



HAL
open science

Proposition d'une architecture de surveillance holonique pour l'aide à la maintenance proactive d'une flotte de systèmes mobiles : application au domaine ferroviaire

Ahmat Fadil Adoum

► To cite this version:

Ahmat Fadil Adoum. Proposition d'une architecture de surveillance holonique pour l'aide à la maintenance proactive d'une flotte de systèmes mobiles : application au domaine ferroviaire. Systèmes embarqués. Université de Valenciennes et du Hainaut-Cambresis, 2019. Français. <NNT : 2019VALE0001>. <tel-02056495>

HAL Id: tel-02056495

<https://theses.hal.science/tel-02056495v1>

Submitted on 4 Mar 2019

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

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



HAL Authorization



Thèse de doctorat

Pour obtenir le grade de Docteur de l'UNIVERSITÉ POLYTECHNIQUE HAUTS-DE-FRANCE Spécialité Automatique et Génie Informatique

**Présentée et soutenue par AHMAT FADIL, ADOUM.
Le 14/01/2019, à Valenciennes**

Ecole doctorale :

Sciences Pour l'Ingénieur (ED SPI 072)

Equipe de recherche, Laboratoire :

Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaines (LAMIH – UMR 8201)

Proposition d'une architecture de surveillance Holonique pour l'aide à la maintenance proactive d'une flotte de systèmes mobiles : application au domaine ferroviaire

JURY

Président du jury

- M. Nouredine Zerhouni, Professeur, FEMTO-ST/Université de Franche-Comté

Rapporteurs

- M. Kondo H. Adjallah, Professeur, LCOMS/Université de Lorraine
- Mme Hind Bril-El Haouzi, Maître de conférences HDR, CRAN/Université de Lorraine

Examineurs

- M. Olivier Cardin, Maître de conférences HDR, LS2N/Université de Nantes
- M. Nouredine Zerhouni, Professeur, FEMTO-ST/Université de Franche-Comté

Directeur de thèse

- M. Damien Trentesaux. Professeur à l'Université Polytechnique Hauts de France

Co-encadrant :

- M. Abdelghani Bekrar. Maître de Conférences à l'Université Polytechnique Hauts de France

Membre invité

- M. Thierry Dailly, Bombardier Transport Franc

Dédicace.

A mes parents et à toute ma
famille,

Remerciements

Je tiens tout d'abord à remercier Monsieur Damien TRENTSAUX, pour la qualité de son encadrement, ses conseils pertinents et son aide tout au long de ces trois années de doctorat.

Je remercie mon encadrant de thèse, M. Abdelghani BEKRAR, et M. Thierry DAILLY de Bombardier 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.

Mes remerciements s'adressent ensuite aux membres de jury qui me font l'honneur de participer à ma soutenance. Je remercie sincèrement M. Kondo H. ADJALLAH et Madame Hind BRIL-EL HAOUZI pour avoir accepté d'évaluer mes travaux en qualité de rapporteurs. Mes remerciements s'adressent également à M. Olivier CARDIN et M. Noureddine ZERHOUNI pour leur participation au jury en tant qu'examineurs.

Je remercie également, et sans exception, l'ensemble du personnel du LAMIH, notamment du département d'Automatique et du laboratoire commun SurferLab, pour l'excellente ambiance qui y règne. Ces années de thèse ont été pour moi l'occasion de connaître des personnes exceptionnelles qui m'ont offert leur amitié et avec qui j'ai partagé d'agréables moments.

Je ne peux terminer sans avoir une pensée pour ma famille, qui m'a soutenu pendant toutes ces années d'études. Un grand merci à mes parents (AHMAT FADIL et TAHIRE ADOUM) pour leur soutien indéterminé.

Table des matières

Abréviations et Notations.....	8
Introduction générale	13
Ch. I Contexte de l'étude.....	15
Introduction	15
1. Enjeux actuels dans le domaine de la maintenance de flottes de systèmes complexes mobiles	16
1.1. Système complexe : notions	16
1.2. Cycle de vie d'un système	17
1.3. Activité de maintenance durant le cycle de vie d'un système	17
1.4. Maintenance d'une flotte de systèmes complexes mobiles	18
1.5. Contextes industriel et sociétal dans le domaine de la maintenance d'une flotte de systèmes mobiles	18
1.6. Missions de de la maintenance et aspects économiques, sociaux, environnementaux	19
1.7. Stratégies de maintenance	20
2. Surveillance : concepts, fonctions et architectures	24
2.1. Concept de surveillance	24
2.2. Fonctions de surveillance	24
2.2.1. Acquisition de données.....	25
2.2.2. Détection	26
2.2.3. Diagnostic	26
2.2.4. Pronostic.....	28
2.2.5. Élaboration de conseils	29
2.3. Architectures de surveillance	30
2.4. Systèmes sous surveillance considérés.....	32
2.5. Problématique globale de surveillance d'une flotte de systèmes complexes mobiles étudiée	33
2.6. Spécifications attendues pour la conception des architectures de surveillance.....	33
3. Conclusion	36
Ch. II Etat de l'art dans le domaine des architectures de surveillance d'une flotte de systèmes mobiles	37
Introduction.....	37
1. Architectures de surveillance centralisées d'une flotte	38
2. Architectures de surveillance Edge-centralisées d'une flotte	41
3. Architectures de surveillance décentralisées d'une flotte	42
4. Architectures de surveillance décentralisées et coopératives d'une flotte.....	45
5. Synthèse	51
6. Conclusion	52
Ch. III Proposition d'une architecture de surveillance	53
Introduction.....	53
1. Conception de l'architecture de surveillance EMH ²	54
2. Modélisation d'une forme générique d'évènements « SurfEvent »	57
3. L'architecture par Niveaux Holoniques de « EMH ² ».....	60
4. La méthode de particularisation	63

4.1.	Particularisation des niveaux Holoniques « Acquisition de données », « Manipulation de données », « Détection » et « Evaluation de Santé »	64
4.2.	Particularisation du Niveau Holonique « Evaluation de Pronostic »	68
4.3.	Particularisation du Niveau Holonique « Maintenance Dynamique »	71
4.3.1.	Holon « Justification de contexte »	75
4.3.2.	Holon « Politique de maintenance dynamique »	76
4.3.3.	Holon « gestion des opérations de maintenance et REX »	82
5.	Conclusion	84
Ch. IV Simulation, mise en œuvre et déploiement de l'architecture.....		85
Introduction.....		85
1.	Simulation du niveau Holonique « Evaluation de Pronostic »	86
1.1.	Objectif de la simulation	86
1.2.	Situation actuelle de référence	87
1.3.	Expérimentation de la situation actuelle.....	88
1.4.	Notre Expérimentation	90
1.5.	Analyse des résultats.....	91
1.6.	Impact et intérêt.....	92
2.	Processus de déploiement et aspects méthodologiques	93
2.1.	Déploiement progressif de l'EMH ²	93
2.2.	Sécurité de l'échanges de données de l'EMH ²	95
2.3.	Formation de l'équipe en charge des opérations de maintenance	96
3.	Une implémentation possible de l'architecture EMH ²	96
4.	Proposition d'un guide méthodologique pour accompagner le déploiement de EMH ²	99
4.1.	Évaluation de la situation actuelle pour la flotte ciblée de systèmes mobiles	99
4.2.	Définition des objectifs visés	100
4.3.	Définition de la structure desSurfEvents (voir la section 2 du chapitre 3)	100
4.4.	Sélection du mode de raisonnement des Holons (voir la section 4.1 du chapitre 3).....	100
4.5.	Implémentation des microservices dans les Systèmes mobiles et nœuds Edge-computing (voir la section 2.1 du chapitre 4).....	100
4.6.	Implémentation des microservices dans le CM (voir la section 3 du chapitre 4).....	101
4.7.	Formation de l'équipe en charge des opérations de maintenance (voir la section 2.3 du chapitre 4)	101
4.8.	Évaluation des résultats obtenus et comparaison avec les objectives visés	101
5.	Conclusion	102
Ch. V Application au domaine ferroviaire		103
Introduction.....		103
1.	Implémentation de la solution EMH ²	104
2.	Évaluation de la situation actuelle pour la flotte ciblée de systèmes mobiles.....	104
3.	Définition des objectifs visés.....	105
4.	Définition de la structure de la SurfEvent.....	105
5.	Sélection du mode de raisonnement des Holons	106
6.	Implémentation des microservices dans les Systèmes mobiles et nœuds Edge-computing	108
7.	Implémentation des microservices dans le Centre de maintenance	110
8.	Formation des équipes en charge des opérations de maintenance	111
9.	Évaluation des résultats obtenus et comparaison avec les objectifs visés.....	112
10.	Discussion globale sur notre étude de cas.....	114
Conclusion générale.....		115

Perspectives de Recherche	116
Références.....	119
Liste de figures.....	127
Liste de Tableaux	129
Annexes.....	130
Annexe A : Modèle Entité-Association(EA) d'un SurfEvent	130
Annexe B : Approches de modélisation des fonctions de surveillance	131
1. Fonctions de diagnostic et de pronostic	131
2. Fonction d'élaborations de conseils pour la maintenance.....	133
Annexe C : choix et définition d'une grammaire et d'un automate pour le système expert.....	135
Annexe D : choix d'une approche d'implémentation de l'architecture EMH ²	138

