiste à inhiber des événements de diagnostic grâce aux informations transmises par d'autres holons et relatives à l'environnement du système sous surveillance. La troisième partie de ce chapitre a défini les comportements attendus des holons de surveillance suivant ces processus. Enfin, la quatrième partie a proposé une approche multi-agents pour implémenter les parties informationnelles des holons de surveillance sous formes d'agents logiciels. Ces agents peuvent être supportés par un ou plusieurs systèmes matériels.

La mise en œuvre de l'architecture EMH proposée dans ce chapitre est appliquée au domaine ferroviaire dans le chapitre suivant.

Chapitre V

Application au domaine ferroviaire

Introduction

Ce chapitre présente une application réelle de l'architecture de surveillance EMH proposée dans le chapitre III. Les holons de surveillance qui composent cette architecture répondent aux différents comportements définis dans le chapitre IV. Dans ce chapitre, cette architecture est appliquée au domaine ferroviaire dans le cadre d'un projet conjoint Industrie-Recherche nommé SURFER (*SURveillance active FERroviaire*). Le contexte industriel de ce projet est présenté dans la première partie de ce chapitre. La deuxième partie de ce chapitre applique l'architecture de surveillance holonique proposée à un système de transport ferroviaire. Dans la quatrième partie, une plateforme expérimentale conçue pour mettre en œuvre l'approche de surveillance holonique proposée est présentée. La dernière partie de ce chapitre présente les principaux résultats obtenus.

1. Contexte industriel du projet SURFER

Le projet SURFER est piloté par Bombardier-Transport en collaboration avec le laboratoire TEMPO EA4542 (Thermique Ecoulement Mécanique Matériaux Mise en Forme Production) - équipe PSI (Production, Service, Information), l'IFSTTAR (Institut Français des Sciences et Technologies des Transports de l'Aménagement et des Réseaux) ainsi que deux entreprises valenciennoises : la société PROSYST et le groupe HIOLLE INDUSTRIES. Ce projet lancé en juin 2010 pour une durée de trois ans est financé par le FUI (Fonds Unique Interministériel), la région Nord-Pas-de-Calais et labellisé par les pôles de compétitivités *i-Trans* et *Advancity*.

1.1 Objectifs du projet SURFER

Le projet SURFER vise à accroitre la disponibilité des matériels roulants ferroviaires et à optimiser les tâches de maintenance des trains équipés d'un système de surveillance active SURFER. Bombardier-Transport est concerné en premier lieu par une meilleure disponibilité de ses produits en exploitation et par une réduction du coût global de possession. Ce projet apporte également au constructeur une aide à la fiabilisation de ses produits grâce à une surveillance avancée dès le milieu de leur cycle de vie (mise en service et exploitation de nouveaux trains). Les défaillances entrainant une immobilisation des trains en exploitation sont des événements redoutés par les constructeurs de matériels ferroviaires et leurs exploitants (insatisfaction des usagers, dégradation de l'image de marque, risques de pénalités, par exemple). Ainsi, le projet SURFER cible la disponibilité des équipements embarqués à bord de trains de passagers, avec comme objectif de concevoir, développer et mettre au point en situation réelle les fonctions suivantes :

- Surveillance embarquée : acquisition de signaux, prétraitement, diagnostic de défaillances et diagnostic prédictif de pannes,

- Aide aux personnels de maintenance : actions de maintenance prédictive et corrective,
- Optimisation dynamique des plans de maintenance : suivi de l'état de santé des équipements embarqués sur l'ensemble d'une flotte de trains.

L'approche de surveillance SURFER s'inscrit dans une démarche de diagnostic embarqué, nécessitant de traiter les données au plus près des équipements surveillés pendant leur fonctionnement (surveillance *en-ligne*). Les exploitants de matériels ferroviaire souhaitant améliorer la disponibilité de leurs flottes de trains et optimiser les coûts de maintenance peuvent ainsi exiger le déploiement de la solution SURFER sur de nouveaux investissements (conception de nouveaux trains) ou équiper leurs flottes actuelles à posteriori (réaménagement ou *retrofit* de trains existants). La problématique majeure est de passer d'une approche de surveillance centralisée, où le diagnostic est réalisé à distance, à une approche de surveillance active embarquée, où le diagnostic est implanté à bord du système train (figure 5.1).

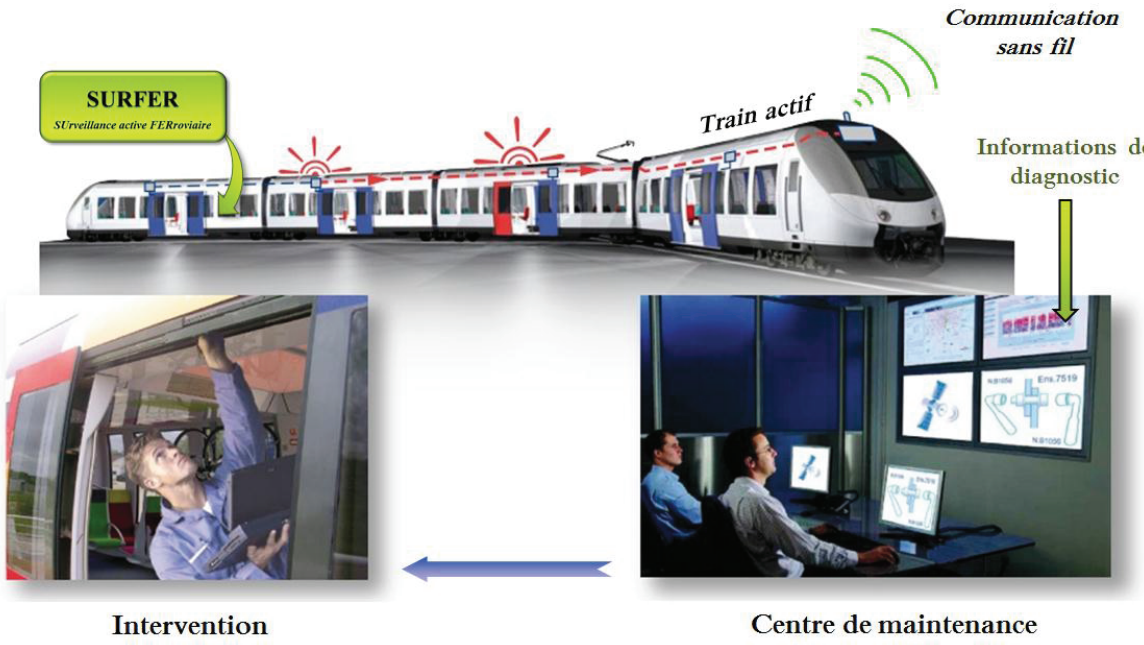

Figure 5.1. Approche de surveillance active embarquée SURFER.

Le cas d'étude retenu dans ce projet est la surveillance des accès voyageurs d'un train de passagers conçu par Bombardier-Transport. Les accès voyageurs sont des équipements critiques d'un train, pouvant entrainer des perturbations du trafic et des interruptions du service ferroviaire. Il est estimé que 30 à 40 % des défaillances d'un train en service commercial sont causées par les accès voyageurs (Cauffriez et al., 2013). Les systèmes à diagnostiquer dans cette étude sont donc les portes d'accès de trains de passagers fournies par des équipementiers du secteur ferroviaire.

1.2 Utilisateurs du système SURFER

Le système SURFER élabore des informations de surveillance destinées à différents utilisateurs. La figure 5.2 ci-dessous présente les principaux services rendus par cette innovation aux différents acteurs de l'industrie ferroviaire.

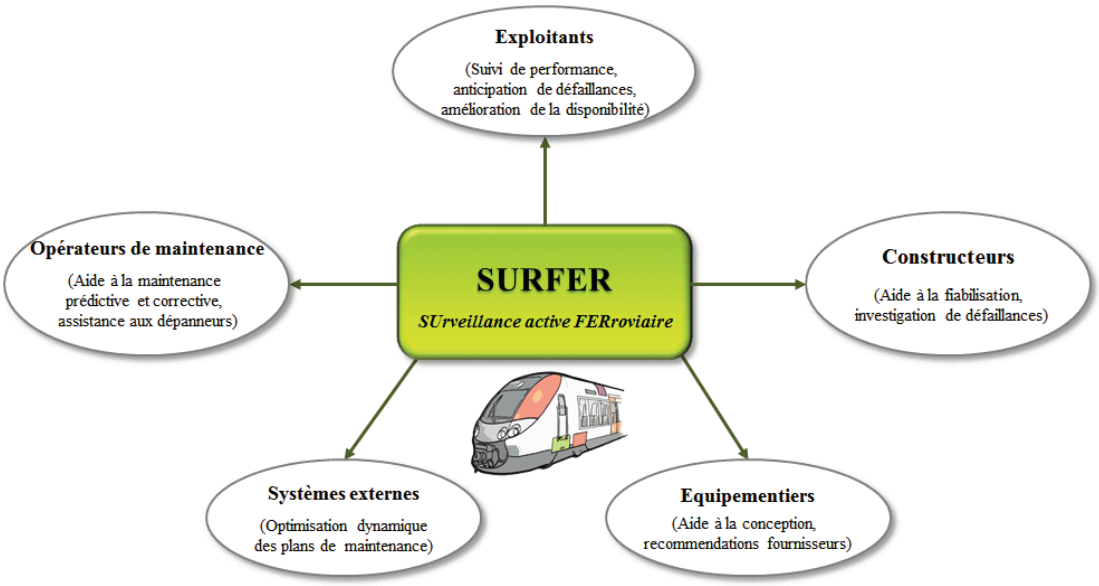

Figure 5.2. Utilisateurs de l'innovation SURFER et principaux services rendus.

Le système SURFER délivre des informations pertinentes aux exploitants de matériels ferroviaires et aux sous-traitants de maintenance, leur permettant de réduire l'indisponibilité des matériels ferroviaires (actions de maintenance préventive et corrective à réaliser, indicateurs de performance des équipements surveillés, par exemple). Ces informations peuvent également alimenter des systèmes externes de maintenance reliés au système SURFER (outil d'aide à la décision de maintenance, service de commande de pièces de rechanges, par exemple). Par ailleurs, les constructeurs de matériels ferroviaires, notamment Bombardier-transport en tant que chef de file, bénéficient de données de surveillance SURFER pour comprendre l'origine de certaines défaillances en exploitation et accroitre ainsi la fiabilité de leurs produits. Les analyses approfondies menées grâce à ces données peuvent être transmises aux équipementiers concernés pour déterminer les mesures correctives nécessaires (modification de la conception d'un équipement, exigences supplémentaires à satisfaire, par exemple) et faciliter la résolution de litiges entre équipementiers, constructeurs et exploitants. Les informations de surveillance élaborées par SURFER n'ont pas vocation à être présentées aux personnels à bord des trains, dont les actions sont limitées et réglementées (agent de conduite, par exemple). Notons que les différents services rendus par SURFER nécessitent des données fournies par les acteurs impliqués dans le développement de ce système (connaissance exploitée pour élaborer le diagnostic à bord du système train, mise à jour de cette connaissance, paramétrage des fonctionnalités du système, par exemple).

La partie suivante se propose d'appliquer l'architecture de surveillance EMH proposée pour un système train dans le cadre du projet SURFER.

2. Application de notre proposition pour un système train

Dans cette partie, notre proposition est appliquée à la surveillance des accès voyageurs d'un train de passagers. La section 2.1 instancie l'architecture de surveillance holonique proposée pour un système train. La section 2.2 présente les algorithmes implémentant les comportements des holons pour la surveillance des accès voyageurs d'un train. La section 2.3

illustre les comportements dynamiques des holons de surveillance à travers différents scénarios de diagnostic.

2.1 Instanciation de l'architecture holonique de surveillance

Un système train est un système cible mobile, noté Σ_i sur la figure 5.3, constitué de plusieurs véhicules ($\Sigma_{i/j}$, $\Sigma_{i/j'}$). Chaque véhicule est composée d'un ensemble de sous-systèmes, dont les sous-systèmes d'accès voyageurs, implantés de chaque côté d'un véhicule ($\Sigma_{i/j/k}$, $\Sigma_{i/j/k'}$, par exemple). Le système train, comme chaque véhicule et sous-système, est associée à un contexte spécifique (C_i, $C_{i/j/k}$, $C_{i'/j'/k'}$, par exemple).

Un sous-système d'accès voyageurs se compose d'une Unité de Contrôle Porte (partie contrôle ou UCP) et d'une partie opérative (partie sous-contrôle). La partie opérative regroupe deux vantaux coulissant à l'extérieur du véhicule, dont le mouvement de translation est assuré par un motoréducteur accouplé à une vis sans fin. De plus, cette partie opérative comprend une ou deux marches mobiles suivant le type de véhicule dans lequel l'accès voyageurs est implanté. Un accès voyageurs situé dans un véhicule d'extrémité du train comporte deux marches mobiles motorisées. La première nommée comble-lacune-UFR (Usagers en Fauteuil Roulant) se déploie au niveau du plancher du véhicule lorsque le quai de la gare ferroviaire est dit "haut". La seconde nommée marche-PMR (Personnes à Mobilité Réduite) facilite l'accès aux usagers lorsque le quai de la gare ferroviaire est dit "bas". Un accès voyageurs situé dans un véhicule intermédiaire comporte une seule marche-PMR.

Les accès voyageurs sont commandés suivant la structure hiérarchisée de contrôle du système train. Les accès voyageurs situés dans un véhicule reçoivent des signaux de commande du système de contrôle centralisé de ce véhicule. Le système de contrôle centralisé du train impose des signaux de commande aux différents systèmes de contrôle des véhicules qui composent le train.

Le système de contrôle centralisé du train, le système de contrôle centralisé de chaque véhicule, comme chaque sous-système d'accès voyageurs est équipé d'un holon de surveillance (H_i, $H_{i/j}$, $H_{i'/j/k}$). Suivant la décomposition hiérarchique du système train retenue, le holon relatif au système de contrôle centralisé du train possède un niveau hiérarchique supérieur aux holons associés aux systèmes de contrôle centralisé des véhicules. De la même manière, les holons associés aux sous-systèmes d'accès voyageurs présentent un niveau hiérarchique inférieur aux holons associés aux systèmes de contrôle centralisé des véhicules. Par conséquent, conformément aux niveaux de décomposition du système train, trois types de holons sont définis.

- Holon de surveillance porte : ce holon élémentaire est associé à un sous-système d'accès voyageurs implanté dans un véhicule du train. Il élabore des informations de diagnostic relatives à cet accès voyageurs grâce à une méthode de diagnostic. Par ailleurs, ce holon exploite des informations locales de contexte et peut coopérer avec d'autres holons de surveillance porte. Les informations de diagnostic élaborés par ce holon sont diffusés au holon véhicule "père" dans lequel il s'inscrit.

- Holon de surveillance véhicule : ce holon composé est associé à un véhicule du train. Il contextualise les résultats de diagnostic des holons de surveillance des portes de ce véhicule et coopère avec d'autres holons de surveillance des véhicules. Ce holon transmet des informations contextualisées de diagnostic au holon de surveillance train.

- Holon de surveillance train : ce holon composé est associé au système train. Il contextualise les informations transmises par les holons de surveillance des véhicules du train et délivre un diagnostic global du train.

Au niveau hiérarchique le plus bas (niveau porte), la fonction de surveillance d'un holon est associée à un sous-système d'accès voyageurs. Cette fonction implémente les trois modules définis dans le chapitre précédent pour élaborer un diagnostic (figure 5.3).

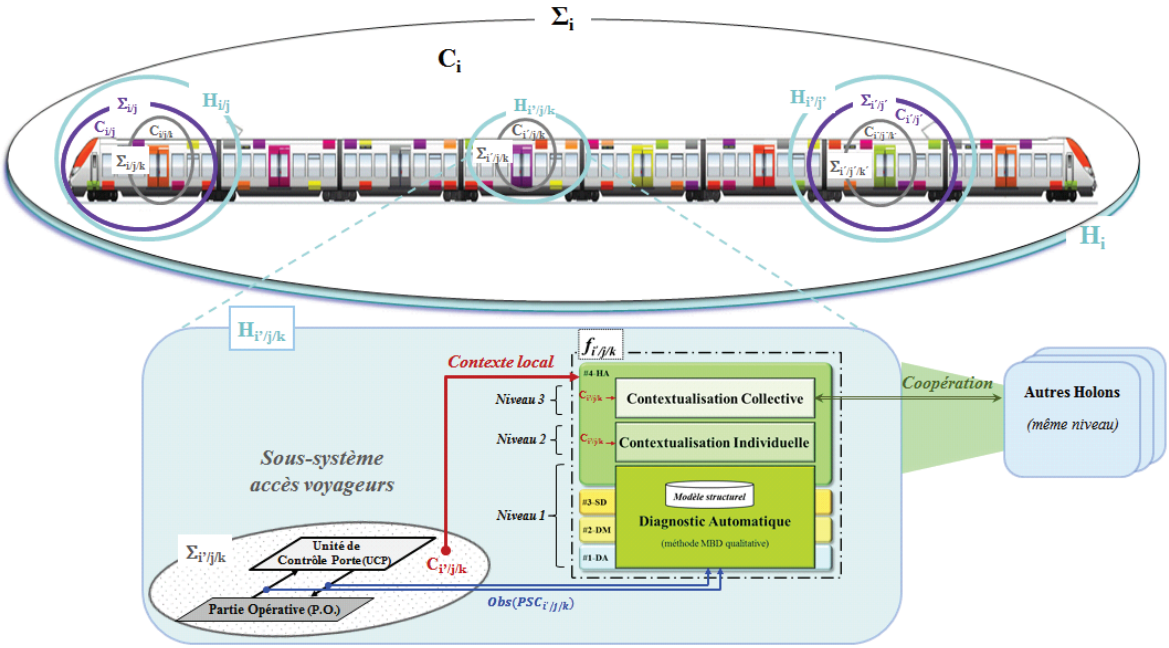

Figure 5.3. Architecture de surveillance holonique proposée appliquée à un système train.

Par exemple, sur la figure 5.3, le holon de surveillance porte $H_{i'/j/k}$ est formé par l'association de la fonction de surveillance $f_{i'/j/k}$ et du sous-système accès voyageurs $\Sigma_{i'/j/k}$ opérant dans le contexte $C_{i'/j/k}$. Cette fonction de surveillance implémente une méthode de diagnostic à base de modèle qualitatif nommée diagnostic automatique (diagnostic de niveau 1), ainsi que deux modules de contextualisation du diagnostic. Le module de contextualisation individuelle repose sur un processus d'inhibition (diagnostic de niveau 2). Le module de contextualisation collective est basé sur un processus d'inhibition étendue (diagnostic de niveau 3).

Les holons de surveillance des véhicules et du train implémentent uniquement un module de contextualisation collective.

La section suivante présente les algorithmes implémentant les comportements de ces trois types de holons.

2.2 Algorithmes des holons de surveillance

Cette section présente les comportements des holons de surveillance élémentaires au niveau porte et des holons de surveillance composés au niveau véhicule et au niveau train sous formes d'algorithmes.

2.2.1 Algorithme des holons de surveillance des portes

Comme évoqué précédemment, un holon de surveillance porte élabore un diagnostic de niveau 1 grâce à une méthode de diagnostic à base de modèle. La méthode utilisée nécessite un modèle structurel de la partie opérative porte. Le diagnostic délivré par cette méthode est ensuite contextualisé de manière individuelle et collective. La contextualisation du diagnostic requiert deux bases de règles distinctes. La première, nommée table d'inhibition individuelle, est exploitée par le module de contextualisation individuelle de ce holon (diagnostic de niveau 2). Cette table contient des événements de diagnostic de niveau 1 à inhiber suivant des variables de contexte locales de l'accès voyageurs surveillé. Un exemple de table d'inhibition exploitée par un holon de surveillance porte $H_{i/j/k}$ est donné par la figure 5.4 ci-après. Une valeur "VRAI" inscrite dans cette table indique que le holon doit inhiber un événement de diagnostic pour un contexte local donné.

Holon de surveillance porte $H_{i/j/k}$		**Contexte** $C_{i/j/k}$	
		Obstacle porte	Dispositif de secours actionné
Evénement diagnostic $Diag(PSC_{i/j/k})$	Evénement non prévu entrée capteur S1	VRAI	VRAI
	Absence événement relecture relais K1	FAUX	VRAI
	Absence événement entrée capteur S2	VRAI	FAUX

Figure 5.4. Exemple de table d'inhibition individuelle d'un holon de surveillance porte.

La seconde base de règles, nommée table d'inhibition étendue, est utilisée par le module de contextualisation collective d'un holon de surveillance porte. Cette table définit les événements de diagnostic de niveau 1 à inhiber suivant un contexte enrichi, obtenu grâce à une coopération entre les holons de surveillance portes. La figure 5.5 présente un exemple de table utilisée par un holon de surveillance porte $H_{i/j/k}$.

Holon de surveillance porte $H_{i/j/k}$		**Contexte enrichi** $C_{Ei/j/k}$	
		Charge passagers importante	Porte opposée non alimentée
Evénement diagnostic $Diag(PSC_{i/j/k})$	Evénement non prévu entrée capteur S1	FAUX	FAUX
	Absence événement relecture relais K1	FAUX	VRAI
	Absence événement entrée capteur S2	VRAI	VRAI

Figure 5.5. Exemple de table d'inhibition étendue d'un holon de surveillance porte.

Le comportement d'un holon de surveillance porte est donné par l'algorithme 5.1. Dans cet algorithme, les mécanismes d'échanges entre holons nécessaires à l'élaboration d'un contexte enrichi et implémentés par le processus d'inhibition étendue ne sont pas représentés.