Abréviations et Notations

SURFER	surveillance active FERROviaire
BT	Bombardier Transport
CNRD	Centre national tchadien de recherche pour le développement
EMH	Embedded Monitoring Hierarchy
SHM	System Health Management
PHM	Pronostics and Health Management
PDC	Predictive Diagnostic Cloud
CPS	Cyber-Physical Systems
LCC	Life Cycle Cost
CBM	Condition Based Maintenance
AFNOR	Association Française de NORmalisation
$C_{TM}(t)$	Coût de maintenance total d'un système
$C_R(t)$	coût de remplacement des composants d'un système sur $[0, t]$
$C_{SC}(t)$	surcoût lié à la maintenance corrective lorsque le système est en panne sur $[0, t]$
$C_{SD}(t)$	surcoût lié au diagnostic lorsque le système est en panne sur $[0, t]$.
FDI	Fault Detection and Isolation
RUL	Remaining Useful Life
CM	Centre de Maintenance
KASEM	Knowledge and Advanced Services for Execution and Monitoring
IoT	Internet of Things
SNCF	Société Nationale des Chemins de Fer français
ARINC	Aeronautical Radio, Incorporated
BIT	built-in test
ACMS	Airplane Condition Management System
MAT	Maintenance Access Terminal
SDK	Smart Distance Keeping
DDS	Data Distribution Service
OSA-CBM	Open System Architecture for Condition-Based Maintenance
MDA	Model-Driven Architecture
VIPR	Vehicle Integrated Prognostic Reasoner
HM	Health Manager
IEDS	Intelligent Embedded Diagnosis System
DDIR	Détection de Défauts, Isolement et Reconfiguration
SurfEvent	forme générique d'évènements
AG	Advisory Generation
S_i	un système i
SS_{ij}	sous-système j d'un système i
$I(S_i)$	Evaluation de la fonction de surveillance d'un système i
$I(SS_{ij})$	Evaluation de la fonction de surveillance sous-système j d'un système i
$ =$	conséquence logique
$I(S_i, t, c_i)$	Evaluation de la fonction de surveillance d'un système i avec la prise en compte de ses contextes.
EA	Entité Association

UML	Unified Modeling Language
EIF	European Interoperability Framework
TIC	Technologies d'Information et de la Télécommunication
SurfConvert	convertisseur d'évènements SurfEvent
XML	Extensible Markup Language
JSON	JavaScript Object Notation
YML	Yet Another Markup Language
CSV	Comma Separated Values
TRDP	Train Real Time Data Protocol
CAN	Controller Area Network
MVB	Multifunction Vehicle Bus
REST	Representational State Transfer
MQTT	Message Queuing Telemetry Transport
OPC	Open Platform Communications
SurfProcessing	Fonction de surveillance pour les niveaux Holoniques Acquisition de données, Manipulation de données, Détection et Evaluation de Santé
$cycle(s)$	Cycle d'un système « s »
« e_1 et e_2 »	ensembles d'évènements représentant respectivement les conditions de début et de fin d'un cycle
$I(e_1, t)$	Evaluation de la fonction de surveillance d'un ensemble d'évènements « e »
$état(s, t)$	changement d'état d'un système
$(SurfEvent_t = SurfEvent_{t-1})$	changement d'état d'un SurfEvent à un instant donné t
$v(SurfEvent_t \neq SurfEvent_{t-1})$	
CART	Classification and Regression Tree
« H »	données historiques des SurfEvents d'un système
H	taille de H
$Entropie(H)$	mesure indiquant le niveau d'entropie (désordre) d'un ensemble de « H »
$Gain(H, surf)$	gain d'un SurfEvent « Surf »
Valeur(surf)	ensemble des valeurs possibles de l'attribut Surf
Hv	Sous ensemble de H, représentant les occurrences d'un SurfEvent
Hv	taille de Hv
REX	Retour d'Expériences
« Rég »	règle de prédiction
« Eq »	équipement
« Base_Eq »	base de connaissance
NH	Niveau Holonique
KPI	Key Performance Indicators
MFPO	Maintenance Free Operating Period
MRP	Maintenance Recovery Period
MFOPS	Maintenance Free Operating Period Survivability
$R_{syst}(t)$	fiabilité du système à l'instant t
t_{MFOP}	longueur de MFOP

RCA	Root cause analysis
T_{syst}	durée de bon fonctionnement d'un système
NC	niveau de confiance d'un système
J_w	critère de décision de maintenance avec w l'indice du critère de décision étudié
$MFOPS(X, t)$	$MFOPS$ d'un système à t après remplacement de l'ensemble X
m'	nombre d'ensembles d'opérations de maintenance dépendantes
$\delta_{\{U_v \subset X\}}$	variable binaire qui vaut 1 quand le v ème ensemble d'opérations de maintenance dépendantes est inclus dans l'ensemble X (0 sinon)
DS_v	durée économisée si la maintenance est effectuée simultanément sur les composants
C_i	coût du composant i
D_i	Durée de remplacement du composant i
τ_{MO}	Taux horaire de la main d'œuvre
M_k	ensemble comprenant les remplacements effectués pendant l'arrêt de maintenance k
$N_{AM}(t)$	nombre d'arrêts de maintenance sur l'intervalle de temps $[0, t]$,
$\delta_{i,k}$	variable binaire qui indique le remplacement du composant i pendant l'arrêt de maintenance k
C_{Log}	Coût logistique lié à l'entrée du système en atelier de maintenance.
$N_p(t)$	nombre de pannes du système sur l'intervalle de temps $[0, t]$
τ_I	coût de la perte d'exploitation par heure d'immobilisation
C_{Remor}	coût de remorquage d'un système
D_{Remor}	durée de remorquage d'un système
D_{Log}	durée logistique de prise en charge d'un système à son entrée en atelier
$C_{U,D}$	coût unitaire de diagnostic pour un composant
$D_{U,D}$	durée unitaire de diagnostic pour un composant
N_{SIF}	nombre de composants dans le système dont l'information de surveillance n'est pas disponible
DCA	Difference of Convex functions Algorithm
m	niveau de confiance d'une flotte / le nombre maximal de visite à ne pas dépasser sur la prochaine MFOP d'une flotte
Y	ensemble des systèmes mobiles à admettre vers les dépôts de maintenance
J	critère de l'impact sur l'exploitation de la flotte
$MFOPS_F(Y, t)$	$MFOPS$ de la flotte(F) à t après sélection de l'ensemble Y
$C_{TMF}(t)$	coût de maintenance total des systèmes mobile de la flotte
$C_{TMSi}(t)$	coût de maintenance total d'un système mobile sur $[0, t]$
δ_i	Variable binaire qui vaut 1 si le système mobile i est sélectionné vers les dépôts de maintenance, 0 sinon
5M	Main d'œuvre, Moyen, Méthode, Matière, Milieu
\bar{x}_n	Moyenne arithmétique
x_n	nombre de visites par jour

$y = a * x + b$	fonction affine décrivant l'influence d'un évènement " x" sur un autre "y"
x_i	occurrence i de l'évènement x
$moye(x)$	moyenne arithmétique de l'évènement x
« E »	ensemble des évènements
« t »	durée d'une expérimentation
« tm »	durée moyenne d'une expérimentation
EC	Edge-Computing
PKI	Public Key Infrastructure
MAD	Microservice acquisition de données
MM	Microservice manipulation
MD	Microservice détection
MES	Microservice évaluation de la santé
MPL	Microservice pronostic local
MPG	Microservice pronostic global
MMD	Microservice maintenance dynamique
DataM	Microservice données
MQTT	Message Queuing Telemetry Transport
TCN	Train Communication Network
ETB	Ethernet Train Backbone
WTB	Wire Train Bus
IP	Internet Protocol
MCG	Mobile Communication Gateways
TSDB	Time Series Database
ACP	Analyse en Composantes Principales
ANCOVA	Analyse de covariance
S.W.O.T	Strengths, Weaknesses, Opportunities, and Threats
SOAP	Simple Object Access Protocol
Exp	Expression d'un SurfEvent
Lexp	Expression logique d'un SurfEvent
Mexp	Expression mathématique d'un SurfEvent
Dexp	Expression temporaire d'un SurfEvent
Texp	Expression logique d'expression textuelle d'un SurfEvent
FL	Fonction logique d'Expression logique d'un SurfEvent
FT	Fonction textuelle d'Expression logique d'un SurfEvent

Introduction générale

La maintenance des flottes de systèmes mobiles dans le monde du transport et de la logistique revêt de nos jours une importance croissante en termes de sécurité, de fiabilité [1], de suivi, diagnostic et maintenance de ces systèmes. Dans ce contexte, l'intégration massive des fonctions automatisées dans les systèmes embarqués fournit potentiellement de nouveaux services afin de répondre à ces attentes. Malgré cela, dans un contexte industriel, l'ensemble du processus de surveillance, prérequis fondamental pour mettre en œuvre une maintenance efficace, reste encore largement sous performant, souvent réalisé manuellement et de manière peu proactive. Elle est en outre exécutée par des opérateurs qui disposent de très peu d'outils numériques pour les aider. La gestion d'une grande quantité d'informations, souvent répartie dans une région géographique étendue qui caractérise ce type de flotte couplée au besoin de réagir rapidement peuvent conduire les opérateurs à prendre des décisions dans l'urgence, potentiellement inefficaces ou même incorrectes, ce qui peut alors dégrader davantage la situation [2].

C'est pourquoi, au cours de ces dernières années, un nombre croissant de constructeurs et d'exploitants du secteur des transports ont orienté leurs efforts sur l'étude et le développement de systèmes de gestion d'événements pour automatiser et améliorer les processus de surveillance et ainsi soulager et éclairer les superviseurs de flotte dans leurs décisions. L'efficacité de ces systèmes de gestion d'événements est décrite non seulement en termes de réactivité, de flexibilité et de fiabilité, mais aussi en termes de capacité à s'adapter à un environnement de plus en plus concurrentiel, instable, coûteux et imprévisible [3].

Cette thèse s'inscrit dans le cadre d'un projet de recherche dont les partenaires sont Bombardier Transport, l'université polytechnique hauts de France et le Centre national de la recherche scientifique (CNRS). Il est également dirigé avec le soutien financier du Centre national tchadien de recherche pour le développement (CNRD).

Dans cette thèse, une architecture de surveillance à base d'agents intelligents pour l'aide à la maintenance d'une flotte de systèmes mobiles, appelée EMH², est proposée. Notre intuition a été que la standardisation des événements et la volonté de traiter les données et informations au niveau le plus proche des systèmes physiques qui génèrent ces événements peut devenir l'épine dorsale d'une stratégie de maintenance proactive d'une flotte efficace. Ce manuscrit présente une telle architecture de surveillance, appelée EMH². Cette architecture étend et généralise l'architecture EMH (Embedded Monitoring Hierarchy) proposée par un ancien doctorant du LAMIH [4]. Une application sur une flotte de 10 trains actuellement en service est présentée. Cependant, les concepts proposés dans cette thèse ont été développés pour être indépendant du domaine ferroviaire.

Ce manuscrit est organisé comme suit:

- Le premier chapitre présente le contexte de l'étude lié à nos travaux qui relève de la maintenance d'une flotte de systèmes complexes, mobiles et homogènes. Il détaille dans un premier temps, les principaux enjeux attachés aux contextes industriel et sociétal ainsi qu'aux aspects économiques, sociaux et environnementaux relatives aux fonctions de maintenance. Ce chapitre se focalise ensuite sur la fonction de surveillance. La problématique est spécifiée. Les types de système cible à maintenir considérés dans nos

travaux sont alors présentés. L'objectif affiché est de définir les éléments d'une architecture de surveillance pertinente et performante.

- Le deuxième chapitre propose une revue de la littérature existante dans le domaine de la conception d'architecture de surveillance d'une flotte des systèmes complexes mobiles. Cette revue considère un ensemble de contributions émanant des recherches dans les domaines du SHM (System Health Management), du PHM (Prognostics and Health Management) et du PDC (Predictive Diagnostic Cloud). Cette revue est structurée sur une typologie des architectures de surveillance proposées pour l'aide à la maintenance de flottes de systèmes mobiles. Cette typologie est composée de quatre catégories: « centralisée », « edge-centralisée », « décentralisée » et « décentralisée et coopérative ».
- Le troisième chapitre présente notre contribution d'une architecture de surveillance appelée « EMH² ». Notre contribution est articulée autour de trois propositions : la modélisation d'une forme générique d'évènements appelée « SurfEvent » ; La décomposition en Niveaux génériques de « EMH² » basée sur l'approche Holonique; et la particularisation des niveaux de l'architecture en fonction de leurs spécificités. L'originalité de notre approche réside dans le fait que chaque événement de la flotte est géré de la même manière. Cette approche est générique, hiérarchique et distribuée. Elle permet de gérer les évènements provenant des systèmes mobiles au travers d'un langage d'interfaçage commun pour l'échanges et la coopération entre les Holons des différents niveaux. Les Holons des différents niveaux sont intégrés dans EMH² au sein de différents niveaux Holoniques. Disposant d'une fonction principale de surveillance adaptée à leur niveau, ils sont sensibles à l'évolution en temps-réel des contextes de la flotte et concourent à un objectif de surveillance plus modulaire, réactif et adaptatif, tout en considérant l'optimisation du coût global de la maintenance dans un contexte proactif.
- Le quatrième chapitre présente tout d'abord une étude en simulation de notre contribution afin d'évaluer sa performance. Cette simulation concerne le niveau Holonique « Pronostic ». Elle vise à estimer la durée moyenne de génération de connaissances et d'intégration les retours d'expérience sur le processus de surveillance et de le comparer avec une situation industrielle réelle. Ce chapitre contient également des éléments de mise en œuvre possible, en fonction de la technologie actuelle de l'architecture EMH². Un processus de déploiement est ainsi proposé, traitant des aspects méthodologiques. Une implémentation possible, basée sur le paradigme de microservice, est finalement justifiée puis décrite.
- Le cinquième et dernier chapitre présente une application réelle de l'architecture EMH² dans le domaine ferroviaire sur une flotte de dix trains actuellement en exploitation. Cette application a été menée en collaboration avec notre partenaire industriel, Bombardier Transport. Le déploiement de l'implémentation de l'architecture de surveillance EMH² a été mené en suivant une méthode de déploiement en plusieurs étapes afin d'évaluer la performance des systèmes CVC (Chauffage, ventilation et climatisation), Accès voyageurs (Porte), Bogie, Traction et Pantographe d'un Train ;

Une conclusion générale résumant les principaux points de cette thèse et décrivant un ensemble de perspectives termine ce document.

Ch. I Contexte de l'étude

Introduction

Une gestion performante de la maintenance d'une flotte de systèmes est cruciale pour le **contrôle** et la **baisse des coûts** d'exploitation. Le **contrôle** complet permet de disposer d'un programme de suivi, de diagnostic et d'entretien régulier et préventif capable d'améliorer la sécurité des usagers et la sûreté de fonctionnement des systèmes et donc d'améliorer le service offert par l'entreprise et permettre la maîtrise et la réduction au mieux de l'impact sur l'environnement. La **baisse des coûts** des opérations de maintenance, de pannes, d'immobilisation des systèmes et de consommation de l'énergie, conduit, quant à elle, à l'augmentation de la rentabilité.

Ce premier chapitre décrit le contexte de notre étude et les enjeux sous-jacents à la maintenance d'une flotte de systèmes mobiles. Pour ce faire, nous présentons d'abord succinctement le concept de systèmes complexe et de cycle de vie d'un tel système, suivi de la description de l'activité de maintenance qui le concerne et son impact sur ses différentes phases de cycle de vie. Les différentes stratégies et le rôle de l'activité de surveillance sont alors spécifiquement étudiés. Nous nous focalisons en particulier sur les informations de surveillance et notamment comment elles peuvent être construites, traitées et enrichies de leur source à leur consommateur final (par exemple, un superviseur de flotte) afin de construire un outil d'aide à la décision efficace.

Les enjeux industriels, économiques, sociaux et environnementaux dans le domaine de la gestion d'une flotte de systèmes mobile sont alors décrits. L'objectif est de caractériser les éléments d'une architecture de surveillance qui serviront de support au positionnement de l'état de l'art, ainsi que ses indicateurs de performances par rapport à un ensemble de systèmes cibles applicatifs que nous considérerons dans la suite de nos travaux.

1. Enjeux actuels dans le domaine de la maintenance de flottes de systèmes complexes mobiles

1.1. Système complexe : notions

Il existe de très nombreuses définitions concernant un système et ce n'est pas l'objet de cette thèse d'en discuter finement les tenants et aboutissants. Selon l'ISO/IEC/IEEE 15288 [5, p. 2015], un système peut être considéré comme « une combinaison d'éléments en interaction organisés pour atteindre un ou plusieurs buts déclarés ». Selon Deming [6], « un système est un réseau de composants interdépendants qui œuvrent ensemble pour essayer d'atteindre le but du système ». Une autre définition est : « un système est constitués d'un ensemble d'équipements embarqués ou non, mobile ou non et répartis géographiquement ou non. Cet ensemble est désigné ci-après sous-systèmes, et organisé suivant une structure hiérarchisée de contrôle [7] ».

Ces définitions ont plus ou moins la même signification que nous retenons dans le cadre de cette thèse : un système est une vue conceptuelle d'un certain nombre d'éléments définis par une frontière les isolant de leur environnement et interagissant physiquement ou numériquement avec cet environnement. Ces éléments peuvent être embarqués ou non, potentiellement mobiles et répartis géographiquement qui interagissent pour atteindre un but [8]. L'interaction entre les éléments garantit que le système est quelque chose de plus que la somme des éléments individuels (principe de la « somme systémique »), et l'interaction entre les éléments signifie également que si un élément change d'une manière ou d'une autre, par exemple en raison d'une défaillance ou d'une modification, alors le système pourra changer également sa structure, sa fonction ou son comportement[9], [10].