Algorithme 5.1 Comportement d'un holon de surveillance porte $H_{i/j/k}$

Entrées : observations de la partie opérative porte $Obs(PSC_{i/j/k})$, contexte local de la porte $C_{i/j/k}$, contexte enrichi $C_{Ei/j/k}$.
Sortie : diagnostic contextualisé niveau porte $D_{i/j/k}$
Connaissance : Modèle structurel de la partie opérative porte $M_{i/j/k}$, table d'inhibition individuelle porte $Tab_Ind_{i/j/k}$, table d'inhibition étendue porte $Tab_Ext_{i/j/k}$.
Variables : Inhibition_Ind : booléenne, Inhibition_Ext : booléenne.

Début
Initialisation de la méthode de diagnostic avec le modèle $M_{i/j/k}$
Chargement de la table d'inhibition individuelle porte $Tab_Ind_{i/j/k}$
Chargement de la table d'inhibition étendue porte $Tab_Ext_{i/j/k}$
Inhibition_Ind ← FAUX
Inhibition_Ext ← FAUX
Tant que VRAI
Surveillance de la partie opérative porte et attente d'un événement de diagnostic $Diag(PSC_{i/j/k})$
Si $Diag(PSC_{i/j/k})$ **Alors**
Acquisition du contexte local $C_{i/j/k}$
/* Exécution du processus d'inhibition avec la table $Tab_Ind_{i/j/k}$ */
Inhibition_Ind ← *Contextualisation_Individuelle*($Diag(PSC_{i/j/k})$)
Si Inhibition_Ind = VRAI **Alors**
Inhibition de $Diag(PSC_{i/j/k})$
Inhibition_Ind ← FAUX
Sinon
Elaboration du contexte enrichi $C_{Ei/j/k}$
/* Exécution du processus d'inhibition étendue avec la table $Tab_Ext_{i/j/k}$ */
Inhibition_Ext ← *Contextualisation_Collective*($Diag(PSC_{i/j/k})$)
Si Inhibition_Ext = VRAI **Alors**
Inhibition de $Diag(PSC_{i/j/k})$
Inhibition_Ext ← FAUX
Sinon
$D_{i/j/k}$ ← $Diag(PSC_{i/j/k})$
Propagation du diagnostic $D_{i/j/k}$ au holon de surveillance véhicule "père" $H_{i/j}$
Fin Si
Fin Si
Fin Si
Fin Tant que
Fin

Lorsqu'un événement de diagnostic relatif à la partie opérative de la porte apparait, le holon exécute tout d'abord le processus d'inhibition (*Contextualisation_Individuelle*) qui analyse la table d'inhibition individuelle. Dans le cas où cet événement de diagnostic n'est pas inhibé par le processus d'inhibition, le holon exécute alors le processus d'inhibition étendue, qui explore la table d'inhibition étendue et dialogue avec d'autres holons (*Contextualisation_Collective*). Enfin, lorsque cet événement n'est pas inhibé par ce second processus, le diagnostic est confirmé par le holon de surveillance porte. Ce diagnostic est transmis au holon véhicule de niveau supérieur.

2.2.2 Algorithme des holons de surveillance des véhicules

Chaque holon de surveillance véhicule implémente un module de contextualisation collective qui exploite une table d'inhibition étendue. Ainsi, ce holon contextualise de manière collective les informations transmises par les holons de surveillance des portes.

Un exemple de table d'inhibition étendue utilisée par un holon véhicule $H_{i/j}$ est présenté ci-après (figure 5.6).

Holon de surveillance véhicule $H_{i/j}$		**Contexte enrichi** $C_{Ei/j}$	
		Inclinaison du véhicule importante	Passage d'un tunnel ferroviaire
Evénement diagnostic $D^*_{i/j/k}$	Evénement non prévu entrée capteur S1	FAUX	VRAI
	Absence événement relecture relais K1	FAUX	FAUX
	Absence événement entrée capteur S2	VRAI	FAUX

Figure 5.6. Exemple de table d'inhibition étendue d'un holon de surveillance véhicule.

Le comportement d'un holon de surveillance véhicule consiste à attendre des événements de diagnostic transmis par des holons de surveillance des portes et à exécuter le processus d'inhibition étendue pour chaque événement de diagnostic reçu. L'algorithme 5.2 ci-après décrit le comportement d'un holon de surveillance véhicule.

Algorithme 5.2 Comportement d'un holon de surveillance véhicule $H_{i/j}$

Entrées : événement de diagnostic transmis par les holons de surveillance des portes $D^*_{i/j/k}$, contexte local du véhicule $C_{i/j}$, contexte enrichi $C_{Ei/j}$.
Sortie : diagnostic contextualisé niveau véhicule $D_{i/j}$
Connaissance : table d'inhibition étendue véhicule $Tab_Ext_{i/j}$.
Variables : Inhibition_Ext : booléenne.

Début
  Chargement de la table d'inhibition étendue véhicule $Tab_Ext_{i/j/k}$
  Inhibition_Ext ← FAUX
Tant que VRAI
  Attente d'un événement de diagnostic de holons de surveillance des portes $D^*_{i/j/k}$
  Si $D^*_{i/j/k}$ **Alors**
    Acquisition du contexte local $C_{i/j/k}$
    Elaboration du contexte enrichi $C_{Ei/j/k}$
    /* Exécution du processus d'inhibition étendue avec la table $Tab_Ext_{i/j}$ */
    Inhibition_Ext ← *Contextualisation_Collective*($D^*_{i/j/k}$)
    Si Inhibition_Ext = VRAI **Alors**
      Inhibition de $D^*_{i/j/k}$
      Inhibition_Ext ← FAUX
    Sinon
      $D_{i/j}$ ← $D^*_{i/j/k}$
      Propagation du diagnostic $D_{i/j}$ au holon de surveillance train H_i
    Fin si
  Fin si
Fin Tant que
Fin

Un événement de diagnostic non-inhibé par le processus de contextualisation collective d'un holon véhicule est propagé au holon de surveillance train, dont le comportement est détaillé dans la sous-section suivante.

2.2.3 Algorithme du holon de surveillance train

Le comportement d'un holon de surveillance train est similaire à celui d'un holon de surveillance véhicule. En effet, un holon de surveillance train exécute également un processus d'inhibition étendue. Cependant, la table d'inhibition étendue au niveau train est restreinte aux variables de contexte du système train. Par conséquent, le contexte enrichi comprend uniquement des variables de contextes observées par le holon de surveillance train. La figure 5.7 présente un exemple de table d'inhibition étendue associée à un holon de surveillance train.

Holon de surveillance train H_i		**Contexte train** C_i	
		Train en maintenance	Train accouplé mené
Evénement diagnostic $D^*_{i/j}$	Evénement non prévu entrée capteur S1	VRAI	VRAI
	Absence événement relecture relais K1	VRAI	FAUX
	Absence événement entrée capteur S2	VRAI	FAUX

Figure 5.7. Exemple de table d'inhibition étendue d'un holon de surveillance train.

Un holon de surveillance train réceptionne les informations de diagnostic des holons de surveillance des véhicules et exécute le processus d'inhibition étendue pour chaque événement de diagnostic reçu. Le comportement de ce holon est donné par l'algorithme 5.3.

Algorithme 5.3 Comportement d'un holon de surveillance train H_i

Entrées : événement de diagnostic transmis par les holons de surveillance des véhicules $D^*_{i/j}$, contexte du système train C_i.
Sortie : diagnostic contextualisé niveau train D_i
Connaissance : table d'inhibition étendue train Tab_Ext_i.
Variables : Inhibition_Ext : booléenne.

Début
  Chargement de la table d'inhibition étendue train $Tab_Ext_{i/j/k}$
  Inhibition_Ext ← FAUX
Tant que VRAI
  Attente d'un événement de diagnostic de holons de surveillance des véhicules $D^*_{i/j}$
  Si $D^*_{i/j}$ **Alors**
    Acquisition du contexte local C_i
    /* Exécution du processus d'inhibition étendue avec la table Tab_Ext_i */
    Inhibition_Ext ← *Contextualisation_Collective*($D^*_{i/j}$)
    Si Inhibition_Ext = VRAI **Alors**
      Inhibition de $D^*_{i/j}$
      Inhibition_Ext ← FAUX
    Sinon
      D_i ← $D^*_{i/j}$
      Transmettre information de diagnostic D_i
    Fin si
  Fin si
Fin Tant que
Fin

Un événement de diagnostic non-inhibé par un holon de surveillance train représente une information de diagnostic pertinente. Cette information doit être transmise aux utilisateurs du système de surveillance. En effet, cet événement a parcouru une succession de filtres hiérarchiques qui confirment l'occurrence d'une défaillance avérée.

La section suivante propose d'illustrer les comportements dynamiques des holons de surveillance définis précédemment à travers trois scénarios de diagnostic.

2.3 Exemples de scénarios de diagnostic

Dans cette section, les holons implémentés par les algorithmes présentés auparavant sont animés suivant trois scénarios de diagnostic. Pour ces trois scénarios, un holon de surveillance train (H_1) composé de deux holons de surveillance véhicule ($H_{1/1}$, $H_{1/2}$), eux-mêmes constitués de deux holons de surveillance porte sont considérés. Les holons de surveillance des portes du premier véhicule sont notés $H_{1/1/1}$ et $H_{1/1/2}$ et les holons de surveillance des portes du second véhicule sont notés $H_{1/2/1}$ et $H_{1/2/2}$. Le formalisme utilisé pour représenter l'activité des holons et leurs interactions est le diagramme de séquences introduit dans le chapitre précédent.

2.3.1 Scénario #1 : diffusion d'un diagnostic contextualisé

Ce scénario illustre les comportements des holons de surveillance lorsqu'une défaillance avérée survient dans la partie opérative d'un sous-système accès voyageurs. Le diagramme de séquences de la figure 5.8 présente les comportements des holons pour ce scénario. En raison d'une défaillance d'un capteur de la partie opérative porte $PSC_{1/1/1}$ noté s1, le holon $H_{1/1/1}$ élabore un diagnostic de niveau 1 grâce à sa méthode de diagnostic à base de modèle. Cet événement de diagnostic n'est pas expliqué par le contexte local $C_{1/1/1}$ de la porte et ne fait l'objet d'une coopération avec d'autres holons de surveillance porte. De ce fait, le holon $H_{1/1/1}$ transmet cet événement de diagnostic au holon de surveillance véhicule $H_{1/1}$ (message repéré 1 sur la figure 5.8). Suite à la réception de cet événement, le holon $H_{1/1}$ exécute alors le processus d'inhibition étendue. Dans ce scénario, il est supposé que cet événement soit présent dans la table d'inhibition étendue associée au holon $H_{1/1}$ et qu'une coopération entre les holons $H_{1/1}$ et $H_{1/2}$ s'avère nécessaire (table d'inhibition présentée sur la figure 5.6, par exemple). Par conséquent, le holon $H_{1/1}$ demande des informations de contexte au holon $H_{1/2}$ (message de requête repéré 2). Les informations transmises par le holon $H_{1/2}$ (message d'information repéré 3) ne permettent pas au holon $H_{1/1}$ d'inhiber cet événement suivant un contexte enrichi. Le holon $H_{1/1}$ propage ainsi cet événement de diagnostic au holon de surveillance train H_1 (message d'information repéré 4).

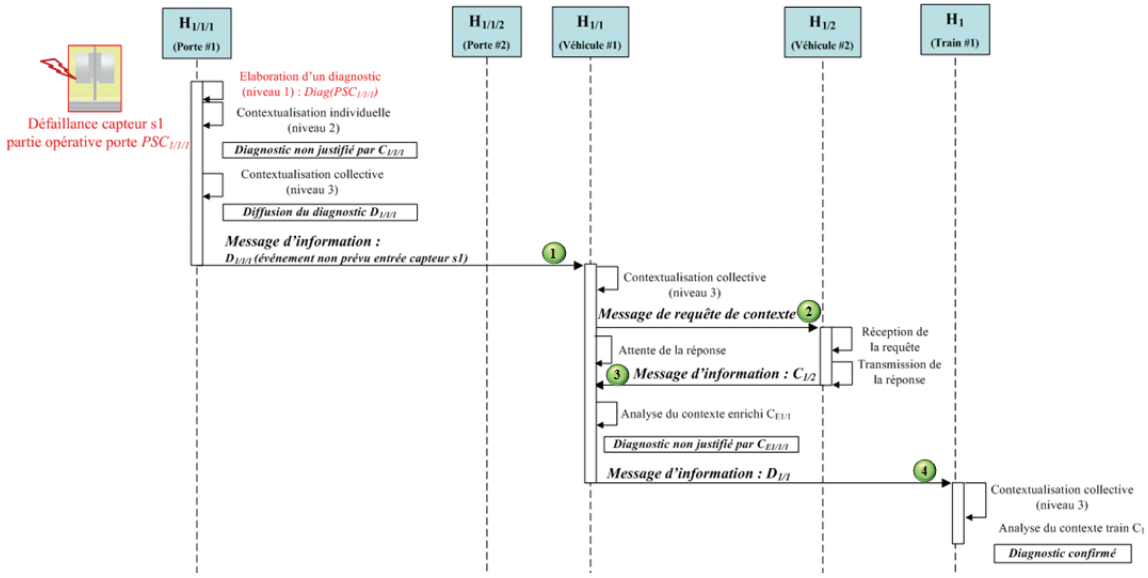

Figure 5.8. Diagramme de séquences du scénario #1.

Après réception de cet événement de diagnostic, le holon de surveillance train exécute son processus d'inhibition étendue et confirme alors cette information de diagnostic. En effet, le contexte du système train C_1 n'apporte aucune information supplémentaire qui permettrait au holon de surveillance train de justifier l'apparition de cet événement. Ce premier scénario permet de mettre en évidence la contextualisation d'un événement de diagnostic et la propagation successive de cet événement qui donne lieu à une information pertinente de diagnostic.

2.3.2 Scénario #2 : Elimination de fausses alarmes

Dans ce scénario, un holon de surveillance véhicule permet d'inhiber des événements de diagnostic erronés émanant des holons de surveillance des portes. En raison d'une inclinaison importante de l'un des véhicules du train (dévers ferroviaire), les accès voyageurs situés dans ce véhicule présentent le même symptôme lors de la fermeture des portes (retard à l'activation d'un capteur noté s2). La figure 5.9 illustre les comportements des holons pour ce scénario. Les holons $H_{1/1/1}$ et $H_{1/1/2}$ élaborent un diagnostic de niveau 1 suite à l'apparition de ce symptôme. Les contextes associés à ces holons ne permettent pas de justifier ces événements de diagnostic. Etant donné que ces événements ne font pas l'objet d'une contextualisation collective, les diagnostics contextualisés $D_{1/1/1}$ et $D_{1/1/2}$, élaborés respectivement par les holons $H_{1/1/1}$ et $H_{1/1/2}$, sont transmis au holon de surveillance véhicule $H_{1/1}$ (messages d'information repérés 1 et 2 sur la figure 5.9). Chacun de ces événements déclenche le processus d'inhibition étendue interne au holon de surveillance véhicule $H_{1/1}$.

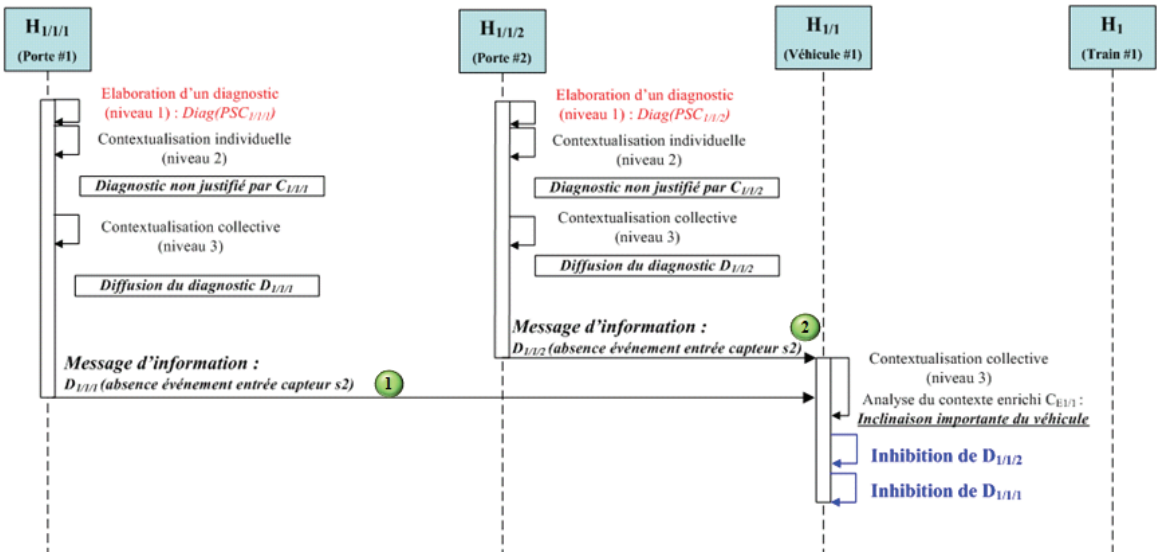

Figure 5.9. Diagramme de séquences du scénario #2.

Une information de contexte locale détenue par le holon $H_{1/1}$ permet de justifier la présence de ces deux événements de diagnostic (inclinaison importante du véhicule). Ainsi, ce holon de surveillance véhicule inhibe les événements de diagnostic $D_{1/1/1}$ et $D_{1/1/2}$ et ne transmet aucune information de diagnostic au holon de surveillance train. Ce second scénario permet d'illustrer l'efficacité de notre approche de surveillance récursive pour éliminer les erreurs de diagnostic (fausses alarmes).

2.3.3 Scénario #3 : coopération entre holons de surveillance

Ce scénario illustre une contextualisation collective du diagnostic réalisée par les holons de surveillance véhicule $H_{1/1}$ et $H_{1/2}$. Dans ce scénario, le train entre dans un tunnel ferroviaire, ce qui provoque des vibrations anormales des accès voyageurs. Les holons de surveillance des portes $H_{1/1/1}$, $H_{1/1/2}$, $H_{1/2/1}$ et $H_{1/2/2}$ détectent une anomalie relative à un capteur noté s1 de la partie opérative porte et élaborent un diagnostic de niveau 1. Les informations de contexte locales de ces holons n'apportent aucune explication plausible sur l'apparition de ces événements. Par conséquent, les diagnostics contextualisés $D_{1/1/1}$ et $D_{1/1/2}$ élaborés respectivement par les holons $H_{1/1/1}$ et $H_{1/1/2}$ sont transmis au holon de surveillance véhicule $H_{1/1}$. De la même manière, les diagnostics $D_{1/2/1}$ et $D_{1/2/2}$ sont transmis respectivement par les holons $H_{1/1/1}$ et $H_{1/1/2}$ au holon $H_{1/2}$. Le diagramme de séquences de ce scénario est présenté sur la figure 5.10.

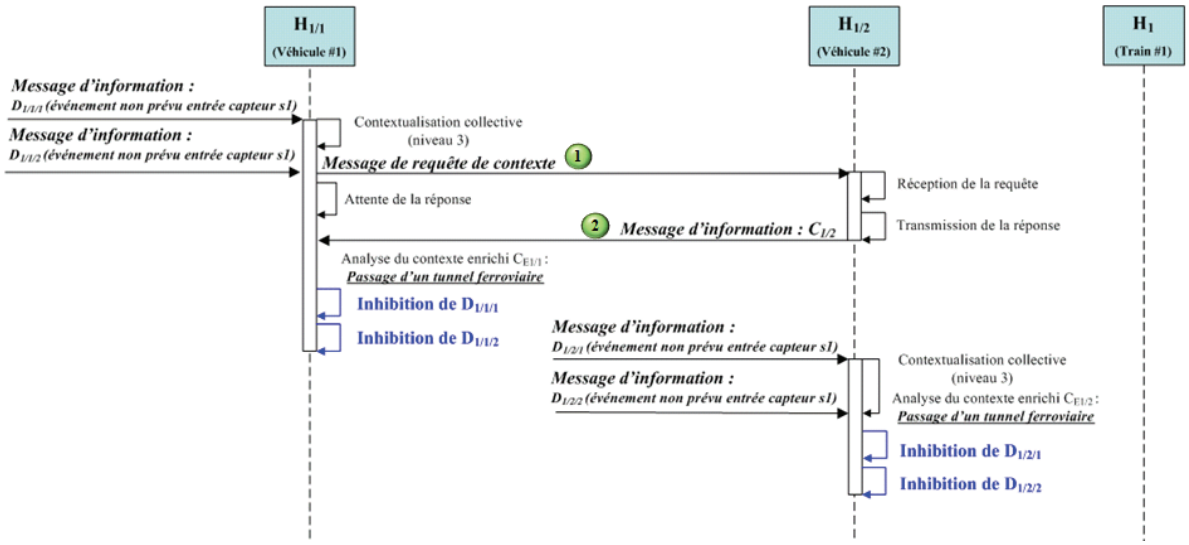

Figure 5.10. Diagramme de séquences du scénario #3.

Il est supposé que ces événements de diagnostic soient justifiés par une information de géolocalisation détenue uniquement par le holon $H_{1/2}$ (véhicule située dans un tunnel ferroviaire). De ce fait, le holon $H_{1/1}$ coopère avec le holon $H_{1/2}$ pour obtenir cette information de géolocalisation. Le contexte enrichi obtenu par le holon $H_{1/1}$ permet alors de justifier les événements de diagnostic $D_{1/1/1}$ et $D_{1/1/2}$. Ces événements sont donc inhibés par le processus d'inhibition étendue interne au holon $H_{1/1}$. Le holon $H_{1/2}$ dispose localement de cette information de contexte, ce qui lui permet d'inhiber les événements $D_{1/2/1}$ et $D_{1/2/2}$ sans coopérer avec le holon $H_{1/1}$. Ce troisième scénario démontre la capacité d'un holon à coopérer avec un autre holon de même niveau pour enrichir ses informations de contexte et limiter ainsi la propagation de fausses alarmes.

Les trois scénarios de diagnostic exposés précédemment peuvent être implémentés dans un environnement de simulation afin de valider l'approche de surveillance holonique proposée. La plateforme expérimentale utilisée pour mettre en œuvre les différents holons de surveillance est détaillée dans la partie suivante.

3. Plateforme expérimentale

Cette partie présente la plateforme expérimentale conçue dans le cadre du consortium SURFER. Cette plateforme dénommée PEMAS (Plateforme Expérimentale Multi-Agents Surfer) a été mise au point au laboratoire TEMPO pour implémenter les différents holons de l'architecture de surveillance EMH proposée. Au sein de cette plateforme, la mise en œuvre des holons de surveillance par des agents logiciels et la validation des scénarios présentés dans la section 2.3 ont été réalisés par deux collègues de l'équipe PSI recrutés depuis le 1er septembre 2011 dans le cadre du projet SURFER (J. Clarhaut - maître de conférences et R. Bekrar - ingénieur de recherche).

Les principaux objectifs de la plateforme PEMAS sont les suivants :

- développer et de valider les algorithmes des holons qui composent l'architecture de surveillance holonique d'un système train avant de les implanter dans des trains réels en exploitation,
- tester les comportements des holons grâce à différents scénarios de diagnostic,
- expérimenter différentes configurations de l'architecture de surveillance holonique du système train (allocation des fonctions logicielles sur des dispositifs matériels), afin d'évaluer la fiabilité de l'architecture,
- mesurer la performance du diagnostic (taux de fausses alarmes).

La section 3.1 présente l'architecture matérielle et logicielle de la plateforme PEMAS. La section 3.2 détaille l'architecture des agents logiciels implémentés dans cette plateforme et décrit un comportement de l'agent contextuel implémenté.

3.1 Architecture de la plateforme PEMAS

La plateforme PEMAS peut être scindée en deux parties. La première partie a pour objectif de simuler le comportement du système train et de ses accès voyageurs. La seconde partie implémente l'architecture de surveillance holonique associée au système train. Dans cette plateforme expérimentale, un holon de surveillance train et quatre holons de surveillance porte sont implémentés. La plateforme PEMAS comprend ainsi cinq machines nommées PEMAS 1 à PEMAS 5. Chaque machine supporte la fonction de surveillance d'un holon. Les machines PEMAS 1 à 4 implémentent chacune une fonction de surveillance relative à un accès voyageur, tandis que la machine PEMAS 5 implémente la fonction de surveillance relative au système train. Cette machine permet également de présenter les résultats de diagnostic aux utilisateurs.

La génération de signaux représentatifs du comportement du train et des accès voyageurs est réalisée par une machine supplémentaire nommée PEMAS 6. Les différentes machines de la plateforme disposent de deux interfaces réseaux (liaison filaire). Ces machines sont interconnectées par le biais de deux réseaux de communication distincts. Le premier réseau assure une diffusion des signaux générés par la machine PEMAS 6 vers les machines PEMAS 1 à 5 (réseau de simulation). Le second réseau est dédié au diagnostic du système train et supporte les échanges entre les fonctions de surveillance (réseau de diagnostic). Ce réseau relie uniquement les machine PEMAS 1 à 5 entre elles. La topologie retenue pour ces deux réseaux est la topologie étoile. Un commutateur Ethernet (*switch*) nommé SW1 est utilisé pour le réseau de simulation et un commutateur nommé SW2 est utilisé pour le réseau de diagnostic. L'architecture matérielle de la plateforme PEMAS est présentée sur la figure 5.11 ci-après.

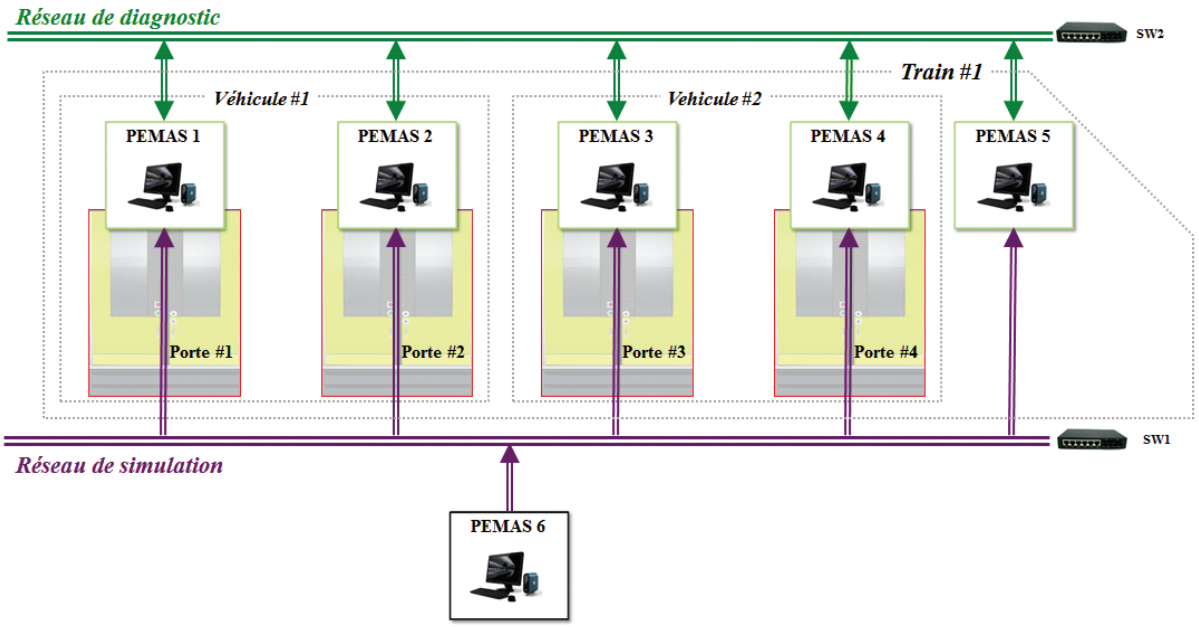

Figure 5.11. Architecture matérielle de la plateforme PEMAS.

Dans la plateforme PEMAS, les agents sont distribués sur les machines PEMAS 1 à 5. Chacune de ces machines représente une plateforme Jade dans laquelle les agents nécessaires à la mise en œuvre d'un holon résident. Par exemple, la machine PEMAS 1 supporte les agents ADPSC (Agent Diagnostic Partie Sous Contrôle) et AC (Agent Contextuel) relatifs à la porte #1 (figure 5. 12). Les agents peuvent s'échanger des messages ACL (*Agent Communication Language*) entre eux. Ces messages transitent par le réseau de diagnostic grâce à un protocole de transport de messages intégré à la plateforme Jade.

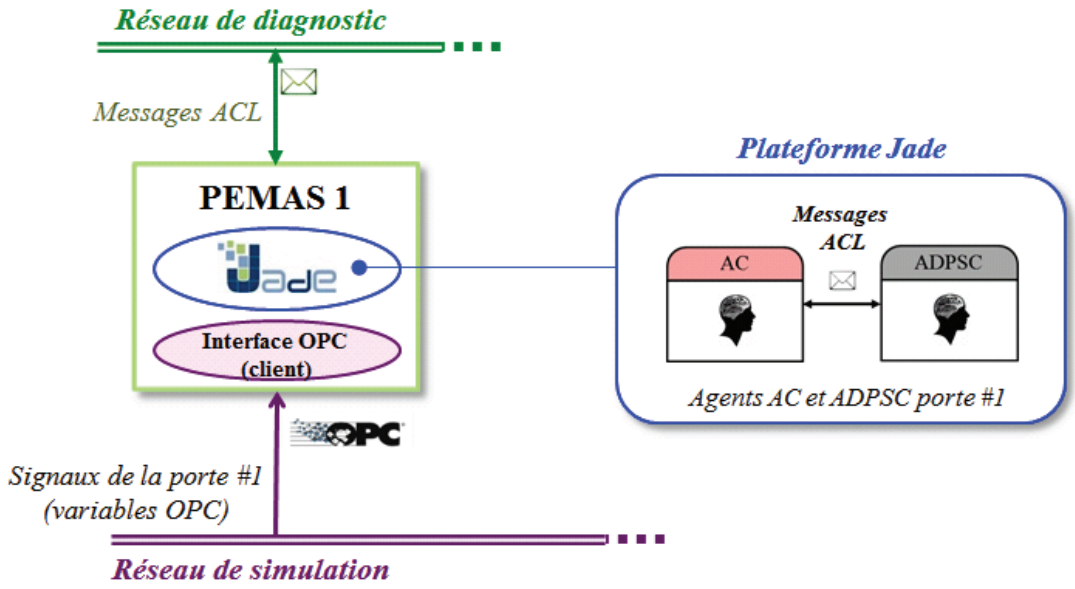

Figure 5.12. Principaux éléments logiciels de la plateforme PEMAS.

La technologie OPC (*Open Platform Communications*) est utilisée pour interfacer la machine PEMAS 6 génératrice de données avec les machines PEMAS 1 à 5 consommatrices de ces données (OPC Foundation, 2013). Les signaux bruts et informations de contextes nécessaires à l'élaboration des diagnostics sont diffusés par la machine PEMAS 6 sous forme de variables OPC. Ainsi, cette machine intègre un serveur OPC et les machines PEMAS 1 à 5 implémentent une interface OPC afin de réceptionner les données fournies par ce serveur. L'interface OPC choisie est implémentée en langage Java et compatible avec l'environnement Eclipse (JeasyOPC, 2013). Cette interface est utilisée en mode asynchrone : elle souscrit à un ensemble de variables fournies par le serveur et reçoit des notifications de la part du serveur lorsque ces variables sont disponibles. Les différents agents de la plateforme Jade peuvent instancier cette interface OPC (client) pour exploiter les données diffusées par le serveur. Par exemple, un agent de la machine PEMAS 1 peut recevoir des variables relatives à la porte #1 dont il a souscrit à travers cette interface.

3.2 Implémentation des agents dans Jade

Un agent est implémenté dans Jade par un seul processus léger (*thread*) codé en langage Java qui réalise un ensemble de tâches. Ces tâches sont regroupées en comportements (*behaviours*). Un agent peut exécuter un ou plusieurs comportements. L'exécution d'un comportement implique la réalisation d'un certain nombre d'actions qui doivent être accomplies par l'agent. Ainsi, les comportements implémentent les capacités et services rendus d'un agent. Cette section présente l'architecture interne des agents implémentés dans la plateforme Jade et précise la structure des messages échangés entre ces agents. Par la suite, un des comportements de l'agent contextuel (agent AC) est détaillé.

3.2.1 Architecture interne d'un agent

Un agent dispose d'un ensemble de comportements implémentés par son concepteur ou prédéfinis par la plateforme Jade. Par exemple, les comportements prédéfinis *SenderBehaviour* et *ReceiverBehaviour* peuvent être utilisés pour la réception et la transmission de messages ACL. Les comportements disponibles dans la bibliothèque de comportements d'un agent peuvent être actifs ou non (bloc repéré 1 sur la figure 5.13). L'activation d'un comportement entraine son exécution jusqu'à qu'à la dernière action qu'il comporte. Un comportement ne peut être interrompu par un autre comportement (comportements non-préemptifs). L'ordonnanceur interne à l'agent gère une liste de comportements actifs et prend en charge l'exécution des comportements de l'agent (bloc repéré 3). Lorsqu'un comportement en cours d'exécution est terminé, celui-ci est retiré de la liste des comportements actifs. Un agent possède un état interne qui peut évoluer au cours de sa vie. Cet état est géré par le gestionnaire de cycle de vie de l'agent (bloc repéré 4). Par exemple selon les spécifications FIPA (*Foundation for Intelligent Physical Agents*), un agent à l'état *ACTIVE* est opérationnel et peut accéder aux différents services de Jade. En revanche, lorsqu'un agent est dans l'état *SUSPENDED*, tous les comportements de cet agent sont suspendus et aucun comportement n'est exécuté. Un agent peut également faire appel à des ressources externes (interfaces avec des applications externes, protocoles de communication spécifiques, par exemple). Les ressources exploitées par un agent sont représentées par le bloc repéré 5 sur la figure 5.13.

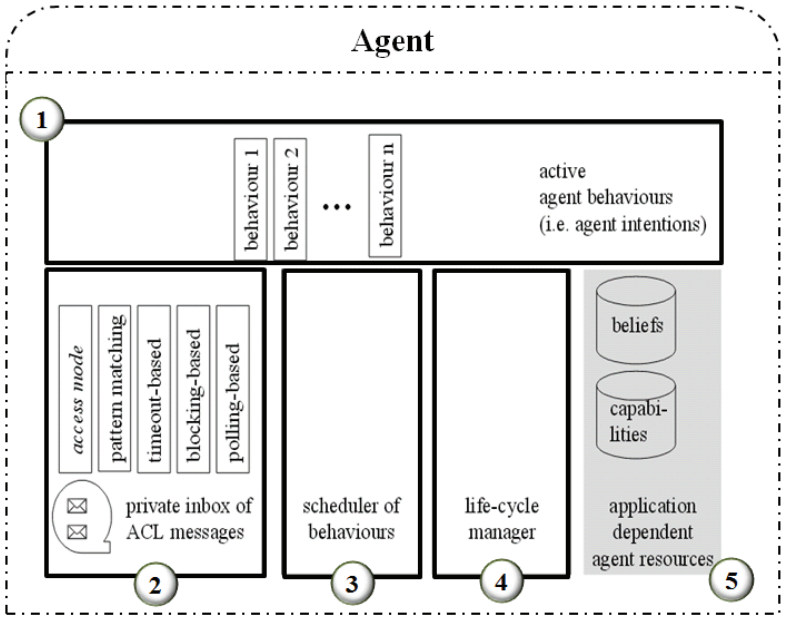

Figure 5.13. Architecture interne d'un agent Jade, adaptée de (Bellifemine et al., 1999).

Chaque agent dispose d'une boite de réception interne (*inbox*) qui contient les messages ACL transmis par les autres agents (bloc repéré 2 sur la figure 5.13). Un agent effectue une scrutation de cette boite pour traiter les messages qu'il reçoit (*polling-based*). La plateforme Jade propose différents mécanismes pour accéder au contenu de la boite de réception d'un agent. Par exemple, un agent peut consulter de manière bloquante (arrêt temporaire de son activité) ou non-bloquante la pile de messages ACL qu'il doit traiter (*access mode*). Ces deux modes d'accès peuvent être complétés par une méthode de reconnaissance de messages afin d'extraire uniquement certains types de messages de la boite de réception (*pattern matching*). Le format des messages échangés entre les agents est détaillé dans la section suivante.

3.2.2 Communication entre agents

Chaque agent intègre une librairie spécifique lui permettant d'envoyer et de recevoir des messages au standard ACL (*jade.lang.acl.ACLMessage*). Un message ACL doit obligatoirement contenir un performatif et un ou plusieurs destinataires du message. Le performatif d'un message représente la nature de l'action souhaitée par l'agent expéditeur de ce message. Un performatif permet ainsi de définir l'objet d'un message. A titre d'exemple, les performatifs usuels définis par les spécifications FIPA sont les suivants :

- *INFORM* : envoi d'une information ou d'une croyance considérée comme vraie par l'agent expéditeur du message.
- *REQUEST* : demande d'une information ou d'une action spécifiée dans le contenu du message par l'agent expéditeur. Ce performatif implique généralement une transmission d'un message ACL de la part du destinataire en réponse à ce type de message.
- *CONFIRM* : envoi d'une confirmation sur une proposition considérée comme vraie par l'agent expéditeur du message et incertaine par l'agent destinataire.

- *REFUSE* : message exprimant le refus d'accomplir une action demandée et dont le contenu précise les raisons de ce refus par l'agent expéditeur.

Un message ACL comporte généralement un performatif (*performative*), un expéditeur (*sender*), un destinataire (*receiver*), un contenu (*content*) et un identifiant de la conversation (*conversation-id*). Un agent peut transmettre un même message à plusieurs agents (diffusion) en les ajoutant à une liste de destinataires. La plateforme Jade propose une méthode qui renseigne automatiquement certains champs d'un message ACL lorsqu'il s'agit d'un message de réponse à transmettre à un agent. Dans ce cas, l'agent émetteur du message de réponse doit renseigner uniquement le performatif et le contenu de ce message. La figure 5.14 présente un exemple de message ACL transmis par un agent ADPSC vers un agent AC. Dans cet exemple, l'agent ADPSC informe l'agent AC d'un résultat de diagnostic non contextualisé relatif à la partie PSC d'un accès voyageurs.

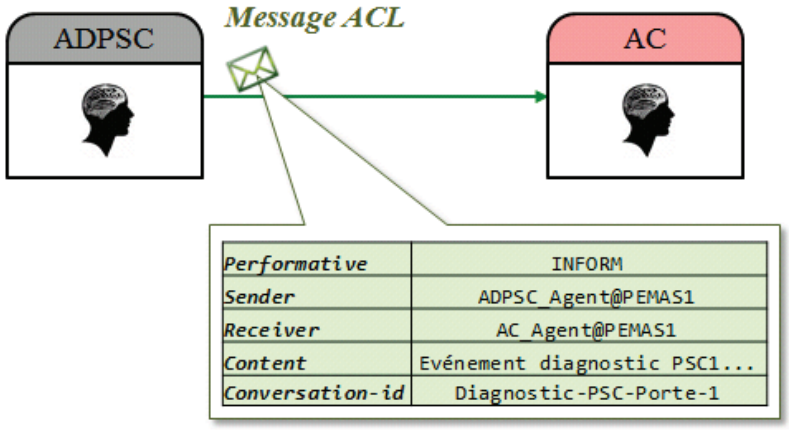

Figure 5.14. Exemple de message ACL transmis par un agent ADPSC.

Pour visualiser les différents échanges entre agents et mettre au point leur comportement, un agent spécifique est disponible dans Jade (Bellifemine et al., 2007). Cet agent de visualisation (*Sniffer Agent*) représente graphiquement les messages échangés entre les agents et permet d'accéder au contenu des messages par le biais d'une interface graphique.

3.2.3 Exemple de comportement de l'agent AC

Le module de contextualisation individuelle implémenté par un agent AC associé à une porte peut être mis en œuvre par un comportement cyclique. Ce type de comportement est exécuté indéfiniment après la phase d'initialisation de l'agent. Le comportement cyclique *InhibitionBehaviour* d'un agent AC consiste à scruter en permanence l'arrivée d'un nouveau message de diagnostic et à analyser les événements de diagnostic qu'il contient. Cette analyse vise à inhiber un ou plusieurs événements de diagnostic. Chaque message de diagnostic reçu par l'agent AC est traité par une procédure de traitement qui implémente le processus d'inhibition. Cette procédure permet de traiter les événements de diagnostic contenus dans un message ACL. Dans un premier temps, une liste d'événements de diagnostic est créée par l'agent AC à partir du message ACL reçu. Le traitement des événements de cette liste est ensuite séquentiel : chaque événement de diagnostic fait l'objet d'une analyse pour déterminer si cet événement peut être justifié ou non par des informations de contextes détenues par

l'agent AC. Les informations de contexte sont obtenues par l'agent AC grâce à une interface OPC interne. Un agent AC exploite une table d'inhibition individuelle au format XML qui détermine les variables de contexte à examiner pour un événement de diagnostic donné. Ainsi, une liste de variables de contexte est générée pour chaque événement de diagnostic analysé. Par la suite, les variables de contexte contenues dans cette liste sont comparées aux informations de contexte mémorisées par l'agent AC. Lorsqu'une variable de contexte permet de justifier l'événement de diagnostic analysé, cet événement est inhibé. Un événement de diagnostic inhibé se traduit par une suppression de cet événement dans la liste des événements de diagnostic. Après analyse de l'ensemble des événements de diagnostic contenus dans la liste, cette liste peut être vide. Dans ce cas, tous les événements de diagnostic contenus dans le message ACL ont été inhibés et la procédure de traitement est terminée. Lorsque des événements de diagnostic sont toujours présents dans la liste d'événements de diagnostic, ces événements n'ont pas été inhibés. Dans ce cas, cette liste d'événements de diagnostic pertinents est transmise à un agent AC de niveau supérieur. La procédure de traitement se termine après la transmission de cette liste d'événements non-inhibés.

4. Résultats obtenus

Dans le cadre du projet SURFER, l'architecture de surveillance holonique proposée a été appliquée à un train réel conçu par Bombardier-Transport et implémentée de manière progressive. Cette partie présente les principaux résultats associés aux différentes étapes de la mise en œuvre de notre proposition.

4.1 Implémentation de la couche acquisition embarquée

Un prototype SURFER a été implanté dans plusieurs trains en service commercial (*c.f.* annexe D). Dans la première version de ce prototype, des modules de surveillance de bas niveau ont été implémentés afin de collecter des signaux relatifs au système train et à ses sous-systèmes. Ces modules indispensables réalisent une acquisition et une datation précise de signaux ainsi qu'un archivage continu des données collectées dans des journaux. L'analyse de ces données SURFER est ensuite réalisée au sol. Selon les informations fournies par Bombardier-Transport, la figure 5.15 compare l'approche de surveillance actuelle (Orbita) et deux approches possibles grâce aux données SURFER pour diagnostiquer une défaillance au sol. Cette comparaison est basée sur le temps d'analyse moyen (temps nécessaire à identifier la cause racine de la défaillance) et le taux de réussite du diagnostic (ratio entre le nombre de défaillances avérées et le nombre de défaillances diagnostiquées avec succès).