Au sein du département automatique du LAMIH, on distingue deux types de complexité : une complexité « interne » due aux nombreuses non-linéarités et singularités qui caractérisent un ou plusieurs de ses éléments dans un contexte multi-physique, et une complexité « externe » due aux nombreuses interactions entre éléments qui eux peuvent être « simples » mais variés et de nature différentes et évoluant dans un environnement perturbé non déterministe. « Simple » signifie ici « dont le comportement reste suffisamment prédictible pour un ensemble de situations prévues ». Dans ce cas, la complexité provient du fait que ces interactions sont potentiellement difficiles à comprendre et à maîtriser. Nos travaux relèvent de ce second type de complexité. Ainsi, de notre point de vue, la complexité d'un système peut être mesurée en termes de nombre d'éléments et d'attributs des relations entre les éléments [11]. Cela signifie qu'un système complexe est une collection de nombreuses d'éléments de natures différentes disposant de propriétés et comportements spécifiques et interagissant entre eux ainsi qu'avec l'environnement. Plus la population d'éléments croît, plus il devient difficile pour l'observateur de prédire le comportement de l'ensemble du système, même en supposant que nous connaissons toutes les propriétés de base et les règles de comportement de chaque élément. À ce stade, nous n'avons aucune présomption sur le degré d'autonomie des entités [12]. Le concept de CPS (Cyber-Physical Systems) est exemple d'un système complexe dont la complexité est au moins « externe » dans la mesure où un CPS comprend des systèmes variés, intégrant une composante physique et informationnelle, intelligents plus ou moins autonomes et connectés en réseau, capables d'échanger des informations, de déclencher des actions et de se contrôler indépendamment[13]. Le concept de CPS peut être considéré comme une confluence des concepts de systèmes embarqués, de systèmes temps réel, de systèmes de capteurs distribués et de contrôle [14].

1.2. Cycle de vie d'un système

L'ISO/IEC/IEEE 15288[5, p. 2015], définit le cycle de vie d'un système comme un ensemble d'activités définies, adaptées et maintenues de manière cohérente afin de répondre à la nature des systèmes. La durée de ces activités est déterminée par les besoins techniques et commerciaux variables exprimés pour ces systèmes. Ces durées peuvent être prolongées ou diminuées selon le niveau d'attention exercé sur un système en terme de surveillance et de maintenance. Les différentes activités composant le cycle de vie d'un système sont usuellement les suivantes :

- L'activité de **Design** a pour objet d'évaluer de nouvelles opportunités (par exemple, commerciales) et de développer des exigences système préliminaires et des solutions de conception réalisable ;
- L'activité de **Développement** a pour objet de développer un système qui répond aux exigences (des acquéreurs, des usagers, etc.) et qui peut être produit, testé, évalué, exploité, maintenu et retiré ;
- L'activité de **Production** a pour objet de produire ou fabriquer les produits, ainsi que les tests, supports et processus de validation associés selon les besoins ;
- L'activité d'**Utilisation** a pour objet d'exploiter et utiliser les produits et services des systèmes dans le cadre d'une utilisation prévue dans des environnements prévus et assurer une efficacité opérationnelle continue.
- L'activité de **Support** a pour objet de fournir des services de logistique, de maintenance et d'amélioration qui permettent un fonctionnement continu du système et un service durable.
- L'activité de **Retrait** a pour objet de permettre le retrait ou le recyclage d'un système (ou d'un élément de système ou d'un sous-produit) et des services opérationnels et de soutien connexes.

1.3. Activité de maintenance durant le cycle de vie d'un système

Le cycle de vie d'un système peut être représenté de façon simplifiée par trois phases principales : une phase de conception ; une phase d'exploitation ; et une phase de fin de vie. L'activité de support, notamment de la maintenance joue un rôle très important durant toute les phases de cycle de vie d'un système[15]. Elle intervient durant la phase de conception afin de minimiser les futurs coûts de maintenance d'une organisation. En particulier, des questions relatives à la structure même du système doivent être posées au moment de sa conception, par exemple, « Comment agencer les composants du système, les uns par rapport aux autres, afin de faciliter les futures tâches de maintenance ? ». La maintenance, dont l'objet est de maintenir ou augmenter la disponibilité et la fiabilité d'un système en phase d'utilisation, intervient principalement durant la phase d'exploitation afin d'accroître la capacité de réponse (réactivité) d'une organisation, de son outil de production et favoriser simultanément un niveau de performance acceptable, suivant le triplet (coût ; délai ; qualité), et une réduction maximale des risques d'apparition d'incidents critiques. La définition que nous retenons est la suivante : la maintenance est l'activité qui consiste à maintenir ou à rétablir un système dans un état dans lequel il peut accomplir la fonction requise [16] . La disponibilité est définie par l'AFNOR (Association Française de Normalisation) comme l' aptitude d'un système à être en état d'accomplir une fonction requise dans des conditions données, à un instant donné, en supposant que la fourniture de moyens extérieurs nécessaires soit assurée [17]. La fiabilité est définie par AFNOR comme l'aptitude d'un système à accomplir une fonction requise ou à satisfaire les besoins des utilisateurs, dans des conditions données, pendant une durée donnée [17]. La maintenabilité consiste enfin à définir l'aptitude d'un système, à être maintenue ou rétablie, sur un

intervalle de temps donné, dans un état dans lequel il peut accomplir une fonction requise, lorsque la maintenance est accomplie dans des conditions données, avec des procédures et moyens prescrits [17]. Ces points seront repris dans la suite de ce document.

La maintenance intervient également durant la phase de fin de vie d'un système afin d'offrir des services supplémentaires aux organisations. Elle permet en effet de prolonger la vie des systèmes grâce à la réalisation d'interventions durant la phase d'exploitation (afin notamment de garantir un retour sur investissement), mais également de les faire évoluer tout en les conservant fonctionnels. De plus, la maintenance peut être mise à profit pour assurer que le système puisse être démantelé, et qu'une partie de ses composants puisse être réutilisée [15].

1.4. Maintenance d'une flotte de systèmes complexes mobiles

Dans le cadre de nos travaux, nous nous focalisons sur l'activité de **maintenance durant la phase d'exploitation d'une flotte de systèmes complexes mobiles**. Par maintenance de flotte, nous entendons la gestion d'un ensemble de systèmes mobiles dispersés sur une vaste zone géographique, qu'il convient d'améliorer directement ou indirectement en influant sur sa fiabilité, sa disponibilité ou sa maintenabilité [18]. Par système mobile, nous entendons un système intégrant un ensemble d'équipements mécatroniques, multiniveaux connectés en réseau capable de se mouvoir par rapport à un référentiel terrestre et en interaction avec une infrastructure qui elle est fixe. Il assure sa fonction en s'appuyant sur cette infrastructure. Cette infrastructure désigne un environnement regroupant un ensemble de services (tels que les services de communication, de géolocalisation, de signalisation, génie civil, etc.) requise au bon fonctionnement d'un système. La maintenance de cette infrastructure est non considérée dans le cadre de nos travaux.

Les résultats de nos travaux peuvent potentiellement être considérés comme un support pour les étapes d'amélioration en phase d'exploitation du système complexe (« overhaul »), de conception des versions ultérieures de ces systèmes (via un retour d'expérience), et de gestion de fin de vie des systèmes. Cependant, ces développements ne sont également pas étudiés dans cette thèse.

La maintenance d'une flotte de systèmes complexes permet aux organisations d'éliminer ou de minimiser les risques associés à l'investissement, d'améliorer l'efficacité, la productivité et de réduire les coûts de frais liés aux exploitations [19], [20]. Elle peut être gérée par un service en interne ou externe d'une organisation [21].

1.5. Contextes industriel et sociétal dans le domaine de la maintenance d'une flotte de systèmes mobiles

La mobilité croissante des personnes et des biens à travers le monde implique de nouveaux besoins en termes de maintenabilité, disponibilité et de fiabilité de flotte des systèmes complexes, par exemple des trains, des camions, des navires et des avions. Pour répondre à ces besoins, les constructeurs et les exploitants de ces systèmes doivent améliorer en continu la qualité de service et la disponibilité des systèmes complexes et coupler ceci à une réduction des coûts d'exploitation de maintenance [22]. De plus, dans un contexte de libéralisation progressive des marchés et d'entrée en jeu des nouveaux acteurs, favorisant ainsi la concurrence, la concurrence entre les constructeurs et entre les exploitants de ces systèmes de transport s'accroît : l'activité de maintenance est devenue un véritable enjeu stratégique et politique [23].

Dans le cadre de la maintenance d'une flotte de systèmes complexes mobiles, les attentes sont:

- **Pour les constructeurs de systèmes mobiles:** assurer une bonne fiabilité et disponibilité des systèmes mobiles durant la phase de garantie selon les exigences des opérateurs et créer les emplois et les services de demain.
- **Pour les exploitants et opérateurs de systèmes mobiles:** maintenir et si possible accroître la disponibilité et la fiabilité des systèmes hors période de garantie et optimiser l'équilibre entre opérations d'exploitation et de maintenance des systèmes mobiles; optimiser globalement les cycles de vie des systèmes d'un point de vue financier (Life Cycle Cost (LCC)), réduire le nombre de systèmes mobiles immobilisés et améliorer l'image et la compétitivité de l'entreprise vis-à-vis de ses clients.
- **Pour les usagers de systèmes mobiles et la société:** maintenir la satisfaction, la confiance, la sécurité et la réduction des temps d'attente dues à des perturbations ; limitation de l'impact environnemental et des risques sur la santé (accidents, pollution, etc.)