Approche utilisée au sol	Temps d'analyse	Taux de réussite
#1 : Approche actuelle (Orbita)	1 heure à plus de 6 heures	80%
#2 : Analyse humaine de signaux bruts (SURFER)	4 à 6 heures	95%
#3 : Analyse assistée par une méthode de diagnostic (SURFER)	30 minutes à 1 heure	95%

Figure 5.15. Comparaison des approches de diagnostic au sol.

Dans l'approche actuelle utilisée par Bombardier-Transport, les données bruts des sous-systèmes du train sont analysées manuellement (états, alarmes) pour diagnostiquer une défaillance signalée par les exploitants.

Dans la seconde approche, les données analysées manuellement sont les signaux enregistrés par le prototype SURFER. Le temps d'analyse de ces données est supérieur à la première

approche. Néanmoins, ces données plus riches permettent une meilleure convergence du diagnostic. Enfin, lorsque les signaux fournis par le prototype SURFER sont analysés à l'aide d'une méthode de diagnostic, le temps d'analyse est réduit considérablement. Cependant, le taux de réussite est identique à la seconde approche (limitations liées à l'absence de certains signaux ou au modèle utilisée par la méthode de diagnostic, par exemple).

4.2 Mise en œuvre de la méthode de diagnostic à base de modèle

La méthode de diagnostic mise en œuvre, nommée diagnostic automatique, a été validée sur une plateforme de simulation des accès voyageurs, également nommé simulateur HIL (*Hardware-in-the-loop*). Conçue dans le cadre du projet SURFER, cette plateforme se compose d'un simulateur de la partie opérative d'un accès voyageurs, associé à des éléments réels de ce sous-système. Grâce à cette plateforme, il est possible d'injecter des fautes sur des composants virtuels de la partie opérative d'une porte (blocage d'un actionneur, par exemple). La méthode de diagnostic mise en œuvre a ainsi été validée *en-ligne* pendant la sollicitation d'un accès voyageur. Dans cette campagne de validation, trois indicateurs ont été définis. Pour chaque faute simulée, les indicateurs suivants ont été renseignés :

- Test de détection : pour une faute simulée, le test de détection est concluant lorsqu'une discordance est détectée entre le comportement prédit par la méthode de diagnostic et le comportement observé de la partie opérative porte.
- Test de discrimination : ce test est positif lorsque le résultat d'un diagnostic contient le composant physique sur lequel une faute a été simulée.
- Taux d'isolation : ce taux caractérise la résolution du diagnostic et correspond au nombre de composants suspects contenus dans le résultat d'un diagnostic. Ce taux est calculé à partir des résultats de diagnostic pour lesquels le test de discrimination est positif.

La figure 5.16 présente les résultats de la campagne de validation réalisée. La méthode de diagnostic a été validée en simulant 61 fautes internes à la partie opérative d'un accès voyageurs. Le taux d'isolation obtenu représente en moyenne, pour chaque faute simulée, 7 composants suspects identifiés par la méthode de diagnostic. Le modèle utilisé par cette méthode intègre une description fine de la partie opérative. Ainsi, un composant suspect représente tout élément physique défini dans ce modèle (capteur, actionneur, liaison électrique, alimentation, par exemple).

Indicateurs	Méthode de diagnostic à base de modèle	Approche actuelle
Test de détection	52 / 61	31/61
Test de discrimination	44 / 61	-
Taux d'isolation moyen (résolution du diagnostic)	14, 35%	-

Figure 5.16. Résultats de la campagne de validation de la méthode de diagnostic.

Les résultats du test de détection ont été comparés avec l'approche actuelle de génération d'alarmes des accès voyageurs. Dans cette approche, les sous-systèmes accès voyageurs réalisent eux-mêmes la détection de fautes sans élaborer le diagnostic. Selon nos expérimentations, la méthode de diagnostic à base de modèle mise en œuvre permet de détecter davantage de fautes. Grâce à cette méthode, le nombre de fautes détectées est augmenté de 34 % par rapport à l'approche actuelle de génération d'alarmes. Par ailleurs, le taux d'isolation moyen obtenu avec cette méthode s'avère être acceptable.

4.3 Contextualisation du diagnostic

Les informations contextuelles associées aux sous-systèmes accès voyageurs et au système train sont élaborées grâce aux données collectées par la couche acquisition embarquée. Une couche de traitement supplémentaire réalise un calcul de variables nécessaires à la contextualisation du diagnostic.

Différentes règles d'expertise visant à inhiber des événements de diagnostic ont été définies. Nous estimons que les règles implémentées au niveau d'un accès voyageurs permettent d'inhiber 12 % des événements de diagnostic générés par la méthode de diagnostic utilisée. Les règles implémentées au niveau du système train permettent d'inhiber, dans certains contextes opérationnels, la quasi-totalité des événements de diagnostic. De nouvelles règles pourront être ajoutées en étendant l'observabilité des sous-systèmes accès voyageurs pendant les phases de roulage des trains.

Par ailleurs, certaines règles peuvent être appliquées sur les alarmes émises actuellement par les sous-systèmes accès voyageurs. La figure 5.17 représente une estimation des gains potentiels de la contextualisation de ces événements à l'aide de données fournies par Bombardier-Transport.

Evénements d'une flotte de trains	Approche actuelle
Alarmes associées aux accès voyageurs	11 410
Alarmes justifiées par des informations de contexte train	2 278

Taux d'inhibition de fausses alarmes ≈ 20 %

Figure 5.17. Estimation du nombre d'alarmes inhibées sur une flotte de trains.

Dans l'approche actuelle, le nombre d'alarmes relatives aux accès voyageurs d'une flotte de 115 trains transmises au sol s'élève à 11 410 événements en trois mois. En considérant des informations de contexte des trains, ces alarmes peuvent être réduites de l'ordre de 20 %. Il est également possible d'appliquer des règles basées sur des informations locales de contexte des sous-systèmes accès voyageurs. Ces règles peuvent être spécifiées aux fournisseurs et être implémentées directement dans les sous-systèmes accès voyageurs.

5. Conclusion

Dans ce chapitre, l'architecture de surveillance holonique proposée a été appliquée au domaine ferroviaire dans le cadre du projet SURFER. Le contexte industriel de ce projet financé par le FUI a été présenté dans la première partie de ce chapitre. Dans la seconde partie, l'architecture holonique proposée a été instanciée pour la surveillance des accès voyageurs d'un train de passagers. Dans notre approche, des holons de surveillance sont associés à chaque niveau de décomposition du système train (niveau porte, niveau véhicule et niveau train). Les comportements dynamiques des holons de surveillance ont été illustrés à travers trois scénarios de diagnostic. Ces scénarios mettent en évidence les bénéfices de l'application d'une surveillance holonique à un système train. La plateforme expérimentale conçue pour implémenter ces holons de surveillance et valider notre approche avant une implantation embarquée du diagnostic a été présentée dans la dernière partie de ce chapitre. Dans cette plateforme nommée PEMAS, les parties informationnelles des holons de

surveillance sont implémentées par des agents logiciels. Ces agents sont distribués sur plusieurs systèmes matériels et communiquent entre eux par le biais d'un réseau de diagnostic dédié. La plateforme multi-agents Jade a été retenue pour implémenter les différents agents de l'architecture de surveillance. Les résultats présentés à la fin de ce chapitre sont liés à une implémentation incrémentale de notre proposition dans des trains réels en exploitation. Par ailleurs, les gains estimés à partir de données fournies par le constructeur s'avèrent très prometteurs.

Conclusion générale

Les travaux réalisés dans le cadre de cette thèse ont porté sur la proposition d'une architecture de surveillance "active" à base d'agents intelligents pour l'aide à la maintenance de systèmes mobiles. Dans notre contribution, un système mobile est considéré comme un objet capable d'émettre des informations pertinentes aux services de maintenance grâce à ses capacités d'auto-surveillance. Nous avons ainsi proposé une architecture générique pour la surveillance embarquée de systèmes mobiles nommée EMH (*Embedded Monitoring Holarchy*). Cette architecture basée sur des entités de surveillance autonomes, récursives et coopératives a été mise en œuvre et appliquée à un système réel de transport dans le domaine ferroviaire.

Les spécificités des systèmes cibles considérés dans nos travaux et les principales notions relatives à la surveillance dans un contexte de maintenance ont été introduites au début de ce manuscrit. Le premier chapitre a notamment permis de définir les différentes fonctions réalisées par un système de surveillance et de retenir un modèle d'architecture de surveillance dans le cadre de notre étude. Suivant ce modèle, différentes problématiques et contraintes à considérer pour la surveillance de systèmes cibles mobiles ont été exposées. Les exigences d'un système de surveillance considérées dans nos travaux ont par la suite été définies.

Le second chapitre s'est focalisé sur la surveillance de systèmes cibles mobiles en dressant tout d'abord un état de l'art des approches de la littérature pour diagnostiquer de tels systèmes. Différentes classes d'architectures de diagnostic ont été définies et une typologie des architectures a été proposée. Les exigences établies dans le premier chapitre ont permis de sélectionner une classe d'architecture dans laquelle le diagnostic est élaboré à bord du système cible de manière décentralisée et coopérative. Dans la littérature, les applications relatives à cette classe d'architecture sont relativement restreintes à un domaine applicatif et manquent de généricité. Le concept de surveillance active nous a permis de caractériser les entités de surveillance qui composent cette classe d'architecture. Dans le cadre de nos travaux, une approche holonique s'est avérée pertinente pour concevoir des entités de surveillance "actives". Cette approche a été retenue afin de proposer une architecture générique pour la surveillance de systèmes cibles mobiles.

Le troisième chapitre fut consacré à notre proposition d'une architecture de surveillance "active" conformément à la classe d'architecture choisie. Une approche holonique nous a permis de proposer une architecture générique, nommée EMH, pour la surveillance embarquée de systèmes cibles mobiles. Cette architecture holonique est formée par un collectif de holons de surveillance qui adhérent à la décomposition systémique du système cible. Un holon est formé par l'association d'une fonction de surveillance et d'une partie tangible du système cible à surveiller. Nous avons distingué trois niveaux de diagnostic internes à une fonction de surveillance d'un holon. Les deux aspects novateurs de notre proposition résident dans la contextualisation individuelle (niveau 2) et collective (niveau 3) du diagnostic. Suivant ces deux niveaux de contextualisation, un holon de surveillance est capable de considérer des informations de contexte locales d'une partie physique sous surveillance comme des informations de contexte fournies par d'autres holons pour élaborer un diagnostic.

Le quatrième chapitre a présenté une mise en œuvre de l'architecture holonique proposée. Une méthode de diagnostic à base de modèle qualitatif a été mise en œuvre et deux modules de contextualisation ont été proposés. Un processus d'inhibition a été défini pour implémenter le module de contextualisation individuelle d'une fonction de surveillance. Une extension de ce processus a été proposée pour implémenter le module de contextualisation collective. Les comportements attendus des holons de surveillance suivant ces processus ont été définis. Le paradigme agent a été choisi pour mettre en œuvre les parties informationnelles des holons de surveillance par des agents logiciels. Ces agents peuvent être répartis sur plusieurs systèmes matériels.

Dans le dernier chapitre, l'architecture de surveillance proposée a été appliquée au domaine ferroviaire. Une instanciation de cette architecture pour la surveillance des accès voyageurs d'un train de passagers a été présentée dans le cadre du projet SURFER. En accord avec les niveaux de décomposition d'un système train considérés, plusieurs types de holons de surveillance ont été définis. Les comportements de ces holons ont été illustrés à travers différents scénarios de diagnostic qui mettent en évidence les bénéfices de notre approche. La plateforme PEMAS développée dans le cadre du projet SURFER a permis de mettre en œuvre ces holons dans un environnement de développement multi-agents. Les comportements des holons ont ainsi été implémentés et validés à l'aide de cette plateforme avant l'implantation finale des holons à bord de trains en situation réelle. Les bénéfices de notre approche ont été confirmés par une estimation du nombre d'alarmes actuellement émises par une flotte de trains qui peuvent être justifiées par des informations de contexte.

Perspectives de Recherche

Les résultats très prometteurs obtenus dans le cadre applicatif du projet SURFER nous rendent confiants dans la poursuite des travaux présentés dans cette thèse. Les perspectives envisagées à court, moyen et long terme sont exposées ci-après.

Les perspectives à court terme portent sur un second volet du projet SURFER qui traite de la maintenance prédictive. Ces travaux visent à étendre l'architecture de surveillance EMH proposée en y intégrant une méthode de diagnostic prédictif. Cette méthode complémentaire à la méthode de diagnostic mise en œuvre dans nos travaux permettra de diagnostiquer des défaillances progressives des accès voyageurs d'un train de passagers. La prise en charge d'un module de pronostic par des holons de surveillance embarqués est également une évolution de notre proposition. Suivant cette approche, un holon de surveillance sera capable de supporter des méthodes hétérogènes de diagnostic et de pronostic et de contextualiser les résultats fournis par ces différentes méthodes.

Les perspectives à moyen terme concernent tout d'abord le développement de nouvelles capacités de raisonnement qui pourront être supportées par les holons de surveillance. En effet, la prise en compte de méthodes liées à la maintenance prédictive permet d'envisager des modules complémentaires pris en charge par certains holons (apprentissage, historisation des interventions de maintenance, par exemple). Par la suite, ces travaux consisteront à étudier d'autres méthodes de diagnostic en considérant la performance du diagnostic obtenu et les coûts liés au déploiement de la méthode (coûts de modélisation, de calibration des modèles, par exemple).

Les perspectives à long terme s'articulent autour de trois axes de Recherche précisés ci-après.

- Surveillance des organes de contrôle : la fonction de surveillance d'un holon proposée dans nos travaux élabore le diagnostic de la partie sous-contrôle d'un équipement (partie opérative d'un accès voyageurs, par exemple). Cette fonction peut néanmoins considérer des informations contextuelles relatives à la partie contrôle de ce système (unité de contrôle d'un accès voyageurs, par exemple). Toutefois, seuls des composants physiques de la partie sous-contrôle sont incriminés par une fonction de surveillance. Ce premier axe a donc pour ambition de surveiller les organes de contrôles des équipements (diagnostic de défaillances logicielles ou diagnostic "fonctionnel" d'une unité de contrôle, par exemple). Ces travaux pourraient enrichir l'architecture de surveillance EMH proposée avec de nouvelles méthodes de diagnostic associées aux parties contrôle des équipements.

- Extension du périmètre de surveillance SURFER : le cas d'étude retenu dans le projet SURFER a été la surveillance des accès voyageurs d'un train de passagers. L'architecture de surveillance proposée peut également être instanciée pour surveiller davantage de sous-systèmes implantés dans un train. Ce deuxième axe concerne le déploiement de l'architecture EMH sur les différents sous-systèmes d'un train et la généralisation de notre approche sur différents types de trains. Par exemple, des holons de surveillance associés aux sous-systèmes de climatisation et de traction d'un train pourraient être proposés.

- Fiabilisation de l'architecture de surveillance : plusieurs hypothèses ont été considérées dans nos travaux quant à l'élaboration d'une information de diagnostic par les holons de surveillance. Nous avons notamment considéré que les fonctions de surveillance des holons tout comme les échanges entre ces entités sont fiables. Aussi, nous avons supposé que les dispositifs matériels supportant les fonctions de surveillance des holons sont infaillibles. Cependant, des défaillances peuvent survenir dans différentes parties de l'architecture EMH mise en œuvre. Par exemple, une faute peut se manifester dans l'un des modules logiciels implémentant une fonction de surveillance d'un holon, ou dans l'un des dispositifs physiques supportant une fonction. Ce troisième axe s'inscrit dans la continuité des travaux réalisés dans le cadre du projet SURFER relatifs à la fiabilité de l'architecture de surveillance (Gandibleux et al., 2012). Une approche holonique pourrait être envisagée pour aborder la supervision de l'architecture de surveillance proposée. Suivant cette approche, la fonction de surveillance d'un holon pourrait être supervisée par un holon de niveau supérieur. De nouvelles capacités associées aux holons de surveillance pourraient ainsi être développées (reconfiguration dynamique des holons de l'architecture, désactivation des holons jugés défaillants, par exemple).

Références

(Abichou, 2013) Abichou, B. Contribution à la formalisation de bilans / états de santé multi-niveaux d'un système pour aider à la prise de décision en maintenance : agrégation d'indicateurs par l'intégrale de Choquet. Thèse de doctorat. Université de Lorraine.

(Alali Alhouaij, 2010) Alali Alhouaij, A. Contribution à l'optimisation de la maintenance dans un contexte distribué. Thèse de doctorat. Université de Grenoble.

(Alanen et al., 2006) Alanen, J., Haataja, K., Laurila, O., Peltola, J., et Aho, I., Diagnostics of mobile works machines. Technical Research Centre of Finland. VTT Research Notes 2343. ESPOO 2006.

(Albert et al., 2001) Albert, M., DIAMOND Reference Model. Karlsruhe University. ESPRIT Project 28735.

(Atlas et al., 2001) Atlas, L., Bloor, G., Brotherton, T., Howard, L., Jaw, L., Kacprzynski, G., et al. An evolvable tri-reasoner ivhm system. In Aerospace Conference, 2001, IEEE Proceedings. Vol. 6, 3023-3037.

(Azarian et al., 2010) Azarian, A., Siadat, A., Martin, P. A new strategy for automotive off-board diagnosis based on a meta-heuristic engine. Engineering Applications of Artificial Intelligence 24, 733-747.

(Balin et Stankunas, 2013) Balin, C. E., & Stankūnas, J. Investigation of Fault Detection and Analysis Methods for Central Maintenance Systems. Aviation technologies, Vol. 1(1).

(Balle et Isermann, 1998) Balle, P., & Isermann, R. Fault detection and isolation for nonlinear processes based on local linear fuzzy models and parameter estimation. In *American Control Conference,* IEEE.

(Basseville et Cordier, 1996) Bassevile, M. et Cordier, M., *Surveillance et diagnostic de systèmes dynamiques : approches complémentaires du traitement de signal et de l'intelligence artificielle*. INRIA. Rapport de recherche n° 2861. Avril 1996.

(Belhadaoui, 2011) Belhadaoui, H. Conception sûre des systèmes mécatroniques intelligents pour des applications critiques. Thèse de doctorat de l'Institut National Polytechnique de Lorraine.

(Bellifemine et al., 1999) Bellifemine, F., Poggi, A., Rimassa, G. JADE-A FIPA-compliant agent framework. In Proceedings of PAAM, Vol. 99 (97-108), 33.

(Bellifemine et al., 2007) Bellifemine, F.L., Caire, G., Greenwood, D. Developing Multi-Agent Systems with JADE. Wiley. Wiley series in Agent Technology.

(Benedettini et al., 2009) Benedettini, O., Baines, T.S., Lightfoot, H.W., Greenough, R.M. State-of-the-art in integrated vehicle health management. Proc. IMechE, Part G: J. Aerosp. Eng. 223 (2), 157–170.

(Bengtsson, 2003) Bengtsson, M. Condition Based Maintenance On Rail Vehicles, Possibilities for a more effective maintenance strategy. Technical Report, Eskilstuna, Sweden.

(Berenji et al., 2005) Berenji, H., Wang, Y., Saxena, A. Dynamic Case-Based Reasoning in Fault Diagnosis and Prognosis. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). Reno, Nevada, USA.

(Bharadwaj et al., 2013) Bharadwaj, R., Mylaraswamy, D., Cornhill, D., Biswas, G., Koutsoukos, X., & Mack, D. Vehicle Integrated Prognostic Reasoner (VIPR) Final Report. National Aeronautics and Space Administration. CR–2013-217972.

(Biteus, 2004) Biteus, J., Jensen, M., & Nyberg, M. Decentralized diagnosis in heavy duty vehicles. In CCSSE, 2004.

(Blanc et al., 2008) Blanc, P., Demongodin, I., Castagna, P. A holonic approach for manufacturing execution system design: An industrial application. Engineering Applications of Artificial Intelligence, 21(3), 315-330.

(Brahim-Djelloul et al., 2012) Brahim-Djelloul, S., Estampe, D., Lamouri, S. Cadre de modélisation pour le pilotage dynamique du transport routier à l'aide de technologie RFID. In 9th International Conference on Modeling, Optimization & SIMulation.

(Bussman, 1998) Bussmann, S. An Agent-Oriented Architecture For Holonic Manufacturing Control, First Open Workshop, Proceedings Of IMS'98 - ESPRIT Workshop In Intelligent Manufacturing Systems(Lausanne, Switzerland).

(Byington et al., 2003) Embedded diagnostic/prognostic reasoning and information continuity for improved avionics maintenance. In *AUTOTESTCON 2003. IEEE Systems Readiness Technology Conference, 320-329.*

(Campos, 2009) Campos, J. Development in the application of ICT in condition monitoring and maintenance. Comput. Ind., 60.

(Cauffriez et al., 2013) Cauffriez, L., Loslever, P., Caouder, N., Turgis, F., & Copin, R. Robustness study and reliability growth based on exploratory design of experiments and statistical analysis: a case study using a train door test bench. The International Journal of Advanced Manufacturing Technology, 1-18.

(Chittaro and Ranon, 1999) Diagnosis of multiple faults with flow-based functional models: the functional diagnosis with efforts and flows approach. Reliability Engineering & System Safety, 64(2), 137-150.

(Cocheteux et al., 2007) Cocheteux, P., Voisin, A., Levrat, E. et Iung, B. Formalisation du pronostic à base d'une approche processus. Colloque Internationnal Francophone Performance et Nouvelles Technologies en Maintenance. PENTOM 2007, Mons, 2007.

(Console et Dressier, 1999) Console, L. and Dressier, O. Model-based diagnosis in the real world: lessons learned and challenges remaining. In: Proc. IJCAI-99, Stockholm, Sweden, 1393-1400.

(Cordier et al., 2000) Cordier, M.O., Dague, P., Dumas, M., Levy, F., Montmain, J., Staroswiecki, M., Trave-Massuyes, L. AI and automatic control approaches of model-based diagnosis: Links and underlying hypotheses. In: Proceedings of the 4th IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes. Budapest, 274-279.

(Cordier et al., 2004) Cordier, M. O., Dague, P., Lévy, F., Montmain, J., Staroswiecki, M., Travé-Massuyès, L. Conflicts versus analytical redundancy relations: a comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 34(5), 2163-2177.

(Cornhill et al., 2013) Cornhill, D., Bharadwaj, R., Mylaraswamy, D. Vehicle Integrated Prognostic Reasoner (VIPR) Metric Report. National Aeronautics and Space Administration. CR-2013-217978.

(Dash et Venkatasubramanian, 2000) Dash, S., Venkatasubramanian, V. Challenges in the industrial applications of fault diagnostic systems. Computers & Chemical Engineering, 24(2), 785-791.

(Davis, 1984) Davis, R. Diagnostic reasoning based on structure and behavior. Artificial intelligence, 24(1), 347-410.

(De Ambrosi et al., 2008) De Ambrosi, C., Ghersi, C., Tacchela, A. Ontology-based diagnosis of railway rolling stock. Intelligent Integration of Railway Systems. Integrail FP6 EU Project. Formal Ontology in Information Systems (FOIS). Fifth International Conference, 2008.

(De Kleer et al., 1992) De Kleer, J., Mackworth, A. K., & Reiter, R. Characterizing diagnoses and systems. Artificial Intelligence, 56(2), 197-222.

(De Kleer et Williams, 1987) De Kleer, J., Williams, B.C. Diagnosing multiple faults. Artificial Intelligence, 32(1):97-130.

(Deen, 2003) Deen, S. M. Agent based manufacturing : advances in holonic approach. Springer Verlag.

(Delépine et Caulier, 2005) Delépine, O., Caulier, P. A modelling concept for complex systems for advanced supervision. In 17th IMACS WORLD CONGRESS Scientific Computation, Applied Mathematics and Simulation.

(Deloux, 2008) Deloux, E., Politiques de maintenance conditionnelle pour un système à dégradation continue soumis à un environnement stressant. Thèse de doctorat. Université de Nantes.

(Demazeau et Briot, 2001) Demazeau, Y., et Briot, J.P., *Principes et architectures des systèmes multi-agents*. Paris. Ed. Hermes, 2001.

(Demazeau, 2001) Demazeau, Y. VOYELLES. Mémoire d'Habilitation à Diriger des Recherches. Institut National Polytechnique de Grenoble (INPG).

(Dievart et al., 2010) Dievart, M., Charbonneau, P., Desforges, X. An embedded distributed tool for transportation systems health assessment. Conference on Embedded Real Time Software and Systems, ERTSS.

(Dievart, 2010) Dievart, M. Architectures de Diagnostic et de Pronostic Distribuées de Systèmes Techniques Complexes de Grande Dimension. Thèse de doctorat. Université de Toulouse.

(Domlan, 2006) Domlan, E.A. Diagnostic des systèmes à changement de régime de fonctionnement. Thèse de doctorat. Institut National Polytechnique de Lorraine (INPL).

(Donnadieu et al., 2003) Donnadieu, G., Durand, Neel D., Nunez E., Saint-Paul, L. L'Approche systémique : de quoi s'agit-il ? Diffusion de la pensée systémique. Synthèse des travaux du groupe AFSCET.

(Durand, 2011) Durand, B. Proposition d'une architecture de contrôle adaptative pour la tolérance aux fautes. Thèse de doctorat. Université de Montpellier II.

(Eclipse, 2013) The Eclipse Foundation open source community website. http://www.eclipse.org/

(El koujok, 2010) El koujok, M., Contribution au pronostic industriel : intégration de la confiance à un modèle prédictif neuro-flou. Thèse de Doctorat. Université de Franche-Comté.

(Euromain, 2002) European Railway Open Maintenance System. http://www.euromain.org/

(Fabre, 2007) Fabre, E. Bayesian Networks of Dynamic Systems. Habilitation à diriger les Recherches. Université de Rennes 1.

(Feldman, 2010) Approximation Algorithms for Model-Based Diagnosis. Thèse de doctorat. Delft University of Technology, ISBN 978-90-9025023-6.

(Ferber, 1995) Ferber J., Les systèmes multi-agents – vers une intelligence collective, InterEditions, Paris. ISBN 2-7296-0572-X.

(Ferrari, 2009) Distributed Fault Detection and Isolation of Large-scale Nonlinear Systems: an Adaptive Approximation Approach. Ph.D. Thesis, University of Trieste.

(Ferreiro et al., 2012) Ferreiro, S., Arnaiz, A., Sierra, B., Irigoien, I. Application of Bayesian networks in prognostics for a new Integrated Vehicle Health Management concept. Expert Systems with Applications, 39(7), 6402-6418.

(Franck, 1996) Frank, P. Analytical and qualitative model-based fault diagnosis, a survey and some new results. European Journal of Control, Vol. 2, 6-28.

(Fröhlich et al., 1999) Fröhlich, P., Móra, I., Nejdl, W., & Schroeder, M. Diagnostic agents for distributed systems. In Formal Models of Agents . Springer, 173-186.

(Gandibleux et al., 2012) Gandibleux J., Cauffriez L., Branger G. Extension des Concepts de la norme CEI-61508 pour la conception d'un système de surveillance/diagnostic embarqué. conférence internationale francophone d'automatique CIFA, Grenoble, juillet.

(Gatti, 2002) Gatti, A. Train as mobile devices: the TrainCom project. http://www.stellastellina.org/assets/applets/3540_WDC2002The_TrainCom_project.pdf.

(Gaud, 2007) Gaud, N.A. Systèmes Multi-Agents Holoniques : De l'analyse à l'implantation. Université de Belfort-Montbéliard. Laboratoire Systèmes et Transport.

(Genc et Lafortune, 2003) Genc, S., Lafortune, S. Distributed diagnosis of discrete-event systems using Petri nets. In Applications and Theory of Petri Nets. Springer, 316-336.

(Gertler, 1998) Gertler, J.J. Survey of model-based failure detection and isolation in complex plants. Control Systems Magazine, IEEE, 8(6), 3-11.

(Giret et Botti, 2004) Giret, A., Botti, V. Holons and agents. Journal of Intelligent Manufacturing, 15(5), 645-659.

(Gorinevsky et al., 2010) Gorinevsky, D., Smotrich, A., Mah, R., Srivastava, A., Keller, K., Felke, T. Open Architecture for Integrated Vehicle Health Management. The American Institute of Aeronautics and Astronautics (AIAA) Infotech @ Aerospace Conference, Atlanta, GA.

(Grondin, 2008) Grondin, G. MaDcAr-Agent: un modèle d'agents auto-adaptables à base de composants. Thèse de doctorat. Ecole Nationale Supérieure des Mines de Saint-Etienne.

(Gucik-Derigny, 2011) Gucik-Derigny, D. Contribution au pronostic des systèmes à base de modèles : théorie et application. Thèse de doctorat. Université Paul Cézanne Aix-Marseille III

(Guillén et al., 2013) Guillén, A. J., Gómez, J. F., Crespo, A., Guerrerro, A., Sola, A., & Barbera, L. Advances in PHM Application Frameworks: Processing Methods, Prognosis Models, Decision Making. Chemical Engineering Transaction, Vol.33, 391-396.

(Gulledge et al., 2010) Gulledge, T., Hiroshige, S., Iyer, R. Condition-based maintenance and the product improvement process. Computers in Industry 61, 813-832.

(Hamscher, 1992) Hamscher, W., Console, L., De Kleer, J. (editors). Readings in Model-Based Diagnosis, Morgan Kaufmann, San Meteo, CA, USA.

(Hayes-Roth, 1985) Hayes-Roth, F. Rule-based systems. Communications of the ACM, 28(9), 921-932.

(Hunter et al., 2013) Hunter, GW., R. Ross, Berger, D.E., Lekki, J.D, Mah, R.W., Perey, D.F., Schuet, S.R., Simon, Smith, S.W. Concept of Operations for an Integrated Vehicle Health Assurance System. Technical report. National Aeronautics and Space Administration.

(IAEA, 2007) Implementation Strategies and Tools for Condition Based Maintenance at Nuclear Power Plants. ISBN 92-0-103907-7.

(IMS, 2013) Intelligent Manufacturing Systems. Global Research and Business Innovation Program. http://www.ims.org/

(Isermann et Ballé, 1997) Isermann, R. et Ballé, P. Trends in the Application of Model Based Fault Detection and Diagnosis of Technical Processes. Control Engineering Practice, Vol. 5 (No. 5) (1997)

(Isermann, 2005) Isermann, R. Model-based fault-detection and diagnosis–status and applications. Annual Reviews in control, 29(1), 71-85.

(Isermann, 2008) Isermann, R. Mechatronic systems—innovative products with embedded control. Control Engineering Practice, 16(1), 14-29.

(ISO 13374-1, 2003) Surveillance et diagnostic d'état des machines - Traitement, échange et présentation des données. Partie 1: Lignes directrices générales.

(ISO 13374-2, 2007) Surveillance et diagnostic d'état des machines - Traitement, échange et présentation des données. Partie 2: Traitement des données.

(ISO 13374-3, 2012) Surveillance et diagnostic d'état des machines - Traitement, échange et présentation des données. Partie 3 : Echange.

(IVHM-Center, 2008) Integrated Vehicle Health Management Centre. Université de Cranfield. www.cranfield.ac.uk/ivhm/

(Jardine et al., 2006) Jardine, A.K.S., Lin, D., Banjevic, D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mechanical Systems and Signal Processing, 20(7), 1483-1510.

(Jarvis et Jarvis, 2003) Jarvis, D.H., Jarvis, J.H. Holonic diagnosis for an automotive assembly line. In: Deen, S.M. (Ed.), Agent-Based Manufacturing: Advances in the Holonic Approach. Springer, 193-206.

(JeasyOPC, 2013) Java Easy OPC Client Project. http://sourceforge.net/projects/jeasyopc/

(Jianjun et al., 2007) Jianjun, C., Peilin, Z., Guoquan, R., Jianping, F. Decentralized and overall condition monitoring system for large-scale mobile and complex equipment. J. Syst. Eng. Electron. 18 (4), 758–763.

(Kamsu-Foguem et Noyes, 2013) Kamsu Foguem, B., Noyes, D. Graph-based reasoning in collaborative knowledge management for industrial maintenance. Computers In Industry 64(8), 998-1013.

(Katipamula and Brambley, 2005) Katipamula, S., & Brambley, M. R. Review Article: Methods for Fault Detection, Diagnostics, and Prognostics for Building Systems—A Review, Part I. HVAC&R Research, 11(1), 3-25.

(Kempowsky, 2004) Kempowsky, T. Surveillance de procédés à base de méthodes de classification : Conception d'un outil d'aide pour la détection et le diagnostic des défaillances. Thèse de doctorat. Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) - CNRS.

(Khol et Bauer, 2010) Khol, J., Bauer, A. Role-Based Diagnosis for Distributed Vehicle Functions. In 21st International Workshop on the Principles of Diagnosis (DX), Portland USA.

(Kiencke et al., 2006) Kiencke, U., Nielsen, L., Sutton, R., Schilling, K, Papageorgiou, M., Asama, H. The impact of automatic control on recent developments in transportation and vehicle systems. Annual Reviews in Control, 30, 81-89.

(Knotek, 2006) Knotek, M. Fault diagnosis based on temporal analysis. Thèse de doctorat. Université Joseph-Fourier Grenoble.

(Koestler, 1967) Koestler, A. The Ghost in the Machine. Hutchinson publishing Group, London.

(Koutsoukos et al., 2010) Koutsoukos, X., Biswas, G., Mylaraswamy, D.A., Hadden, G.D., Mack, D., Hamilton, D. Benchmarking the Vehicle Integrated Prognostic Reasoner. Annual Conference of the Prognostics and Health Management Society.

(Kunche et al., 2012) Kunche, S., Chen, C., Pecht, M. A review of PHM system's architectural frameworks. The 54th Meeting of the Society for Machinery Failure Prevention Technology, Dayton, OH.

(Kurien et al., 2002) Kurien, J., Koutsoukos, X., Zaho, F. Distributed Diagnosis of Networked, Embedded Systems. Proceedings of the 13st International Workshop on Principles of Diagnosis (DX-2002). Semmering,Austria, 179-188.

(Lanigan et al., 2011) Lanigan, P. E., Kavulya, S., Fuhrman, T. E., Narasimhan, P., Salman, M. A. Diagnosis in automotive systems: A survey. Technical Report CMU-PDL-11-110, Carnegie Mellon University PDL.

(Larsson, 1999) Larsson, M. Behavioral and structural model based approaches to discrete diagnosis. Thèse de doctorat. Université de Linköping, Suède.

(Le Moigne, 1994) Le Moigne, J. L. La théorie du système général: théorie de la modélisation. collection Les Classiques du Réseau Intelligence de la Complexité (IC-MCX-APC).

(Le Mortellec et al., 2012) Le Mortellec, A., Clarhaut, J., Sallez, Y., Berger, T., Trentesaux, D. An Embedded Cooperative Holarchy for Diagnosing Complex Moving Systems. In: Borangiu T., Dolgui A. (Ed.), 14th IFAC symposium on Information & Control problems in Manufacturing (INCOM), IFAC, Bucarest, Romania, may 2012.

(Le Mortellec et al., 2013) Le Mortellec, A., Clarhaut, J., Sallez, Y., Berger, T., Trentesaux, D. Embedded holonic fault diagnosis of complex transportation systems. Engineering Applications of Artificial Intelligence 26 (2013), 227-240. DOI : 10.1016/j.engappai.2012.09.008.

(Lebold et al., 2003) M. Lebold, K. Reichard, D. Boylan : Utilizing dcom in an open system architecture framework for machinery monitoring and diagnostics. In Aerospace Conference Proceedings.

(Lebold et Thurston, 2001) M. Lebold et M. Thurston : Open standards for condition-based maintenance and prognostic systems. In 5th Annuel Maintenance and Reliability Conference (MARCON), Gatlinburg, USA.

(Lefebvre et al., 2000) Lefebvre, A., Simeu-Abazi, Z., Derain, J.P., Glade, M. Diagnostic of the avionic equipment based on dynamic fault tree. IFAC-CEA, Monterrey, Mexico.

(Leitão et Restivo, 2006) Leitao, P., Restivo, F. ADACOR: a holonic architecture for agile and adaptive manufacturing control. Computers in Industry 57, 121-130.

(Leitão, 2004) Leitao, P. An Agile and Adaptive Holonic Architecture for Manufacturing Control. Thèse de doctortat. Université de Porto, Portugal.

(Louati et al., 2012) Louati, T., Ounnar, F., Pujo, P., Pistoresi, C. Intelligent control of people flow in restricted area. Communications, Computing and Control Applications (CCCA).

(Lunze et al., 2001) Lunze J, Schröder J, Supavatanakul P. Diagnosis of discrete event systems: the method and an example. In: Proceedings of the workshop on principles of diagnosis, DX'01, via Lattea, Italy, 111-118.

(Ly et al., 2013) Ly, F., Simeu-Abazi, Z., Léger, J-B. Terminologie Maintenance : bilan. Groupe de Recherche S.P.S.F. Groupe de travail Maintenance. http://www-ljk.imag.fr/membres/Olivier.Gaudoin/FIMA/TerminologieMaintenance.pdf

(Mangina et al., 2001) Mangina, E. E., McArthur S. D .J., McDonald, J.R. COMMAS (Condition Monitoring Multi Agent System). Journal of Autonomous Agents and Multi-Agent Systems, Vol. 4(3), 279-281.

(Maquin et al., 1997) Maquin, D., Cocquempot, V., Cassar, J., Staroswiecki M., and J. Ragot. Generation of analytical redundancy relations for FDI purposes. Proceedings of IFAC Symposium on Diagnosis for Electrical Machines, Power Electronics and Drives (SDEMPED'97).

(Maŕık et al., 2002) Mařík, V., Fletcher, M., & Pěchouček, M. Holons & agents: Recent developments and mutual impacts. In Multi-Agent Systems and Applications II. Springer, 233-267.

(Mathur et al., 2001) Mathur, A., Cavanaugh, K.F., Pattipati, K.R., Willett, P.K., Galie, T.R. Reasoning and Modeling Systems in Diagnosis and Prognosis. Proceedings of the SPIE – The International Society for Optical Engineering, Vol. 4389, 189-194.

(McFarlane et al., 1995) McFarlane, D., Marett, B., Elsley, G., Jarvis, D. Application of holonic methodologies to problem diagnosis in a steel rod mill. IEEE Int. Conf. Syst., Man and Cybern., Int. Syst. 2lst Century, Vancouver 1, 940–945.

(Medina-Oliva et al., 2012) Medina-Oliva, G., Voisin, A., Monnin, M., Peysson, F., Leger, JB. Prognostics Assessment Using Fleet-wide Ontology. Annual Conference of the Prognostics and Health Management Society 2012, Minneapolis, Minnesota : United States

(Monnin et al., 2011) Monnin, M., Voisin, A., Léger, J-B., Iung, B. Fleet-wide health management architecture. Annual Conference of the Prognostics and Health Management Society, Montreal, Canada.

(Müller et al., 1995) Müller, J. P., Pischel, M., Thiel, M. Modeling reactive behaviour in vertically layered agent architectures. In Intelligent Agents. Springer, 261-276.

(Muller et al., 2008) Muller, A., Marquez, A.C., Iung, B.: On the concept of e-maintenance: Review and current research. Rel. Eng. & Sys. Safety (2008)1165-1187.

(Müller, 2000) Müller, J.P., Modélisation organisationnelle en systèmes multi-agents. Septième École d'été de l'ARCo, Bonas, 10-21 juillet 2000.

(Naedele et al., 2004) Naedele, M., Sager, P., Frei, C., Using Multi-Agents Systems for Intelligent Plant Maintenance Functionality. 5th World Congress of Intelligent Control and Automation. WCICA 2005. Hangehou, China, 2004.

(Nasri et al., 2012) Nasri, O., Shraim, H., Dague, P., Heron, O., Cartron, M. Modeling and deployment of model-based decentralized embedded diagnosis inside vehicles: application to smart distance keeping function. Int. J. of Veh. Technol. vol. 2012.

(NF-EN-13306, 2010) DIN EN 13306. Maintenance terminology. European Committee for Standardization.

(OPC Foundation, 2013) The OPC Foundation - The Interoperability Standard for Industrial Automation & Other Related Domains. http://www.opcfoundation.org/

(Orbita, 2006) Orbita-BT. Data monitoring Cuts rail running costs. Professional Engineering journal, 19(20): 50-53.

(OSA-CBM, 2006) Machinery Information Management Open Standards Alliance. Open Systems Architecture for Condition-based Maintenance. Primer. Août 2006. http://www.mimosa.org/

(OSA-EAI, 2010) MIMOSA OSA-EAI Version 3.2.1. Open System Architecture for Enterprise Application Integration. Avril 2010. http://www.mimosa.org/

(Pandalai and Holloway, 2000) Pandalai, D. and Holloway, L. Template languages for fault monitoring of timed discrete event processes. IEEE transactions on automatic control, 45(5):868-882.

(Park et al., 2006) Park, H.G., Barrett, A., Baumann E., Grage, M., Narasimhan, S. Modular Architecture for Hybrid Diagnostic Reasoners. Second IEEE International Conference on Space Mission Challenges for Information Technology, 277-284.

(Patton et Chen, 1997) Patton, R. J., Chen, J. Observer-based fault detection and isolation: robustness and applications. Control Engineering Practice, 5(5), 671-682.

(Pecht et al., 2001) Pecht, M., Dube, M., Natishan, M., Williams, R., Banner, J., Knowles, I. Evaluation of built-in test. IEEE Transactions on Aerospace and Electronic Systems. Vol 37, no.1, 266-271.

(Pencolé et Cordier, 2005) Pencolé, Y., & Cordier, M. O. A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks. Artificial Intelligence, 164(1), 121-170.

(Pipatsrisawat et al., 2009) Pipatsrisawat, K., Darwiche, A., Mengshoel, O.J., Schumann, J. Software Health Management: A Short Review of Challenges and Existing Techniques. Extended Abstract. In Proc. SMC-IT.

(Poole, 1994) Poole, D. Representing diagnosis knowledge. Annals of Mathematics and Artificial Intelligence, 11(1-4), 33-50.

(Provan, 2002) Provan, G. A model-based diagnosis framework for distributed system. In Proceedings of 13th International Workshop on Principles of Diagnosis, 16-24.

(Pujo et al., 2009) Pujo, P., Broissin, N., Ounnar, F. PROSIS : An isoarchic structure for HMS control. Engineering Applications of Artificial Intelligence 22, 1034-1035.

(Qiu et Kumar, 2004) Qiu, W., Kumar, R. Decentralized Failure Diagnosis of Discrete Event Systems, in Proc. of 7th Int. Workshop on Discrete Event Systems. Reims, France, 22-24.

(Rabatel et al., 2009) Rabatel, J., Bringay, S., Poncelet, P. Aide à la décision pour la maintenance ferroviaire préventive. Rapport de recherche. Université Montpellier 2, LIRMM.

(Raccoceanu, 2006) Racoceanu, D., Contribution à la surveillance des systèmes de production en utilisant les techniques de l'intelligence artificielle. Mémoire d'Habilitation à Diriger des Recherches. Université de Franche-Comté de Besançon, 2006.

(Rao et al., 1998) Rao, M., Yang, H., & Yang, H. Integrated distributed intelligent system architecture for incidents monitoring and diagnosis. Computers in Industry, 37(2), 143-151.

(Rao et Georgeff, 1995) Rao, A. S., Georgeff, M. P. BDI Agents: From Theory to Practice. In ICMAS Vol. 95, 312-319.