Dans ce contexte, une difficulté que rencontrent les industriels du secteur des transports provient du fait que les améliorations de la gestion de la flotte de systèmes complexes, et donc de leur maintenance, sont en fait entravées par l'inefficacité de la gestion des événements [4]. De plus, la complexité d'une flotte de systèmes (par exemple, un train intégrant un ensemble d'équipements mécatroniques, multiniveaux et en réseau), dispersés sur une vaste zone géographique, conduit à des difficultés à obtenir une vision globale, précise et exhaustive de l'état de la flotte. Ceci a un impact négatif sur la réactivité, le coût, et la qualité des services rendus par la flotte[18], [21].

Comme indiqué auparavant, le cadre de notre étude est celui de la maintenance d'une flotte de systèmes complexes mobiles. La partie suivante étudie plus en détail la fonction « maintenance » et sa particularisation à ce cadre. Cette étude nous permettra également de positionner nos travaux relatifs à une des sous-fonctions importantes de la maintenance, celle de la surveillance des systèmes.

1.6. Missions de de la maintenance et aspects économiques, sociaux, environnementaux

La norme CEN - EN 13306 définit La maintenance comme une combinaison d'activités techniques, administratives et de management destinées à maintenir ou à rétablir d'un système dans un état dans lequel il peut accomplir sa fonction durant son cycle de vie [24, p. 360]. Son objectif est donc de surveiller (inspections, contrôles, prédictions, etc.), remplacer, investiguer, réparer, etc., sur un équipements neuf, usé ou défectueux, que l'on souhaite maintenir ou rétablir [25]. La maintenance possède de nombreuses retombées sur la croissance économique et le fonctionnement d'une entreprise, bien au-delà d'assurer du bon fonctionnement des systèmes. Il représente également un facteur très important de protection de l'environnement, vis-à-vis de malveillances et de dégradations et vieillissements, agissant ainsi pour la sûreté des systèmes[26].

L'activité de maintenance d'un système, comme celle d'une flotte constituée d'un ensemble de systèmes complexes mobiles, s'intègre dans le cycle de vie du système (voir Figure 1. 1). Elle est opérée par le constructeur durant la phase de garantie, et par l'exploitant durant le reste du cycles vie de des systèmes (Par exemple un train peut avoir une durée de vie moyenne de 30 ans). Cette

activité de maintenance englobe différentes sous-activités qui 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)[4]. L'activité de maintenance, comme toute autre activité productive, générera autant d'activités pouvant impacter les aménagements des territoires, la consommation d'énergie, la gestion des déchets, les rejets dans l'atmosphère, etc. Ces activités ont ainsi également un impact sur les aspects économiques, technologiques, sociaux et environnementaux.

Selon l'hypothèse « coûts-bénéfices »[27] , appelé aussi « Arbitraire et contingence de l'analyse », ces aspects sont étroitement liés à des impératifs de sécurité (des hommes et des matériels) et dépendent de facteurs externes et de pratiques internes, tels que : l'entrée sur le marché des nouveaux acteurs, la libéralisation des activités de la maintenance, l'innovation et la recherche, l'évolution des normes technologiques et environnementales, le retour sur expérience, l'évolutions des besoins sociétaux, etc.

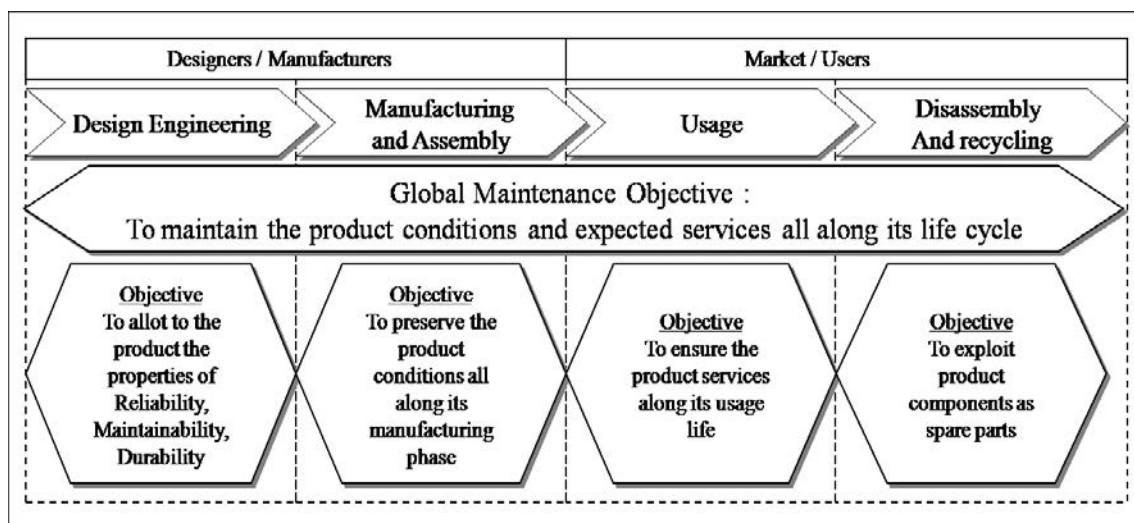


Figure 1. 1 : La maintenance dans le cycle de vie du système[28]

Toute stratégie de maintenance est mise en œuvre au travers de deux types fondamentaux de maintenance (maintenance préventive et maintenance corrective). La partie suivante décrit ces aspects.

1.7. Stratégies de maintenance

Fondamentalement, il existe deux types de maintenance [29] : la maintenance préventive qui vise à initier des opérations de maintenance sur un système avant que des défaillances ne se produisent, et la maintenance corrective qui intervient après la détection et la localisation d'un défaut c'est-à-dire sur un système déjà en état de panne [24, p. 360]. Ces deux types de maintenance servent de base pour construire différentes stratégies de maintenance plus ou moins élaborées (par exemple, maintenance prévisionnelle, conditionnelle, opportuniste, furtive, etc.). Toutes ces stratégies reposent sur trois éléments : l'événement déclencheur d'une opération de maintenance, le type de maintenance correspondante et la nature des opérations de maintenance réalisées sur le système[30]. Nous étudions ci-après ces deux types de maintenance et leur déclinaison :

- **La maintenance préventive** fait l'objet ces dernières années d'une demande croissante dans un grand nombre d'applications dans différents domaines, on peut citer les domaines de la logistique, du transport et de la production par exemple. Elle est effectuée soit sur un intervalle prédéterminé soit selon des critères prescrits. Elle a pour objectif de prévenir les pannes (défaillances) lors du fonctionnement du système[31]. La maintenance préventive se décline en trois sous-types normalisés: la maintenance préventive à caractère systématique (Predetermined Maintenance), conditionnelle (Condition Based Maintenance) et prévisionnelle (Predictive Maintenance) [24, p. 360]. Ces trois types sont présentés comme suit :
 - **La maintenance préventive à caractère systématique** consiste à planifier les interventions en fonction d'un calendrier défini à l'avance, indépendamment des événements qui surviennent. Ce calendrier est basé sur la durée moyenne de fonctionnement des sous-systèmes par exemple, ou sur un intervalle conformément à une échéance (recommandations constructeurs, par exemple). Il s'agit donc de la maintenance planifiée des biens, conçue pour améliorer leur durée de vie et éviter toute activité de maintenance imprévue [32].
 - **La maintenance conditionnelle** repose sur les conditions des mesures de paramètres, de déclenchement des alarmes et signalements, en fonction de l'évaluation de paramètres significatifs (seuils prédéterminés) de l'état d'un bien [33], par exemple, des niveaux de vibration, de bruit, de température, etc., et alerte sur la nécessité d'une opération de maintenance lorsque des éléments annonciateurs d'une défaillance sont observés. Bien que la maintenance conditionnelle soit plus dynamique que la maintenance systématique puisqu'elle considère l'état de santé réel d'un bien, le temps restant avant l'occurrence de la défaillance reste indéterminée avec ce type de maintenance [7].
La maintenance prévisionnelle repose sur des techniques analytiques élaborées afin de réduire à la fois le coût de la maintenance et les temps d'arrêt d'un bien grâce à l'identification précoce (prévision) des défaillances imminentes du bien [34]. Ce type de maintenance part du principe que la maintenance la plus efficace est celle qui est réalisée au plus juste avant la panne, ni trop en avance, ni trop tard. Elle peut être qualifiée de maintenance « **prédictive** » si elle repose sur la projection des évolutions de paramètres surveillés dans le futur,
- **La maintenance corrective** intervient après panne, mais seulement après la détection et la localisation d'un défaut. Elle a pour objectif de rétablir un bien en état fonctionnel au travers d'opérations de réparations ou de modifications pour rendre ou améliorer soit de manière provisoire ou pérenne un système en état de bon fonctionnement. Elle peut prendre soit la forme d'une maintenance palliative soit d'une maintenance curative[30]:
 - **La maintenance palliative** consiste à assurer la remise en état provisoire d'un bien sans traiter les causes de défaillance sous-jacente pour assurer notamment et dans l'urgence un état stable, non dangereux ou terminer un service difficilement préemptif. Une action de maintenance palliative doit être suivie systématiquement d'une action de maintenance curative [35].
 - **La maintenance curative** consiste à assurer la remise en état de manière durable et définitive d'un bien en tenant compte de ses causes de défaillance sous-jacente et profondes. Contrairement à la maintenance palliative, la maintenance curative a pour objectif de supprimer les causes des défaillances [7].

La Figure 1. 2 ci-dessous récapitule les principales stratégies de maintenance recensées dans la littérature selon ces deux catégories.