(Reed et al., 2000) Reed, E., Schumann, J., Mengshoel, O. J. Verification and Validation of System Health Management Models using Parametric Testing. In Proc. Infotech@Aerospace, 2011.

(Reiter, 1987) Reiter, R. A theory of diagnosis from first principles. Artificial Intelligence, 32(1):57-95.

(Ribeiro et Barata, 2011) Re-thinking diagnosis for future automation systems: an analysis of current diagnostic practices and their applicability in emerging IT based production paradigms. Comput. Ind. 62 (7), 639-659.

(Ribeiro, 2012) Ribeiro, L. Diagnosis in Evolvable Production System. Thèse de doctorat. Université Nova de Lisboa.

(Ribot, 2009) Ribot, P., Vers l'intégration diagnostic / pronostic pour la maintenance des systèmes complexes. Thèse de doctorat. Université de Toulouse 3 Paul Sabatier.

(Roos et al., 2003) Roos, N., ten Teije, A., Witteveen, C. Multi-agent diagnosis with semantically distributed knowledge. In Proceedings of the 15th Belgium-Dutch Conference on Artificial Intelligence (BNAIC), 259-266.

(Roth et al., 2011) Roth, M., Lesage, J-J., Litz, L.: The concept of residuals for fault localization in discrete event systems. Control Engineering Practice 19, 978-988.

(Roychoudhury et al., 2009) Roychoudhury I., Biswas, G., Koutsoukos, X. Designing Distributed Diagnosers for Complex Continuous Systems, IEEE Transactions on Automation Science and Engineering, 6(2): 277-290.

(SAE-HM1, 2013) SAE INTERNATIONAL HM-1 Integrated Vehicle Health Management Committee. http://www.sae.org/works/committeeHome.do?comtID=TEAHM1IVHM

(Salama et al., 2012) Salama, A. M., Mohd, S. A., Mazin, A. M., Omar, I. O. Implementing an Expert Diagnostic Assistance System for Car Failure and Malfunction. International Journal of Computer Science Issues (IJCSI), 9(2-2):1694-0814.

(Sallez et al., 2010) Sallez, Y., Berger, T., Deneux, D., Trentesaux, D. The life cycle of active and intelligent products: the augmentation concept. Int. J. Comput. Integrated Manuf. 23 (10), 905–924.

(Sallez et al., 2011) Sallez, Y., Berger T., Sénéchal, O., Branger, G., Trentesaux, D. Augmentation concept for the embedded diagnosis of complex systems. International conference on Industrial Engineering and Systems Management (IESM11), I4E2, Metz, 1444-1453.

(Sallez, 2012) Sallez, Y. Produit « actif » tout au long de son cycle de vie. Habilitation à diriger des Recherches. Université de Valenciennes et du Hainaut Cambrésis.

(Sammouri et al., 2012) Sammouri, W., Come, E., Oukhellou, L., Aknin, P., Fonlladosa, C. E., Prendergast, K. Temporal association rule mining for the preventive diagnosis of onboard subsystems within floating train data framework. In Intelligent Transportation Systems (ITSC), 15th International IEEE Conference. Présentation du 12 juin 2012.

(Sampath et al., 1996) Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., Teneketzis, D. Failure diagnosis using discrete-event models. IEEE Transactions on Control Systems Technology, 4(2), 105-124.

(Schoeller et al., 2007) Schoeller, M.H, Roemer, M.J, Leonard, M.S, Derriso, M. Embedded Reasoning Supporting Aerospace IVHM. AIAA Infotech Conference and Exhibit. Rohnert Park, CA.

(Schumann et al., 2013) Schumann, J., Mbaya, T., Mengshoel, O., Pipatsrisawat, K., Srivastava, A., Choi, A., & Darwiche, A. Software health management with Bayesian networks. Innovations in Systems and Software Engineering, 1-22.

(Schwarzenbach et al., 2010) Schwarzenbach, J., Wilkinson, L., West, M., Pilling, M. Mapping the remote condition monitoring architecture. Research Programme. Rail Safety and Standards Boards (RSSB) LTD. RSSB Core Report.

(Sheppard et al., 2009) Sheppard J.W., Kaufman M.A., Wilmering T.J. IEEE Standards for Prognostics and Health Management. Aerospace and Electronic Systems Magazine, IEEE. Vol.24, no.9, 34-41.

(Silva et al., 2012) Silva, R. M., Arakaki, J., Junqueira, F., Santos Filho, D. J., Miyagi, P. E. Modeling of active holonic control systems for intelligent buildings. Automation in Construction, 25, 20-33.

(Simeu, 2005) Simeu, E. Test et surveillance intégrés des systèmes embarqués. Mémoire d'habilitation à diriger des Recherches. Université Joseph Fourier de Grenoble.

(Simeu-Abazi et al., 2010) Simeu-Abazi, Z., Di Mascolo, M., Notek., M. Fault diagnosis for discrete event systems: Modelling and verification. Reliability and System Safety. Vol. 95(4), 369-378.

(Sreenuch et al., 2013) Sreenuch, T., Tsourdos, A., Jennions, I.K. Distributed embedded condition monitoring systems based on OSA-CBM standard. Computer Standard & Interfaces 35 (2013), 238-246.

(Stecca et al., 2013) Stecca, G., Baffo, I., Galiano, G., & Clemente, F. Design of a holonic remote monitoring and diagnosis system for fleet management. Measurement Vol 46(6), 1947-1956.

(Suaréz et al., 2013) Suárez, S., Leitao, P., Adam, E. Holonic Recursiveness with Multi-Agent System Technologies. In Trends in Practical Applications of Agents and Multiagent Systems. Springer, 103-111.

(Subias, 2006) Subias, A. Contribution au diagnostic des systèmes complexes. Mémoire d'Habilitation à Diriger des Recherches. Université Paul Sabatier-Toulouse III.

(Tanaya et al.,1997) Tanaya, P. I., Detand, J., Kruth, J. P. Holonic Machine Controller: A study and implementation of holonic behaviour to current NC controller. *Computers in Industry*, *33*(2), 323-333.

(Tobon-Mejia et al., 2012) Tobon-Mejia, D.A., Medjaher, K., Zerhouni, N. CNC machine tool's wear diagnostic and prognostic by using dynamic Bayesian networks. Mechanical Systems annd Signal Processing, Vol. 28, 167-182.

(TrainTracer, 2006) Alstom TrainTracer. http://www.cleanmobility.alstom.com/traintracer-v21.html

(Trentesaux, 2009) Trentesaux, D. Distributed control of production systems. Engineering Applications of Artificial Intelligence 22 (7), 971-978.

(Tweedale et al., 2007) Tweedale, J., Ichalkaranje, N., Sioutis, C., Jarvis, B., Consoli, A., Phillips-Wren, G. Innovations in multi-agent systems. Journal of Network and Computer Applications, 30(3), 1089-1115.

(Umiliacchi et al., 2011) Umiliacchi, P., Lane, D., Romano, F., 2011. Predictive maintenance of railway subsystems using an Ontology based modelling approach. The 9th World Conference on Railway Research, WCRR 2011, Lille, France.

(Unland, 2003) Unland, R. A holonic multi-agent system for robust, flexible, and reliable medical diagnosis. In On The Move to Meaningful Internet Systems 2003: OTM 2003 Workshops , 1017-1030. Springer Berlin Heidelberg.

(Vachtsevanos et Lewis, 2006) Vachtsevanos, G. J., Lewis, F.L. Intelligent fault diagnosis and prognosis for engineering systems. 1st ed., John Wiley & Sons, Hoboken, New Jersey.

(Van Brussel et al., 1998) Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P. Reference architecture for holonic manufacturing systems: PROSA. Computers in Industry 37, 255-274.

(Venkatasubramanian et al., 2003a) Venkatasubramanian, V., Rengaswamy, R., Yin, K., & Kavuri, S. N. A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. Computers & chemical engineering, 27(3), 293-311.

(Venkatasubramanian et al., 2003b) Venkatasubramanian, V., Rengaswamy, R., & Kavuri, S. N. A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies. Computers & Chemical Engineering, 27(3), 313-326.

(Venkatasubramanian et al., 2003c) Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., & Yin, K. A review of process fault detection and diagnosis: Part III: Process history based methods. Computers & Chemical Engineering, 27(3), 327-346.

(Vichare et al., 2007) Vichare, N., Rodgers, P., Eveloy, V., Pecht, M. Environment and Usage Monitoring of Electronic Products for Health Assessment and Product Design. Quality Technology & Quantitative Management Vol. 4, No. 2, 235-250.

(Widodo and Yang, 2007) Widodo, A., Yang, B.S. Support vector machine in machine condition monitoring and fault diagnosis. Mech. Syst. Signal Proc. 21 (6), 2560–2574.

(Wiig, 2006) Wiig, J. Optimization of fault diagnosis in helicopter health and usage monitoring systems. Thèse de doctorat. Ecole Nationale Supérieure d'Arts et Métiers (ENSAM).

(Willaeys et Asse, 2011) Willaeys, D., Asse, A. Device and method for a system analysis and diagnosis. United States Patent Application Publication. Pub. no. US008037005B2, Oct. 2011. Prior Pub. no. US 2008/0086288 A1, Apr. 2008.

(Wooldrige et Jennings, 1995) Wooldridge, M. et Jennings, N. R. Agent theories, architectures, and languages: a survey. In Intelligent agents. Springer, 1-39.

(Yang et al., 2002) Yang, H., Mathew, J., Ma, L. Intelligent diagnosis of rotating machinery faults - a review. In Proceedings of the 3rd Asia-Pacific Conference on System Integrity and Maintenance. Queensland, Australia, 385-392.

(Zaki et al., 2007) Zaki, O., Brown, K., Fletcher, J.E., Lane, D., 2007. Detecting faults in heterogeneous and dynamic systems using DSP and an agent-based architecture. Engineering Applications of Artificial Intelligence 20 (8), 1112-1124.

(Zhang et Zhang, 2013) Zhang, Q., Zhang, X. Distributed Sensor Fault Diagnosis in a Class of Interconnected Nonlinear Uncertain Systems. In Fault Detection, Supervision and Safety of Technical Processes, Vol. 8(1), 1101-1106.

(Zille, 2009) Zille, V. Modélisation et évaluation des stratégies de maintenance complexes sur des systèmes multi-composants. Thèse de doctorat, Université de Technologie de Troyes, Institut Charles Delaunay.

Liste des figures

Figure 1. Organisation des chapitres de ce manuscrit.
Figure 1.1. Exemples de système cible et de système de maintenance considérés.
Figure 1.2. Relation causale entre faute, erreur et défaillance.
Figure 1.3. Différents types de maintenance, inspirée de (Alali Alhouaij, 2010).
Figure 1.4. Système de surveillance pour l'aide à la maintenance.
Figure 1.5. Fonctions de surveillance dynamique et classique, adaptée de (Raccoceanu, 2006).
Figure 1.6. Modèle d'architecture ISO 13374, adaptée de (ISO 13374-1, 2003).
Figure 1.7. Exemple d'architecture opérationnelle, adaptée de (ISO 13374-2, 2007).
Figure 1.8. Exemple de module OSA-CBM, adaptée de (OSA-CBM, 2006).
Figure 1.9. Interfaces entre modules, adaptée de (OSA-CBM, 2006).
Figure 1.10. Surveillance d'un système cible suivant les couches fonctionnelles ISO 13374.
Figure 1.11. Exemple de surveillance d'un système cible mobile dans son environnement.
Figure 2.1. Partitionnement du diagnostic : diagnostic distant (a) et diagnostic embarqué (b).
Figure 2.2. Architecture du système TrainTracer, adaptée de (Sammouri et al., 2012).
Figure 2.3. Exemple de diagnostic embarqué, adaptée de (Hunter et al., 2013).
Figure 2.4. Diagnostic centralisé (a), décentralisé (b) et distribué (c), adaptée de (Ferrari, 2009).
Figure 2.5. Architecture VIPR, adaptée de (Cornhill et al., 2013).
Figure 2.6. Positionnement des architectures de diagnostic suivant la typologie proposée.
Figure 2.7. Exemple de surveillance active basée sur le concept d'augmentation.
Figure 2.8. Exemple de SMA appliqué à la surveillance de systèmes, adaptée de (Ferber, 1995).
Figure 2.9. Exemples d'architectures internes d'un agent, adaptée de (Müller et al., 1995).
Figure 2.10. Exemple de système de holons ou holarchie, adaptée de (Gaud, 2007).
Figure 2.11. Architecture interne d'un holon, adaptée de (Bussman, 1998).
Figure 2.12. Périmètres de modélisation considérés dans le cadre de notre étude.
Figure 3.1. Modèle organique du système cible considéré.
Figure 3.2. Modèle organique arborescent du système cible considéré.
Figure 3.3. Structure hiérarchisée de contrôle du système cible considéré.
Figure 3.4. Diagramme de classes UML d'un holon de surveillance.
Figure 3.5. Système sous surveillance et fonction de surveillance d'un holon.
Figure 3.6. Récursivité d'un holon de surveillance.
Figure 3.7. Exemples d'interactions entre un système et son environnement.
Figure 3.8. Coopération entre holons de surveillance.
Figure 3.9. Système cible sous surveillance (a) et architecture de surveillance EMH (b).
Figure 3.10. Données exploitées par une fonction de surveillance.
Figure 3.11. Modules internes d'une fonction de surveillance.
Figure 3.12. Niveaux de diagnostic d'une fonction de surveillance.
Figure 4.1. Modules de diagnostic internes à une fonction de surveillance.
Figure 4.2. Schéma de principe de la méthode de diagnostic mise en œuvre.
Figure 4.3. Modules implémentées par une fonction de surveillance composée et élémentaire.
Figure 4.4. Schéma de principe du processus d'inhibition.
Figure 4.5. Processus d'inhibition sous forme de réseau de Petri.
Figure 4.6. Format de la base de règles exploitée par le processus d'inhibition.
Figure 4.7. Schéma de principe du processus d'inhibition étendue.
Figure 4.8. Processus d'inhibition étendue sous forme de réseau de Petri.
Figure 4.9. Format de la base de règles exploitée par le processus d'inhibition étendue.
Figure 4.10. Architecture EMH considérée pour la description comportementale des holons.
Figure 4.11. Exemple de diagramme de séquences entre les holons $H_{1/1/1}$ et $H_{1/1/2}$.
Figure 4.12. Diffusion d'un diagnostic contextualisé.
Figure 4.13. Inhibition individuelle d'un diagnostic.
Figure 4.14. Inhibition collective d'un diagnostic.
Figure 4.15. Propagation d'un diagnostic contextualisé.
Figure 4.16. Inhibition d'un diagnostic propagé.
Figure 4.17. Implémentation de la partie informationnelle d'un holon élémentaire.
Figure 4.18. Exemple de holon composé implémenté par un système multi-agents.
Figure 4.19. Exemple de projection de holons de surveillance sur des systèmes matériels.

Figure 5.1. Approche de surveillance active embarquée SURFER.
Figure 5.2. Utilisateurs de l'innovation SURFER et principaux services rendus.
Figure 5.3. Architecture de surveillance holonique proposée appliquée à un système train.
Figure 5.4. Exemple de table d'inhibition individuelle d'un holon de surveillance porte.
Figure 5.5. Exemple de table d'inhibition étendue d'un holon de surveillance porte.
Figure 5.6. Exemple de table d'inhibition étendue d'un holon de surveillance véhicule.
Figure 5.7. Exemple de table d'inhibition étendue d'un holon de surveillance train.
Figure 5.8. Diagramme de séquences du scénario #1.
Figure 5.9. Diagramme de séquences du scénario #2.
Figure 5.10. Diagramme de séquences du scénario #3.
Figure 5.11. Architecture matérielle de la plateforme PEMAS.
Figure 5.12. Principaux éléments logiciels de la plateforme PEMAS.
Figure 5.13. Architecture interne d'un agent Jade, adaptée de (Bellifemine et al., 1999).
Figure 5.14. Exemple de message ACL transmis par un agent ADPSC.
Figure 5.15. Comparaison des approches de diagnostic au sol.
Figure 5.16. Résultats de la campagne de validation de la méthode de diagnostic.
Figure 5.17. Estimation du nombre d'alarmes inhibées sur une flotte de trains.

Annexes

Figure B.1. Classification des méthodes de diagnostic.
Figure B.2. Principe des méthodes de diagnostic à base de modèles.
Figure B.3. Méthode de diagnostic à base de modèles quantitatifs (FDI).
Figure B.4. Exemple de matrice de signatures, adaptée de (Domlan, 2006).
Figure B.5. Approche de diagnostic basée sur la cohérence (DX), adaptée de (Feldman, 2010).
Figure B.6. Grille d'évaluation des méthodes de diagnostic proposée.
Figure C.1. Vue interne de la méthode de diagnostic à base de modèle mise en œuvre.
Figure C.2. Discordances associées à un signal d'entrée de la partie contrôle.
Figure D.1. Prototype SURFER embarqué et visualisation des informations au sol.

Annexes

Annexe A

Complément sur l'approche systémique

Selon l'AFSCET (Association Française de Science des Systèmes Cybernétiques Cognitifs et Technique), la systémique est définie comme une "*nouvelle discipline qui regroupe les démarches théoriques, pratiques et méthodologiques, relatives à l'étude de ce qui est reconnu comme trop complexe pour pouvoir être abordé de façon réductionniste, et qui pose des problèmes de frontières, de relations internes et externes, de structure, de lois ou de propriétés émergentes caractérisant le système comme tel, ou des problèmes de mode d'observation, de représentation, de modélisation ou de simulation d'une totalité complexe*".

La triangulation systémique regroupe les visions fonctionnelle, organique et historique du système étudié dans le but d'approfondir sa compréhension (Le Moigne, 1994) :

- vision fonctionnelle : elle décrit la finalité du système dans son environnement (ce que fait le système, ce pour quoi il est fait),
- vision organique : elle représente l'agencement des organes du système, leurs structures et leurs relations (comment se compose le système, de quoi est-il fait),
- vision historique : elle retrace l'histoire du système, suivant sa nature évolutive, ses mécanismes d'apprentissage, d'adaptation et d'auto-organisation, en y intégrant une dimension temporelle (comment évolue le système, que devient-il).

Suivant les choix de conception du système étudié, les fonctions du système et sa structure peuvent être étroitement liés. Cette propriété dépend des solutions retenues par les concepteurs lors de l'allocation des fonctions sur les organes d'un système. De ce fait, un organe peut supporter une unique fonction, un ensemble de fonctions, ou plusieurs organes sont nécessaires pour assurer une fonction. Ainsi, une décomposition fonctionnelle s'avère équivalente à une décomposition structurelle dans le cas où chaque fonction du système est supportée par un seul organe.

Annexe B

Etude comparative des méthodes de diagnostic

Le diagnostic constitue un vaste domaine de Recherche, dont les méthodes sont principalement issues de trois communautés : la communauté Automaticienne (FDI : *Fault Detection and Isolation*), la communauté des Systèmes Experts - Intelligence Artificielle (DX : *Diagnosis eXpert system*), et la communauté des Systèmes à Evénements Discrets (DES : *Discrete-Event System*) (Feldman, 2010 ; Knotek, 2006 ; Simeu-Abazi et al., 2010). L'efficacité d'une méthode de diagnostic est variable d'un système à un autre et aucune méthode ne présente un caractère universel (Rao et al., 1998). Une méthode de diagnostic doit donc être rigoureusement choisie pour un certain problème de diagnostic et être adaptée au système à diagnostiquer (systèmes continus, discrets ou hybrides, par exemple) (Larsson, 1999). Les méthodes présentées dans cette annexe se concentrent sur le diagnostic de défaillances physiques d'un système sous surveillance. Les travaux de Recherche sur le diagnostic de défaillances logicielles sont relativement récents et ne sont pas abordés dans nos travaux (Schumann et al., 2013). Cette annexe recense les principales méthodes de diagnostic développées par ces trois communautés et propose une étude comparative de ces méthodes.

Les méthodes de diagnostic visent à déterminer les fautes ayant conduit à des défaillances et peuvent être classées suivant la connaissance du système sous surveillance qu'elles exploitent. Selon les classifications proposées par (Katipamula and Brambley, 2005 ; Venkatasubramanian et al., 2003a), les méthodes de diagnostic peuvent être regroupées en deux principales catégories : les méthodes à base de modèles et les méthodes basées sur des données historiques, nommées respectivement par la suite MBD (*Model-Based Diagnosis*) et HBD (*History-Based Diagnosis).* Dans chaque catégorie, la connaissance exploitée peut être de nature quantitative ou qualitative (figure B.1).

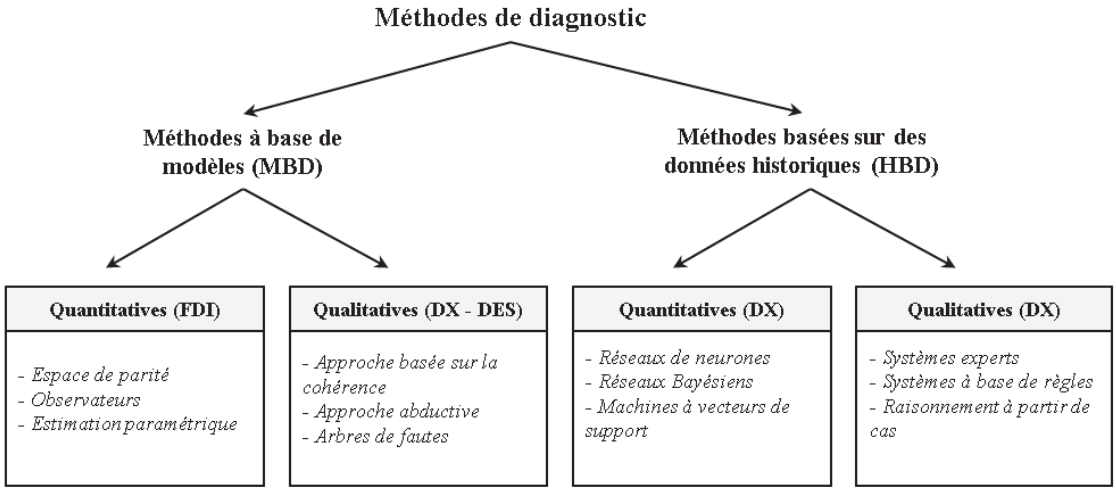

Figure B.1. Classification des méthodes de diagnostic.