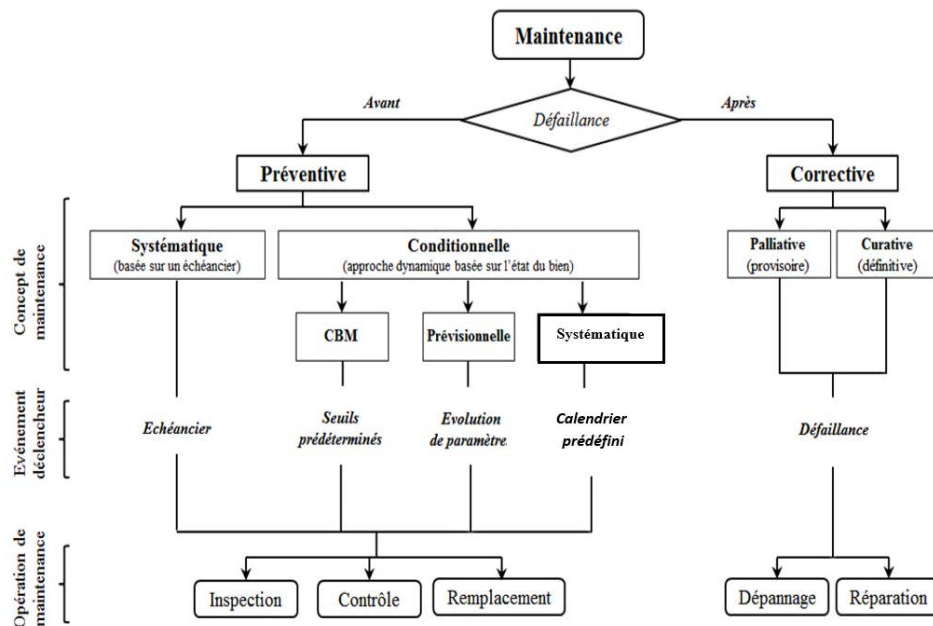


Figure 1. 2: Différents types de maintenance[7], [36].

Les stratégies de maintenance classées dans la Figure 1. 2 ont le point commun de se focaliser sur des objectifs de fiabilité, disponibilité ou de maintenabilité[37],[3]. Cependant, comme indiqué auparavant, les constructeurs ou les exploitants cherchent à rationaliser leurs propres coûts et à diminuer l'impact de la maintenance sur l'indisponibilité d'un système principal. Ces notions ont fait apparaître deux nouvelles stratégies de maintenance visant à atteindre ces objectifs : les maintenances **opportuniste** et **opportune**. Toutefois vis-à-vis de la performance globale, l'objectif de limiter l'impact économique en exploitation seule n'est pas suffisant. Ainsi une stratégie ayant pour objectif l'optimisation de critère représentatif du cycle de vie tel que le LCC doit être envisagée : dans ce cas est introduit le concept de stratégie **proactive**[37]. Ces stratégies récentes sont essentiellement dédiées à la maintenance « avancée » de systèmes complexe à composants multiples. Elles sont présentées comme suit :

- **La maintenance opportuniste** consiste à déclencher une opération de maintenance pour un composant non directement concerné par cette maintenance, en fonction de l'état (dégradé ou non dégradé) d'un ou de plusieurs autre(s) composant(s) surveillé(s) en continu [37]. Cette opération de maintenance est associée à une action initiale déclenchée par l'occurrence d'une défaillance (maintenance corrective)[38] ou par la réalisation d'une action de maintenance préventive non prévisionnelle (maintenance systématique ou conditionnelle)[39].
- **La maintenance opportune** permet de décider d'une opération de maintenance planifiée durant un arrêt de production d'un système ou sur un composant dégradé. Elle se démarque de la maintenance opportuniste en s'appuyant sur la maintenance prévisionnelle, et principalement sur sa capacité à évaluer le futur [37].

- **La maintenance proactive** a pour objectif d'évaluer les options de maintenance en intégrant les aspects coût globaux dans un contexte Life Cycle Cost (LCC) : coût de maintenance, coût d'approvisionnement mais aussi coût de perte sur les performances d'un système : perte de qualité, indisponibilité... et logistique. Il se base sur l'évolution future d'un système et s'appuie pour cela sur la maintenance prévisionnelle [37]. La maintenance proactive se base sur l'estimation de la durée de vie résiduelle d'un système (Remaining Useful Life) en tenant compte des conditions à venir d'exploitation et de maintenance du système [7].

Le Tableau 1. 1 positionne ces stratégies de maintenance opportuniste opportune et proactive [40].

	POURQUOI ?	QUAND ?	OÙ ?	QUOI ?
Maintenance opportuniste	Rationaliser localement les coûts d'indisponibilité et de maintenance	Une action de maintenance (corrective ou préventive non prévisionnelle) est programmée	Composant dégradé à proximité du composant visé par l'action de maintenance initiale	Action préventive décidée par l'expert sur la base du plan de maintenance
Maintenance opportune	Rationaliser globalement les coûts d'indisponibilité et de maintenance	Une opportunité future par rapport à la production ou une action de maintenance prévisionnelle	Composant dégradé à l'instant futur à proximité de l'action initiale ou dont la maintenance est réalisable pendant un arrêt de production	Action opportune de première ou deuxième espèce
Maintenance proactive	Rationaliser le LCC	Au moment le plus opportun vis-à-vis du LCC (doit tenir compte de la disponibilité du système, de la criticité de la mission, de sa qualité de réalisation...)	Composant dégradé le plus critique par rapport à l'évolution de la performance et du LCC (maintenance) ou action sur le système (pilotage)	Issue d'un processus d'aide à la décision portant sur des actions de : maintenance, pilotage du système ou modifications des missions

Tableau 1. 1: Prise de décisions pour les nouvelles stratégies de maintenance[37]

On remarque que les approches de maintenance sont complémentaires : leurs applications aux systèmes à maintenir, à fiabiliser ou à optimiser selon des critères représentatifs du cycle de vie du système à maintenir, dépendent notamment non seulement d'un équilibre économique et social, mais aussi de facteurs contextuels interne et externe amenant à un meilleur compromis (coût, objectif fixés et disponibilité, par exemple)). Quelle que soit l'approche de maintenance, et à un niveau très global, le coût de maintenance total d'un système $C_{TM}(t)$ (voir Formule 1. 1) peut être calculé sur un horizon de temps fini (la durée de contrat de maintenance, par exemple) et peut se décomposer de la façon suivante:

$$C_{TM}(t) = C_R(t) + C_{SC}(t) + C_{SD}(t)$$

Formule 1. 1: Coût de maintenance total d'un système[21]

Avec $C_R(t)$ le coût de remplacement des composants du système sur $[0, t]$, $C_{SC}(t)$ le surcoût lié à la maintenance corrective lorsque le système est en panne sur $[0, t]$ et enfin $C_{SD}(t)$ le surcoût lié au diagnostic lorsque le système est en panne sur $[0, t]$. Ces coûts intègrent tous les remplacements effectués de manière préventifs ou correctifs. Cette relation sera reprise et adaptée à notre cadre d'étude dans le chapitre III.

Au travers de notre étude, nous remarquons que la fonction de surveillance constitue une fonction essentielle des maintenances conditionnelle, prévisionnelle et proactive. A l'origine de toute action de maintenance non systématique, elle permet aussi de mettre à disposition des décideurs un ensemble d'informations leur permettant de prendre une décision efficace dans des délais raisonnables [41]. C'est pourquoi, dans la sous-section suivante, nous nous focalisons sur cette activité dans le cadre de la maintenance proactive. Dans cette sous-partie, nous allons nous intéresser à la façon dont ces informations de surveillance peuvent être construites, traitées et enrichies de leurs sources à leurs impacts, et leurs utilisations pour construire un outil d'aide à la décision efficace.

2. Surveillance : concepts, fonctions et architectures

En phase d'exploitation, la fonction de surveillance évalue dynamiquement les défaillances, dégradations et anomalies prématurées de certains sous-systèmes et estime leurs impacts sur la disponibilité, la fiabilité et l'exploitation sur les systèmes mobiles. De ce fait, La surveillance concerne les stratégies de maintenance conditionnelle, prévisionnelle ou proactive. Pour ces stratégies de maintenance, la surveillance occupe une place primordiale. Elle permet de mettre à disposition des décideurs un ensemble d'informations leur permettant de prendre une décision. Ces informations aident à localiser et identifier les paramètres révélateurs de la dégradation, et comprendre le passé, le présent et l'évolution future de du système considérée pour bâtir une décision de maintenance [21].

Dans cette partie, les principales fonctions de surveillance pour l'aide à la maintenance conditionnelle, prévisionnelle ou proactive sont présentées. Pour ces différentes stratégies de maintenance, la surveillance constitue la première d'une série de fonctions aboutissant à une opération de maintenance. De manière générique, une stratégie de maintenance non systématique est construite autour de l'articulation entre une fonction de surveillance, une fonction d'inspection, une fonction de contrôle, une fonction de dépannage et une fonction de réparation[16]. Dans la suite, nous nous focalisons sur le concept de surveillance et la fonction correspondante.