Les méthodes MBD impliquent une phase de modélisation orientée bon ou mauvais fonctionnement du système sous surveillance. Ces méthodes exigent généralement une connaissance profonde pour établir un modèle du système à diagnostiquer (connaissance structurelle, comportementale, fonctionnelle ou téléologique, par exemple). Contrairement aux méthodes MBD, les méthodes HBD ne nécessitent pas de modèles, mais elles supposent que des données de retours d'expériences importantes du système soient disponibles. Les méthodes importantes associées à ces deux catégories sont présentées ci-après.

B.1 Méthodes à base de modèles (MBD)

Lorsque des données de retours d'expériences sont insuffisantes ou inexistantes, les méthodes MBD permettent de diagnostiquer un système à partir d'un modèle de son comportement. Ce modèle peut notamment être établi grâce aux informations de conception disponibles (spécifications fonctionnelles, schémas électrique et mécanique du système, par exemple). Les méthodes MBD consistent à comparer le comportement réel d'un système avec un comportement inféré par un modèle de ce système. Par exemple, un modèle de bon fonctionnement permet de prédire le comportement correct et exigé d'un système sous surveillance lors de son fonctionnement. Un écart entre le comportement prédit par le modèle et le comportement observé témoigne d'une anomalie de fonctionnement (symptôme), et révèle la présence d'une faute pouvant aboutir à une défaillance. Ces comportements anormaux sont par la suite analysés dans le but de localiser les fautes. Les méthodes MBD regroupent ainsi les fonctions de détection et de localisation de défaillances (figure B.2).

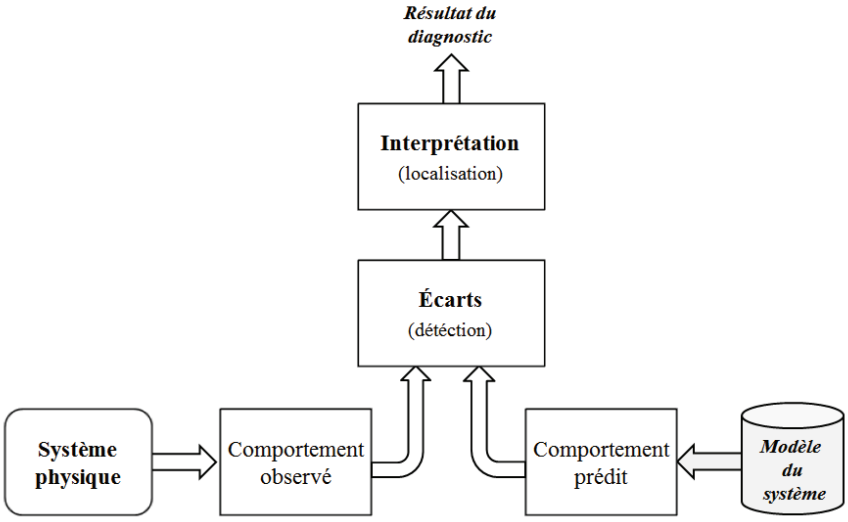

Figure B.2. Principe des méthodes de diagnostic à base de modèles.

La communauté FDI prône l'utilisation de modèles quantitatifs basés sur des lois fondamentales. Ces modèles analytiques décrivent le comportement de systèmes continus ou hybrides par des équations mathématiques (équations différentielles associées à des lois physiques, par exemple). Par ailleurs, les communautés DX et DES développent des modèles qualitatifs, exprimant le comportement d'un système sous forme de relations qualitatives. Les approches proposées par les communautés FDI et DX sont notamment comparées dans (Cordier et al., 2004). Les principales méthodes de diagnostic à base de modèles sont détaillées ci-après.

B.1.1 Méthodes basées sur des modèles quantitatifs (FDI)

Les méthodes FDI se concentrent principalement sur la génération de symptômes de dysfonctionnements pertinents, nommés *résidus*, à partir d'un modèle de bon fonctionnement du système sous surveillance (comportement nominal). La procédure de diagnostic se déroule en deux grandes étapes (Durand, 2011 ; Kempowsky, 2004).

- Génération de résidus : elle consiste à comparer le comportement théorique du système sous surveillance fourni par un modèle avec son comportement réel. Il s'agit donc de mesurer les différences entre des grandeurs collectées sur le système et des valeurs estimées par le modèle. Cette étape permet de produire des signaux porteurs de symptômes, généralement exprimés par un *vecteur de résidus* (Franck, 1996).

- Décision : les résidus générés sont évalués pour confirmer l'occurrence d'une faute (principe de seuillage fixe ou adaptatif, par exemple) et ensuite analysés afin de localiser la faute et identifier sa nature et sa cause. Les activités associées à cette étape sont la détection, la localisation et l'identification (Gertler, 1998) :
    - la détection signale la présence d'une faute avérée sur le système (valeur d'un résidu dépassant un seuil à une certaine date, par exemple),
    - la localisation incrimine des éléments défaillants du système sous surveillance,
    - l'identification détermine les caractéristiques de la faute (type, amplitude et cause de la faute).

En l'absence de fautes sur le système, les résidus générés par la première étape ont une valeur théoriquement nulle (hypothèses d'une modélisation parfaite et de l'absence de bruits de mesure, par exemple), et non nulle en présence de fautes. La phase de génération de résidus consiste à générer un vecteur noté r (sortie du bloc repéré 1 sur la figure B.3), qui repose généralement sur les signaux observables d'entrées (vecteurs de commandes) et de sorties (vecteur de mesures) du système physique sous surveillance à un instant t, notés respectivement $u(t)$ et $y(t)$.

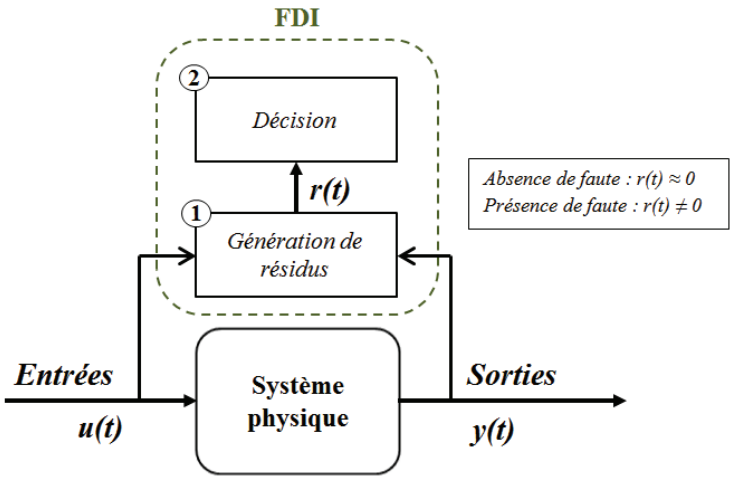

Figure B.3. Méthode de diagnostic à base de modèles quantitatifs (FDI).

Les méthodes usuelles proposées par la communauté FDI pour la génération de résidus sont l'espace de parité, les observateurs et l'estimation paramétrique (Isermann, 2005) :

- Espace de parité : cette méthode exploite le modèle du système à diagnostiquer afin d'établir des relations de redondances analytiques (RRA). Ces relations expriment les entrées et les sorties du système en éliminant les variables d'états inconnues (non observables) du modèle (Maquin et al., 1997). De ce fait, les RRA permettent de vérifier directement la cohérence des entrées-sorties du système à partir de variables mesurables. La forme générique du vecteur de résidus est donnée par l'équation B.1 ci-dessous.

$$r(t) = \phi(u(t), y(t)) \qquad \text{(B.1)}$$

- Observateurs : la méthode des observateurs permet d'évaluer les variables de sorties du système à partir des variables d'états obtenues par des observateurs (Franck, 1996). Les sorties réelles du système sont ensuite comparées aux sorties estimées par le modèle. Le vecteur de résidus correspond alors à la différence entre les valeurs des sorties observées sur le système et les valeurs des sorties fournies par le modèle, notées $\hat{y}$ (équation B.2).

$$r(t) = y(t) - \hat{y}(t) \qquad \text{(B.2)}$$

- Estimation paramétrique : cette méthode consiste à estimer en permanence un vecteur de paramètres basé sur les signaux d'entrées-sorties du système lors de son fonctionnement (grandeurs physiques du système, par exemple). Ce vecteur est ensuite comparé à un vecteur nominal de paramètres qui caractérise le comportement normal du système. La distance séparant les paramètres estimés des paramètres de référence du système permet de conclure sur l'état du système (Balle et Isermann, 1998). Le vecteur de résidus est obtenu en faisant la différence entre les paramètres nominaux notés $\tilde{\theta}$ et les paramètres estimés notés $\hat{\theta}$ (équation B.3).

$$r(t) = \tilde{\theta} - \hat{\theta}(t) \qquad \text{(B.3)}$$

A partir du vecteur de résidus généré, l'étape de décision permet de déceler un comportement anormal significatif du système surveillé (*fault detection*) et de localiser (isoler) la faute (*fault isolation*). Lorsqu'une faute est détectée, la localisation peut être réalisée suivant deux approches : les résidus structurés et les résidus directionnels (Patton et Chen, 1997). Les résidus structurés sont déterminés de telle sorte que chaque résidu soit sensible à un sous-ensemble de fautes connues et insensible aux autres fautes. Ainsi, certains résidus sont différents de zéro pour une faute donnée, tandis que les autres résidus ont une valeur nulle. Le type de faute peut alors être discriminé par une table nommée *matrice de signatures*, traduisant l'impact des fautes sur les résidus (figure B.4).

	F_1	F_2	F_3	F_4
r_1	1	1	0	0
r_2	1	1	1	0
r_3	0	0	1	1

Figure B.4. Exemple de matrice de signatures, adaptée de (Domlan, 2006).

Par exemple, dans la matrice de signature illustrée sur la figure B.4, les fautes F_1 et F_2 ont un impact sur les résidus r_1 et r_2 et n'influencent pas le résidu r_3. Par ailleurs, les fautes F_1 et F_2 ne sont pas "isolables" car ces deux fautes présentent la même signature. Dans la seconde approche, les résidus directionnels sont conçus de manière à orienter le vecteur de résidus selon chaque type de faute présente sur le système. Par conséquent, un vecteur de résidus spécifique doit être établi pour chacune des fautes. La direction prise par le vecteur de résidu doit permettre de discriminer le type de faute.

B.1.2 Méthodes basées sur des modèles qualitatifs (DX-DES)

Les modèles qualitatifs sont utilisés pour décrire le comportement de systèmes continus, discrets ou hybrides en considérant la structure du système, les éléments qui le composent et les relations entre ces éléments (Hamscher, 1992 ; Venkatasubramanian et al., 2003b). Les deux principales méthodes à base de modèle proposées par la communauté DX sont l'approche basée sur la cohérence (*consistency-based approach*) et l'approche abductive (*abductive approach*) :

- Approche basée sur la cohérence : elle repose uniquement sur des modèles du comportement normal du système, qui n'intègrent pas la connaissance de son comportement en présence de fautes (*weak-faults models*). Cette approche vise à identifier les composants responsables des incohérences détectées entre le comportement prédit par le modèle et le comportement observé (De Kleer et Williams, 1987).

- Approche abductive : elle exploite des modèles du comportement du système affecté de fautes (*strong-fault models*). Cette connaissance recense alors les fautes et les symptômes de dysfonctionnements du système. L'approche abductive consiste à identifier les fautes qui, lorsqu'elles sont supposées présentes, permettent de prédire les comportements anormaux observés (Poole, 1994).

L'approche basée sur la cohérence est largement utilisée par la communauté DX. Elle est fondée sur une théorie logique du diagnostic nommée diagnostic issu des principes premiers (*diagnosis from first principles*) (Reiter, 1987). Suivant cette approche, le comportement correct d'un système à diagnostiquer est prédit grâce à une description explicite de ce système (modèle du comportement nominal des composants et de la structure du système, par

exemple) (Davis, 1984). Le comportement prédit par le modèle est ensuite confronté au comportement réel du système, tel qu'il est observé au travers de signaux de mesures. Les écarts entre les observations du système et les prédictions issues du modèle représentent des symptômes de dysfonctionnements nommés discordances (*discrepancies*). Contrairement aux méthodes FDI, les écarts de comportements détectés apportent des précisions sur la localisation de la faute. Par exemple, un modèle peut permettre d'établir une prédiction sur la base de composants notés C_1 à C_n définis dans le modèle. Lorsque cette prédiction est en contradiction avec une observation, la discordance générée signifie qu'au moins un des composants C_1 à C_n est défectueux. Ce groupe de composants forment alors un *conflit*. La première étape de la procédure de diagnostic consiste à générer des discordances (ensemble de conflits) à partir des prédictions et des observations (bloc repéré 1 sur la figure B.5). La deuxième étape permet d'analyser ces différents conflits et d'émettre des hypothèses sur le mode de fonctionnement des composants suspectés (mode correct ou défaillant, par exemple). Cette étape recherche donc les modes de dysfonctionnements de composants pour lesquels tous les conflits sont éliminés. Un ensemble de composants (candidats) qui permet de rétablir une cohérence avec les observations lorsqu'ils sont supposés être dans un mode de dysfonctionnement est alors un résultat de diagnostic (sortie du bloc repéré 2 sur la figure B.5).

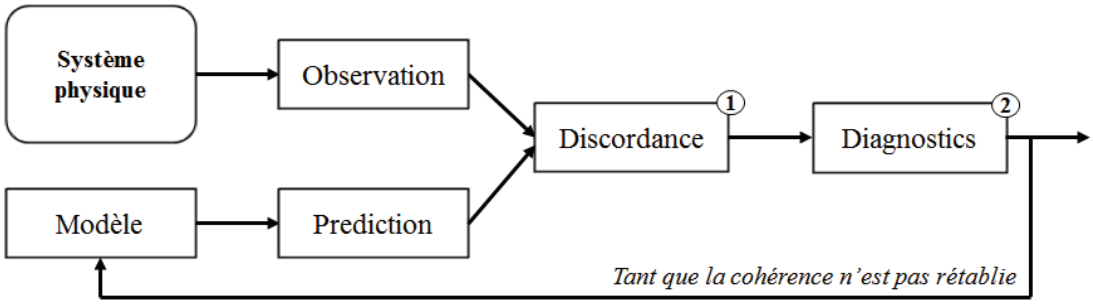

Figure B.5. Approche de diagnostic basée sur la cohérence (DX), adaptée de (Feldman, 2010).

Dans cette approche étendue et formalisée par (De Kleer et al., 1992), un système physique sous surveillance est un triplet $\{SD, COMPS, OBS\}$, dans lequel SD est la description du système (ensemble de formules de la logique des prédicats du premier ordre avec égalité), $COMPS$ représente les composants du système (ensemble fini de constantes) et OBS désigne les observations du système (ensemble de formules du premier ordre avec égalité). Le comportement anormal d'un composant $C \in COMPS$ est noté $AB(C)$ et la négation de ce terme, notée $\neg\, AB(C)$, désigne le comportement normal de ce composant (*not ABnormal*). Le diagnostic consiste alors à affecter les modes de fonctionnements AB et $\neg AB$ aux composants suspectés afin de rétablir la cohérence entre la description du système et les observations. Les composants présents dans COMPS sont donc les éléments incriminés par l'algorithme de diagnostic. Notons que ces composants figurent également dans la description SD du système et éventuellement dans les observations OBS (signaux de mesures et de commandes du système, par exemple). Lorsque le système surveillé ne présente aucune faute et en faisant l'hypothèse que chaque composant du système fonctionne normalement, les observations OBS sont cohérentes avec la description SD du système. La phase de génération de discordance vérifie que la relation suivante est cohérente (la relation est dite *consistante*) :

$$SD \cup \{\neg AB(C_1), \ldots, \neg AB(C_n)\} \cup OBS \quad \text{(B.4)}$$

Un diagnostic pour $\{SD, COMPS, OBS\}$ est un ensemble de composants noté $D \subseteq COMPS$ déterminé de manière à ce que l'expression suivante soit *satisfiable* :

$$SD \cup OBS \cup \{AB(C) | c \in D\} \cup \{\neg AB(C) | c \in COMPS - D\} \quad \text{(B.5)}$$

Le nombre de composants candidats générés par l'algorithme de diagnostic est souvent exponentiel en nombre de composants (Feldman, 2010). Par conséquent, un *diagnostic minimal*, noté D' tel que $D' \subset D$, est préférable. Ce diagnostic minimal peut être obtenu à l'aide de différents critères utilisés pour réduire le nombre de candidats (nombre minimal de composants simultanément défaillants, probabilités des défaillances associées aux composants, par exemple).

Cette approche basée sur la cohérence peut également être adaptée pour diagnostiquer des systèmes à événements discrets, à partir d'une modélisation orientée bon fonctionnement du système (Lunze et al., 2001 ; Pandalai and Holloway, 2000). Dans la communauté DES, les modèles intègrent généralement le comportement nominal et en présence de fautes du système surveillé. Par exemple, l'approche *diagnostiqueur* proposée par (Sampath et al., 1996) repose sur une modélisation du comportement normal et anormal des différents composants du système par des états observables et non-observables. Par ailleurs, les arbres de fautes ou FTA (*Fault Tree Analysis*) sont couramment utilisés pour recenser les défaillances pouvant survenir dans un système. Ces modèles de causes à effets sont établis lors des études de sureté de fonctionnement des systèmes et ne constituent pas réellement une méthode de diagnostic (Ribeiro et Barata, 2011). Un arbre de faute illustre graphiquement les relations entre des événements de base (fautes) qui sont susceptibles de conduire à un événement redouté situé au sommet de l'arbre (défaillance). Les événements sont liés entre eux par des nœuds symbolisant des fonctions logiques (fonctions ET, OU, par exemple), qui produisent des événements de niveau supérieur. Dans (Lefebvre et al., 2000), les auteurs proposent notamment un modèle basé sur des arbres de fautes dynamiques pour le diagnostic d'équipements avioniques. Ces arbres de fautes sont construits avec des fonctions logiques évoluées qui considèrent les aspects temporels entre les événements observés pour localiser les fautes.

B.2 Méthodes basées sur des données historiques

Les méthodes sans modèles visent à capitaliser des données historiques acquises sur le système afin d'en extraire une connaissance suffisante pour élaborer un diagnostic. L'extraction de cette connaissance peut être réalisée de manière qualitative ou quantitative (Venkatasubramanian et al., 2003c). La plupart de ces méthodes sont issues de l'Intelligence Artificielle (IA), telles que les réseaux de neurones, les réseaux Bayésien ou les machines à vecteurs de support de nature quantitative. Les systèmes experts ou à base de règles et les approches de raisonnement à partir de cas sont des méthodes qualitatives.

B.2.1 Réseaux de neurones

Les réseaux de neurones artificiels sont constitués de plusieurs éléments de calcul interconnectés et organisés en couches qui imitent la structure de neurones biologiques. Un réseau de neurone ou ANN (*Artificial Neural Network*) est capable de modéliser des relations complexes entre des signaux d'entrées et de sorties et également de réaliser des tâches de reconnaissance et de classification de données. Lorsqu'ils sont utilisés à des fins de diagnostic, les réseaux de neurones sont attrayants pour leurs capacités d'apprentissage et de classification (localisation des fautes). Différents modèles de réseaux de neurones sont exploités pour réaliser le diagnostic tels que les réseaux MLP (*multi-layer perceptrons*) et RBF (*radial basis function*) (Yang et al., 2002).

B.2.2 Réseaux Bayésiens

Un réseau Bayésien ou BN (*Bayesian Network*) est un graphe orienté acyclique dans lequel chaque nœud du réseau représente un événement (état du système, par exemple), et dont les arcs entre les nœuds sont associés à des probabilités conditionnelles. De par sa nature probabiliste, un réseau Bayésien, également nommé réseau de croyance, permet de considérer les incertitudes dans la représentation d'une connaissance. Par exemple, un réseau Bayésien peut capturer les liens entre les symptômes de dysfonctionnements et les modes de défaillances des composants d'un système. Les probabilités associées aux défaillances peuvent être renseignées par les experts du système. De plus, les réseaux Bayésiens ont des capacités d'apprentissage et peuvent être utilisés pour la tâche de classification dans la procédure de diagnostic (Ferreiro et al., 2012).

B.2.3 Machines à vecteur de support

Les machines à vecteurs de support ou SVM (*Support Vector Machine*) sont des algorithmes de classification par apprentissage développés initialement pour repartir des données en deux classes. Dans un premier temps, une phase d'apprentissage est réalisée sur la base de deux ensembles de données. Cette phase consiste à déterminer un classificateur (hyperplan) défini par un certain nombre de vecteurs de support et qui permet de dissocier les deux jeux de données avec une distance optimale. Par la suite, une fonction de décision réalise une classification des nouvelles données à l'aide de ce classificateur. Dans une procédure de diagnostic, cette méthode peut être utilisée pour identifier les différentes fautes ou classes de défaillances d'un système en augmentant le nombre de classes (Widodo et Yang, 2007).