2.1. Concept de surveillance

La surveillance est définie comme l'ensemble des fonctions relative à la collecte, au traitement et au suivi des évènements qui surviennent dans des systèmes automatisés. Elle consiste donc à détecter les variations de paramètres des systèmes, de comprendre ces variations et interprétations observées, à distinguer celles qui résultent de défauts avérées de celles qui résultent de comportements normaux, et à décider si ces variations sont effectivement significatives par rapport à un comportement de référence (perception) [42]. La surveillance s'effectue généralement en-ligne, pendant le fonctionnement du système cible sous surveillance, mais certaines de ses fonctions peuvent être exécutées hors-ligne [43]. La partie suivante décrit les principales fonctions d'un système de surveillance dans le cadre d'une maintenance non systématique.

2.2. Fonctions de surveillance

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 (Prognostics and Health Management)[44], [45]. Dans ce cadre, un système de surveillance comprend généralement les fonctions d'acquisition de données, de détection d'anomalies de

fonctionnement, de diagnostic, de pronostic et d'élaboration de recommandations de maintenance[46], [47].

La *Figure 1. 3* schématise les et articule les différentes fonctions de surveillance d'un système. On y retrouve les trois grandes fonctions de détection, de diagnostic, de pronostic nécessaires à une bonne surveillance articulée selon la présence ou non d'une défaillance au sein du système sous surveillance. On constate généralement deux activités de surveillance:

- **La Surveillance dynamique** : la surveillance dynamique regroupe les fonctions de détection prédictive de dégradations, de diagnostic prédictif et de pronostic. Elle vise à prévenir la défaillance du système sous surveillance en projetant l'état de santé actuel du système sous surveillance dans le temps. Sa robuste est vital pour ne pas délivrer des alertes intempestives aux opérateurs de maintenance[48].
- **La Surveillance classique** : elle regroupe les fonctions de détection des défaillances survenues dans le système sous surveillance et de diagnostic. Elle intervient après l'occurrence d'une défaillance afin de rechercher les causes et de localiser les éléments défaillants du système[43], [49].

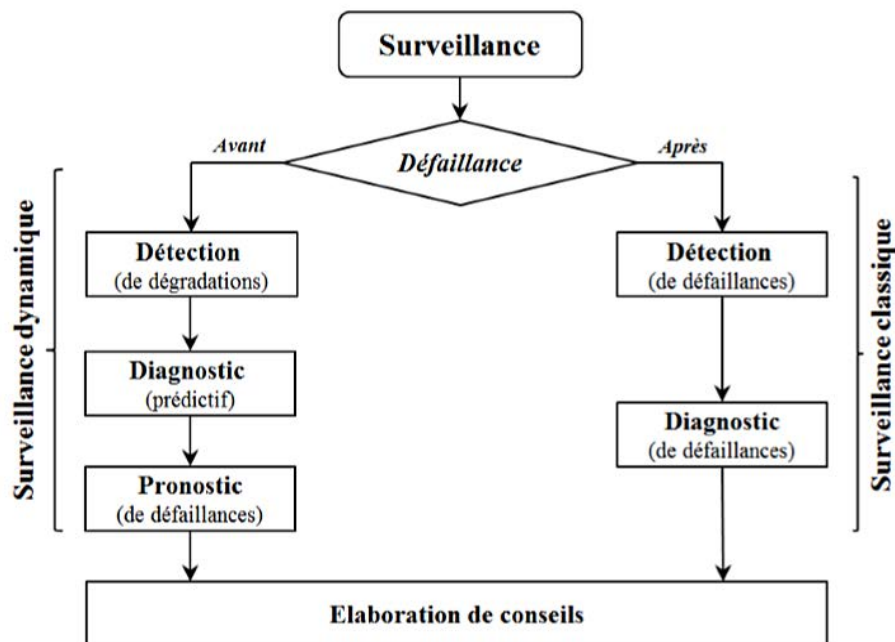


Figure 1. 3: Fonctions de surveillance dynamique et classique, adaptée de[7], [50].

Les différentes fonctions de surveillance illustrées sur la *Figure 1. 3* sont détaillées dans les sections suivantes.

2.2.1. Acquisition de données

C'est la première fonction d'un système de surveillance qui consiste à recueillir des données brutes sur le système sous surveillance, notamment des signaux qui mesurent les conditions physiques réelles et de convertir ces signaux en valeurs pouvant être manipulées par la fonction de détection. Cette fonction d'acquisition de données est généralement réalisée à l'aide de méthodes dites passives, non intrusives, n'altérant pas le fonctionnement du système sous surveillance[51].

2.2.2. Détection

Cette fonction de la surveillance est associée aux états des systèmes et les transitions entre ces états (par exemple, la transition de l'état de « en bonne santé » en l'état « en panne »). 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[7]. Les dégradations et les défaillances sont inévitables dans tout type de système industriel. Le risque de la panne s'accroît également avec l'augmentation de la complexité du système[11]. Bien que la détection soit une étape importante, ce n'est qu'un premier pas. Lorsqu'un défaut est détecté, l'estimation de la dynamique du défaut, la détermination du type de défaut et sa localisation deviennent importantes, car elles peuvent faciliter l'analyse de la cause première des défauts pour la maintenance et la réparation du système. De plus, il est souvent utile de déterminer la durée de vie restante du système afin d'améliorer la disponibilité du système et d'éviter les dommages aux composants ou la défaillance complète du système.

2.2.3. Diagnostic

Le diagnostic est une fonction de la surveillance qui suit logiquement la phase de détection. Il peut être exécuté en ligne ou a posteriori sur la base des symptômes détectés sur le système. Les symptômes traduisent les effets observables des défaillances. Le Diagnostic est défini comme le processus permettant la localisation, le traitement et la compréhension (identification) des dépendances entre événements, ainsi que leurs causes et conséquences sur la disponibilité du système[52]. Le traitement correct de ces événements constitue donc un outil clé pour rendre la fonction de surveillance efficace, réactive et adaptable. Son but est la détermination du type du mode de défaillance, de la localisation et de l'instant d'occurrence d'un défaut, en se basant sur une connaissance du système sous surveillance. Cette connaissance intègre, par exemple, une référence du comportement normal ou du comportement anormal du système sous surveillance[30]. Il est l'un des facteurs contribuant à la disponibilité des systèmes qui est une composante de la sûreté de fonctionnement[53]. La littérature sur les méthodes de diagnostic industriel est vaste, allant des méthodes analytiques à l'intelligence artificielle et aux méthodes statistiques, du point de vue de la modélisation, on recense des méthodes de diagnostic basées sur des modèles, basées sur des règles (base de connaissances) et basée sur les historiques des événements sont identifiées[54] (voir la *Figure 1. 5*). Les principales méthodes de diagnostic développées par les différentes communautés de recherche scientifique, sont:

- **Méthodes de diagnostic basées sur les modèles** : Dans ces méthodes, le système de supervision compare la grandeur déduite d'un modèle représentatif de structure et comportement (nominal ou fautif) du système sous surveillance observé avec le comportement prédit à l'aide de son modèle[55]. L'occurrence de faute est détectée à partir d'un raisonnement sur les incohérences entre les deux comportements. Un modèle incluant les comportements fautifs du systèmes (modèle de faute ou de dysfonctionnement) permet de localiser et d'identifier les fautes survenues[56]. Le modèle peut être de type quantitatif, qualitatif ou hybrides. On distingue généralement deux approches du diagnostic à base de modèles : l'approche FDI (Fault Detection and Isolation)[57] issue de la communauté automatique qui utilise des modèles quantitatifs (modèles continus) pour représenter le

comportement du système et l'approche DX (Diagnosis)[49] issue de la communauté informatique et intelligence artificielle qui utilise des modèles qualitatifs (modèles logiques ou discrets) pour décrire le système. Les méthodes de diagnostic basées sur des modèles regroupent ainsi les fonctions de détection et de localisation de défaillances (Figure 1. 4) [7].

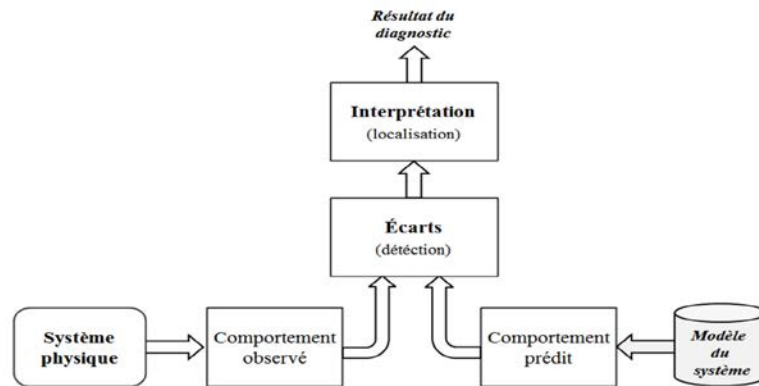


Figure 1. 4: Principe des méthodes de diagnostic à base de modèles[7]

- Méthodes de diagnostic basées sur les connaissances** : Dans ces méthodes, le système de supervision utilise les connaissances explicites des relations causales entre les symptômes, les défaillances et les fautes, pour garantir les fonctions de localisation et d'identification des systèmes sous surveillance. Ces relations, qui se doivent être les plus exhaustives possibles, sont souvent déterminées durant la phase de conception et proviennent d'une analyse fonctionnelle ou structurelle du système[56]. On recense dans cette catégorie de méthodes de diagnostic des méthodes tels que l'arbre de défaillance et le système expert et l'analyse des modes de défaillance, de leurs effets et de leur criticité (AMDEC)[58], [59].
- Méthodes de diagnostic basées sur des données historiques** : Dans ces méthodes, le système de surveillance exploite la disponibilité de grandes quantités de données historiques du système sous surveillance afin de déterminer des changements de caractéristiques par rapport à des valeurs de référence, ou par rapport à des valeurs mesurées à des instants passés dans une autre fenêtre temporelle[60]. L'objectif de ces méthodes est d'associer des ensembles de mesures à des états de fonctionnement connus du système. Ces approches utilisent des techniques de reconnaissance de formes, de classification, de partitionnement de données, et d'apprentissage de manière générale, pour traiter les données disponibles et construire un modèle du système fondé sur l'expérience. Un tel modèle ne provient donc pas d'une spécification physique ou structurelle du système[56]. Parmi ces méthodes, on compte les méthodes de représentation et d'analyse telles que l'analyse en composantes principales [61], la régression multiple[62], la classification et la reconnaissance des formes[63], Réseaux neuronaux (neural networks)[64] et la logique floue[65].