B.2.4 Systèmes experts

Un système expert est un programme informatique reproduisant le comportement cognitif d'un expert humain pour résoudre certains problèmes d'un domaine particulier (Racoceanu, 2006). Un tel système est constitué d'une base de connaissances et d'un moteur d'inférence, capable d'explorer cette base de connaissance et d'émettre des conclusions à partir de faits (Venkatasubramanian et al., 2003c). La base de connaissance est élaborée à partir d'une expertise humaine et constitue un savoir-faire du domaine considéré. Le diagnostic élaboré par un système expert repose sur un ensemble de règles qui déterminent les causes d'une défaillance en fonction des symptômes observés et d'une liste de fautes répertoriées.

B.2.5 Systèmes à base de règles

Un système à base de règles ou RBS (*Rule-Based System*) est un système expert particulier dans lequel l'expertise humaine est représentée sous forme de règles prédéfinies (assertions de type "Si-Alors", par exemple) (Hayes-Roth, 1985). Cette base de règles est exploitée par un moteur d'inférence qui consiste à rechercher les règles actives selon les symptômes observés (ou ensemble de faits). Ce moteur d'inférence élabore un diagnostic par un mécanisme de chaînage avant. Un exemple de système expert à base de règles pour le diagnostic de véhicules routiers est proposé dans (Salama et al., 2012). Les auteurs proposent un arbre de décision pour représenter les relations entre les différentes règles implémentées (relations entre les faits et les fautes). Cet arbre permet de sélectionner la règle appropriée lorsqu'une règle implique plusieurs règles de niveaux inférieurs.

B.2.6 Raisonnement à partir de cas

Dans un système de raisonnement à partir de cas ou CBR (*Case-Based Reasoning*), une base de connaissance contenant un ensemble d'expériences de problèmes passés (ou cas) est utilisée pour résoudre de nouveaux problèmes. Le principe de cette méthode est analogue à celui d'un expert humain, qui raisonne sur un problème à partir de ses connaissances antérieures acquises. Une propriété importante des méthodes CBR est leur capacité d'apprentissage (Berenji et al., 2005). Contrairement aux systèmes à base de règles, la base de connaissance est enrichie automatiquement par l'expérience. Le comportement anormal du système surveillé est comparé aux situations déjà rencontrées par le passé. Un nouveau problème de diagnostic peut alors être résolu à partir des cas mémorisés dont la solution est connue ou demeure inconnue. Cette situation peut également être mémorisée en tant que nouvelle expérience et être complétée par les conclusions d'un expert. Un exemple de système CBR évolué combinant différentes bases de connaissances est étudié dans (Kamsu-Foguem et Noyes, 2013).

B.3 Comparaison des méthodes de diagnostic

Les méthodes de diagnostic abordées précédemment peuvent être comparées à partir d'un ensemble de caractéristiques communes. Cette étude comparative vise à orienter le concepteur d'une architecture de surveillance dans le choix des méthodes diagnostic à implémenter. Afin de comparer la performance des méthodes de diagnostic, cinq caractéristiques issues des exigences d'un système de surveillance considérées dans nos travaux ont été retenues. Ces exigences de performance du diagnostic concernent la précision, la facilité d'explication, l'adaptabilité, la réactivité et la confiance du diagnostic. De plus, une exigence supplémentaire est introduite pour évaluer la compatibilité des méthodes de diagnostic avec une implémentation embarquée du diagnostic. Cette exigence fait intervenir les capacités de calcul et de stockage pour établir un diagnostic à bord d'un système cible et suppose également un traitement *en-ligne* de la méthode de diagnostic.

Les méthodes de diagnostic à base de modèles reposent sur une modélisation du système à diagnostiquer en un point de fonctionnement considéré. Un modèle est donc associé à un domaine de validité (système modélisé dans un mode de fonctionnement spécifique, par exemple). Lorsqu'un modèle précis du système à diagnostiquer peut être obtenu, les méthodes MBD quantitatives sont performantes en termes de précision, de réactivité et de confiance du

diagnostic (Ribeiro et Barata, 2011). Cependant, ces méthodes ont une capacité explicative quasiment inexistante et sont difficilement adaptables aux changements survenant dans le système diagnostiqué (modification de composants du système, par exemple). Par ailleurs, les algorithmes associés à ces méthodes peuvent être complexes, ce qui entraine une augmentation de la puissance de calcul nécessaire à leur exécution (Venkatasubramanian et al., 2003a).

Les méthodes MBD qualitatives permettent d'élaborer un diagnostic précis avec un temps de calcul acceptable et une consommation de mémoire raisonnable (Azarian et al., 2011). De plus, les modèles utilisés sont généralement agencés de façon modulaire, de manière à faciliter la conception et la réutilisation de librairies de composants (Chittaro and Ranon, 1999). Ces méthodes apportent également des informations explicatives sur les résultats du diagnostic. Néanmoins, l'élaboration d'un modèle qualitatif exhaustif d'un système physique peut s'avérer difficile (Zaki et al., 2007). Les arbres de fautes (FTA) reposent sur une analyse de fiabilité du système à diagnostiquer, qui influence largement la précision du diagnostic (Ribeiro et Barata, 2011). Les arbres de fautes présentent une facilité d'explication, une réactivité et une confiance importante, mais sont sujets à des erreurs de développements par leurs concepteurs (Venkatasubramanian et al., 2003a). Aussi, un arbre de faute est généralement utilisé comme un outil d'analyse *hors-ligne* et ne peut être considéré comme une méthode de diagnostic *en-ligne*.

Les réseaux de neurones (ANN) élaborent des diagnostics pertinents avec une réactivité importante après la phase d'apprentissage. Les limitations de ces méthodes résident dans la facilité d'explication des résultats de diagnostic et dans l'adaptabilité des algorithmes (Ribeiro and Barata, 2011; Venkatasubramanian et al., 2003b).

Les réseaux Bayésien (BN) offrent une précision importante à condition de disposer d'une connaissance suffisante pour renseigner les probabilités conditionnelles des réseaux (Ribeiro and Barata, 2011). Par ailleurs, ces méthodes dépendent du système à diagnostiquer et manquent d'adaptabilité. De plus, lorsque la dimension du réseau est importante, une puissance de calcul élevée est nécessaire. Certaines méthodes proposent alors un compromis entre la précision du diagnostic et la complexité de l'algorithme afin de réduire le temps de calcul (Ferreiro et al., 2012).

Les machines à vecteurs de supports (SVM) sont des méthodes très précises pour la classification de fautes et présentent une réactivité importante. Cependant, ces méthodes nécessitent une connaissance à priori des caractéristiques des fautes et exigent une phase d'apprentissage à chaque changement survenant dans le système surveillé. Aussi, elles requièrent une taille mémoire plus importante lorsque le nombre de classes est élevé (Widodo and Yang, 2007).

Les systèmes experts à base de règles (RBS) présentent une forte réactivité lorsque des règles permettent de relier directement les symptômes observés à des fautes. Ces méthodes ont également la capacité de fournir des explications sur les solutions proposées (Venkatasubramanian et al., 2003b). Toutefois, il est impossible de diagnostiquer des fautes non répertoriées (Zaki et al., 2007). De plus, l'adaptabilité de ces méthodes est limitée car les règles doivent être réécrites lorsque des modifications sont apportées au système (Ribeiro et Barata, 2011).

Les méthodes CBR fournissent des résultats précis en un temps acceptable, mais nécessitent une expérience suffisante sur le système à diagnostiquer. Une mémoire importante est généralement nécessaire pour stocker les expériences passées. Par ailleurs, une méthode CBR est spécifique à un certain type de système et délivre peu d'explications sur les résultats obtenus (Azarian et al., 2011).

Inspiré des travaux de (Dash et Venkatasubramanian, 2000 ; Ribeiro et Barata, 2011 ; Ribeiro, 2012 ; Venkatasubramanian et al., 2003c), une grille d'évaluation des méthodes de diagnostic est proposée ci-dessous en considérant les caractéristiques retenues dans nos travaux (figure B.6).

Exigences	Caractéristiques	Méthodes MBD			Méthodes HBD				
		Quantitatives (FDI)	Qualitatives (DX-DES)	FTA	ANN	BN	SVM	RBS	CBR
Performance	Précision	+	++	+	+	+	++	+	++
	Facilité d'explication	-	++	++	-	+	+	+	-
	Adaptabilité	-	+	-	-	-	+	-	-
	Réactivité	+	+	+	++	+	+	++	++
	Confiance	++	+	++	++	++	+	+	++
Diagnostic embarqué	Capacités de calcul et de stockage	+	+	++	++	+	+	++	++
	Traitement en-ligne	OUI	OUI	NON	OUI	OUI	OUI	OUI	OUI

Légende

- : Caractéristique défavorable ou absente + : Caractéristique acceptable ++ : Caractéristique favorable

Figure B.6. Grille d'évaluation des méthodes de diagnostic proposée.

Comme évoqué précédemment, une analyse comparative des méthodes de diagnostic doit permettre de privilégier une méthode pour un système donné. En pratique, certaines méthodes de diagnostic sont inapplicables pour diagnostiquer un système et d'autres méthodes sont à proscrire. A titre d'exemple, une méthode de type HBD peut convenir à un système pour lequel un historique important des défaillances survenues dans le passé est disponible.

Dans le cadre applicatif de nos travaux, les données de retours d'expériences sont insuffisantes pour envisager la mise en œuvre de méthodes de type HBD. Parmi les méthodes de type MBD, les arbres de fautes (FTA) ne peuvent être implémentés en-ligne à bord du système cible et sont plutôt dédiés à une utilisation hors-ligne, par exemple, dans un centre de maintenance. Les méthodes MBD qualitatives ont potentiellement la capacité de fournir des résultats de diagnostic précis et sont compatibles avec une implantation embarquée du diagnostic (Console et Dressier, 1999). Par ailleurs, ces méthodes apportent des explications sur les résultats de diagnostic fournis. Ainsi, une méthode MBD qualitative s'avère pertinente dans le cadre applicatif de nos travaux. La méthode MBD qualitative mise en œuvre est détaillée dans l'annexe suivante.

Annexe C

Méthode de diagnostic mise en œuvre

La méthode de diagnostic MBD qualitative mise en œuvre a été développée par la société PROSYST. Cette méthode nommée *Diagnostic Automatique* analyse les signaux discrets échangés entre la partie contrôle et la partie sous-contrôle d'un système pour élaborer le diagnostic de la partie sous-contrôle (figure C.1).

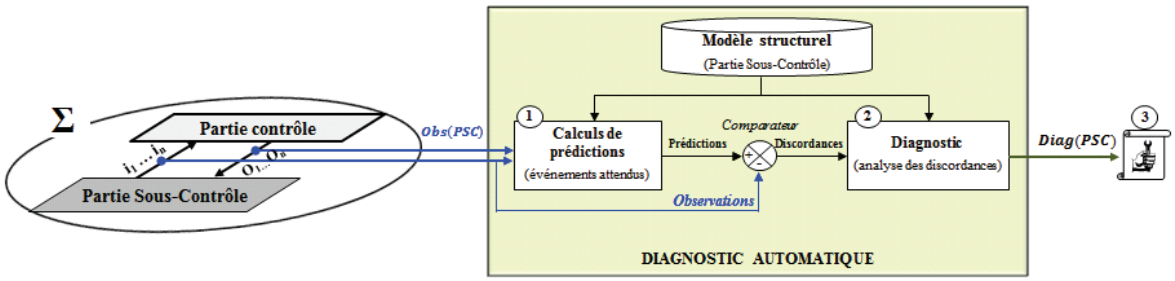

Figure C.1. Vue interne de la méthode de diagnostic à base de modèle mise en œuvre.

Lorsque le système sous surveillance est sollicité, le modèle est initialisé et évolue de manière synchrone avec le comportement observé du système. Dans un premier temps, ce modèle est utilisé pour calculer un ensemble de prédictions sur la base des signaux collectés sur le système (bloc repéré 1 sur la figure C.1). Les événements prédits résultent généralement de mouvements mécaniques commandés par le contrôleur qui conduisent à un ou plusieurs changements d'états de signaux de mesures (signaux en entrée du contrôleur). Ces événements attendus sont ensuite comparés aux signaux de mesures observés (comparateur symbolisé en sortie du bloc 1 sur la figure C.1). Une discordance est générée lorsqu'un signal d'entrée prédit diffère du signal d'entrée observé. La détection d'une discordance indique la présence d'une faute dans la partie opérative du système.

Dans un deuxième temps, un algorithme de diagnostic analyse les discordances en utilisant une représentation dynamique du système établie grâce au modèle (bloc repéré 2 sur la figure C.1). Cet algorithme émet des hypothèses sur l'état des composants de la partie opérative et génère une liste préliminaire de composants suspects qui pourraient expliquer les discordances détectées. Cette liste peut évoluer en fonction des observations futures du système. En effet, les événements observés après l'apparition d'une discordance peuvent donner lieu à la suppression de un ou plusieurs composants suspects. Un composant suspecté est conservé tant que les événements observés ne reflètent pas son comportement normal. Cette procédure permet d'affiner la liste de composants suspects associée à une discordance. Le résultat d'un diagnostic délivré par cette méthode (bloc repéré 3 sur la figure C.1) correspond à une liste de discordances, dans laquelle chaque discordance est accompagnée d'un ensemble de composants suspects.

Une discordance est associée à un signal d'entrée de la partie contrôle. Dans le système sous surveillance considéré, ce signal peut prendre deux valeurs possibles traduisant deux états logiques. La relation entre la valeur de ce signal et l'état physique d'éléments de partie opérative dépend de la conception électrique-électronique du système (entrée reliée à un capteur fin de course, par exemple). Deux cas de figure peuvent se présenter lors de la génération de discordances. Une discordance est détectée lorsqu'un événement est observé mais n'a pas été prédit (événement non prévu), ou lorsqu'un événement est prédit mais n'a pas été observé (absence d'événement). Par conséquent, une discordance de type événement non prévu est détectée sur un front montant ou descendant d'un signal d'entrée, tandis qu'une discordance de type absence d'événement est détectée uniquement sur un état logique stable. Chaque discordance est caractérisée par son type (événement non prévu front montant, par exemple) et sa date d'occurrence (figure C.2).

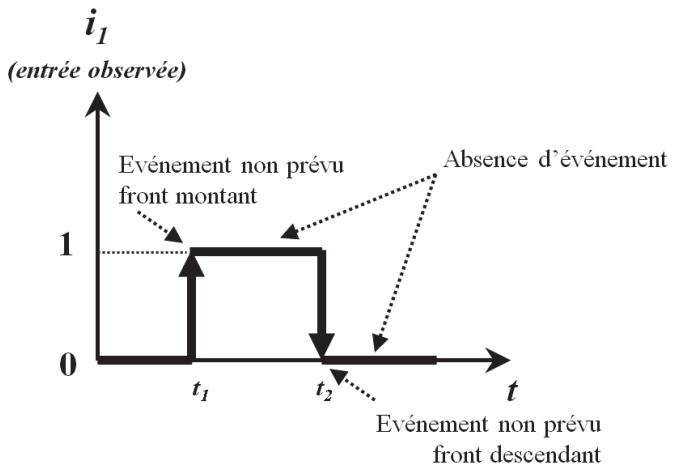

Figure C.2. Discordances associées à un signal d'entrée de la partie contrôle.

Plusieurs discordances peuvent être détectées pour une même défaillance survenant dans la partie opérative du système. Cette méthode de diagnostic fait l'hypothèse qu'un seul composant de la partie opérative puisse être défaillant. La défaillance d'un composant de la partie opérative doit alors justifier l'ensemble des discordances détectées. La liste de composants suspects minimale associée à une discordance primaire est donc une information pertinente pour un opérateur de maintenance.

Annexe D

Prototype SURFER

Trois prototypes ont été conçus et implantés à bord de trains réels en exploitation dans le cadre du consortium SURFER. Ces prototypes supportent les modules logiciels développés par les différents partenaires du projet. Chaque prototype est équipé d'une liaison sans fil qui assure une communication bidirectionnelle avec le système SURFER situé au sol (figure D.1).

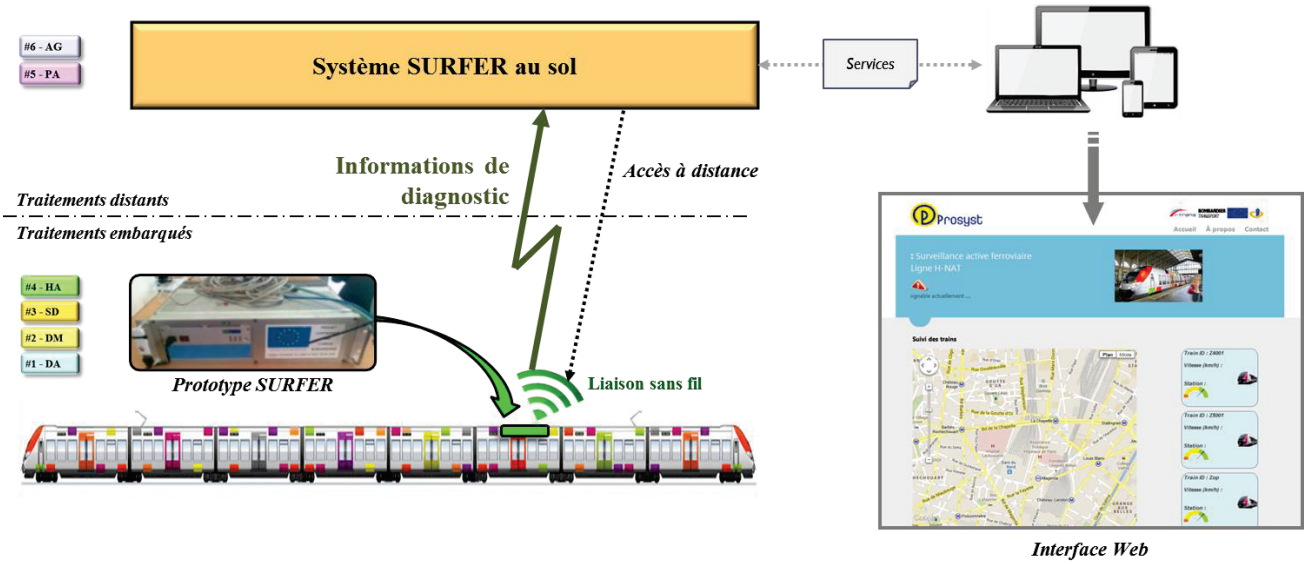

Figure D.1. Prototype SURFER embarqué et visualisation des informations au sol.

Les informations de surveillance fournies par ces prototypes sont accessibles à travers une interface Web dédiée. Cette interface permet de visualiser des informations de contexte des trains et de leurs sous-systèmes, ainsi que les événements de diagnostic pertinents élaborés par les modules logiciels embarqués.

UNIVERSITE DE VALENCIENNES ET DU HAINAUT CAMBRESIS
LABORATOIRE TEMPO - EA4542
LE MONT HOUY - 59313 VALENCIENNES CEDEX 9

Antoine LE MORTELLEC

Proposition d'une architecture de surveillance "active" à base d'agents intelligents pour l'aide à la maintenance de systèmes mobiles - Application au domaine ferroviaire

Directeur de thèse : Pr. Damien TRENTESAUX, TEMPO-PSI
Co-encadrants de thèse : Dr. Yves SALLEZ, TEMPO-PSI – Dr. Thierry BERGER, TEMPO-PSI
Spécialité : Automatique et Génie Informatique

Résumé : ces deux dernières décennies, les systèmes embarqués ont été introduits dans de nombreux domaines d'application (transport, industrie, habitat, médical...). Ces systèmes se sont vu confier des tâches plus importantes pour délivrer de nouveaux services aux utilisateurs avec des délais de mise sur le marché toujours plus courts et à moindre coût. L'intégration rapide de ces systèmes au sein de produits manufacturés est un avantage concurrentiel pour les industriels. Cependant, les pannes associées à ces systèmes et le niveau de complexité croissant des équipements ont rendu les interventions de maintenance bien plus délicates. L'identification des causes de certaines pannes représente actuellement un véritable challenge dans les activités de la maintenance. Elles entrainent une indisponibilité excessive des équipements.
Cette thèse propose une architecture générique de surveillance "active" pour l'aide à la maintenance de systèmes mobiles. Cette architecture repose sur des entités de surveillance "intelligentes" capables d'évaluer l'état de santé des équipements surveillés. Notre contribution se situe à la rencontre de différentes communautés de Recherche et s'appuie notamment sur des concepts développés par la communauté PHM (*Pronostics and Health Management*).
L'architecture proposée est mise en œuvre et appliquée à la surveillance d'un système réel de transport ferroviaire dans le cadre du projet SURFER (SURveillance active FERroviaire) conduit par Bombardier-Transport.

Mots-clés : surveillance, diagnostic, maintenance, systèmes embarqués, systèmes holoniques, systèmes de transport ferroviaires.

Abstract: over the last two decades, embedded systems have been introduced in several application areas (transportation, industry, housing, medical...). These systems have achieved more important tasks for delivering new services to users with ever shorter time-to-market deadlines at lower cost. The rapid integration of these systems within manufactured products is a competitive edge for companies. However, breakdowns related to these systems, along with the increasing level of equipment complexity, have made maintenance interventions much more difficult. Identifying root causes of some breakdowns currently represent a real challenge in the maintenance activities. They lead to an excessive downtime of equipment.
This thesis proposes a versatile "active" monitoring architecture for the maintenance assistance of mobile systems. This architecture relies on "smart" monitoring entities that can assess the health state of monitored equipment. Our contribution gathers different Research communities and relies particularly on concepts developed by the PHM (Pronostics and Health Management) community.
The proposed architecture is implemented and applied for monitoring a real railway transportation system within the SURFER project (SURveillance active FERroviaire) led by Bombardier-Transport.

Key words: monitoring, diagnosis, maintenance, embedded systems, holonic systems, railway transportation systems.