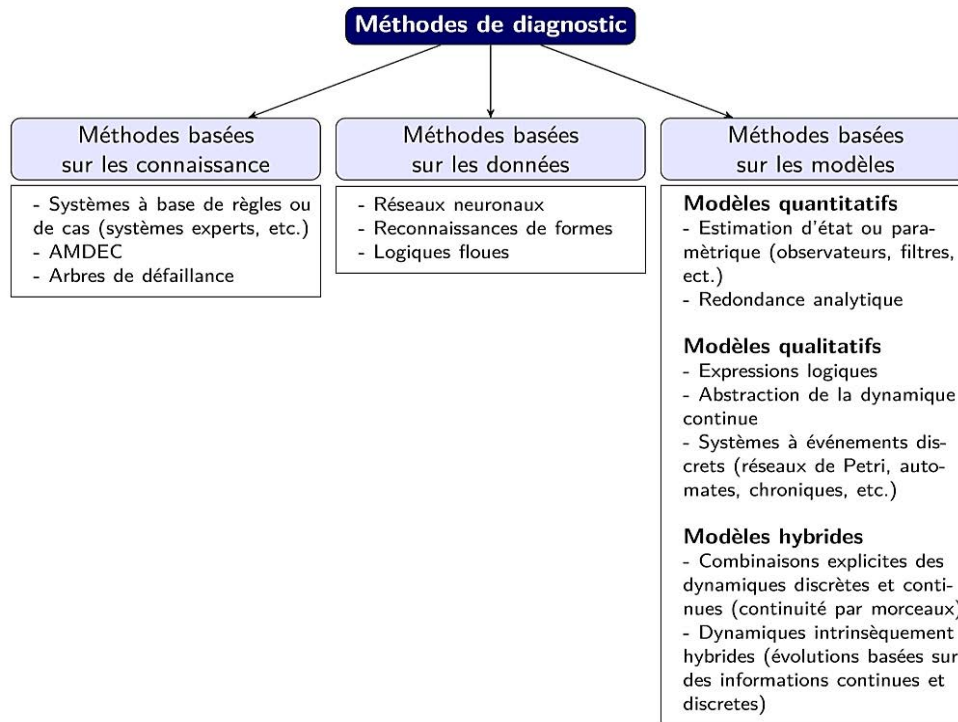


Figure 1. 5: Classification des méthodes de diagnostic[56]

2.2.4. Pronostic

De nombreux travaux menés dans le cadre des activités de recherche dans le domaine du PHM se concentrent sur la conception de modèles dits « robustes » et précis pour évaluer l'état de santé des systèmes pour des applications particulières afin de soutenir la prise de décision[66].

La fonction de pronostic a pour mission de choisir le meilleur moment pour la maintenance tout en réduisant les exigences d'inspection, le nombre d'occurrence de pannes du système et celui des défaillances secondaires. Elle est donc associée à l'évolution de la défaillance ou à son comportement futur et notamment, à l'estimation de sa durée de vie résiduelle utile RUL (Remaining Useful Life)[67] qui fait actuellement l'objet d'un nombre très élevé d'études et de recherches[68]. Le pronostic est spécifique à la maintenance prédictive et essentielle dans une approche CBM, car il permet d'anticiper au mieux les interventions de maintenance suivant la prédiction de l'état futur d'un équipement et la détermination de l'impact d'une défaillance sur les performances d'un système [69]. Comme dans le diagnostic, les méthodes de pronostic peuvent être divisées en trois types :

- **Méthodes de pronostic basées sur les données:** En ce qui concerne les méthodes axées sur les données pour le pronostic, un processus de collecte de données plus intensif est nécessaire afin de caractériser l'accumulation et la progression des défaillances. Le principal défi consiste à analyser un flux de données multidimensionnel provenant de nombreuses sources (conditions d'utilisation, contextes, conditions environnementales, etc.) dans un ensemble des systèmes homogènes. Il est important de dire que la gestion de l'incertitude est fondamentale dans l'estimation du RUL. Lorsque le composant est nouveau et que les

défaillances cumulées sont négligeables, les marges d'incertitude sur la durée exacte de la défaillance sont très importantes. Ces marges deviennent bien entendu plus étroites à mesure que le composant vieillit[70].

- **Méthodes de pronostic basées sur les modèles :** Les méthodes basées sur des modèles sont utilisées généralement pour obtenir des résultats plus précis. Leur conception nécessite une connaissance approfondie du système. Premièrement, il est nécessaire d'identifier et de valider expérimentalement les événements préjudicieux, ce qui n'est pas toujours une tâche facile et généralisable, car cela implique généralement de très longues études et larges expériences dans des conditions contrôlées qui ne reflètent pas nécessairement le vieillissement réel. Deuxièmement, dès lors qu'un événement perturbant est identifié, il reste la difficulté d'extraire de manière fiable un ensemble de caractéristiques ou d'estimer des paramètres influents à partir de données expérimentales étroitement corrélées avec ces événements. Troisièmement, l'évolution des défaillances est invariablement un phénomène difficile à tracer, ce qui rend leurs modélisations plus difficile et ce d'autant plus qu'elles dépendent également et fortement des conditions initiales (par exemple la distribution des défauts structurels ou matériels)[71].
- **Méthodes de pronostic basées sur les connaissances :** Les méthodes basées sur des connaissances reposent sur l'expérience et constituent une alternative à une modélisation délicate de la nature physique du système complexe sous surveillance. Ces techniques dérivent des techniques de type « systèmes experts », ou sont basées sur un raisonnement à base de cas pour prédire la probabilité de défaillance d'un système tout au long de sa vie et donc déterminer sa durée de vie [56].

2.2.5. Élaboration de conseils

L'objectif final d'une supervision, qu'elle soit classique ou dynamique, est d'aider à résoudre les problèmes provoqués par les défauts et pannes. Les fonctions d'acquisition de données, de détection, diagnostic et pronostic sont transparents pour les acteurs du système sous surveillance. A travers de la fonction d'élaboration de conseils, le système de surveillance, conseille, et partage ses recommandations sur l'état de santé actuel et prévisionnel des systèmes, ou privilégie un mode de fonctionnement ou une configuration à adopter pour que les systèmes puissent achever sa mission en évitant la panne[72]. La fonction d'élaboration de conseils et assiste principalement l'opérateur de maintenance dans la préconisation sur 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)[7]. 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. 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 cadre plus global du cycle de vie des systèmes et produits[7].

Par la suite, la **surveillance dynamique** des systèmes est retenue dans nos travaux car elle constitue un levier majeur pour prévenir la défaillance d'un système sous surveillance et en projetant l'état de santé actuel du système sous surveillance dans le temps. Les contributions développées pour ce type

de surveillance par différentes communautés de recherche scientifique seront détaillées dans le chapitre III. Le choix d'une méthode de diagnostic, pronostic et d'élaboration de conseils pour la surveillance d'une flotte des systèmes mobiles sera alors discuté plus tard dans ce document, lors de la particularisation des niveaux de surveillance.

Dans la sous-section suivant, nous présentons les différents types d'architectures de surveillance et les interactions de ses activités pour une meilleure prise de décisions de systèmes sous surveillance.

2.3. Architectures de surveillance

Dans le cadre de ce manuscrit, une architecture de surveillance est définie comme un système informatisé intégrant les fonctions de surveillance, et les algorithmes (logiciels) à supporter, ainsi que la localisation des différents éléments de calcul (hardware) en charge du flux d'événements, d'informations, et connaissances numérisées d'une flotte des systèmes [18]. Un événement est ici défini comme un changement d'état ou de variable horodaté dans un système (détecté ou non). Les événements sont des éléments clés de la surveillance, car les flux d'informations et de connaissances sont déclenchés, traités, stockés et communiqués à la suite de leur apparition.

Il est important de noter que notre contribution majeure se situe clairement au niveau de la proposition d'une telle architecture de surveillance qu'il convient dès lors de positionner clairement. Pour nous aider dans cette tâche, nous avons identifié deux typologies différentes.

La première typologie d'architectures de surveillance est proposée par [56] qui identifie deux types d'architecture de surveillance selon le niveau d'autonomie de ces architectures. L'autonomie traduit la réalisation d'une mission d'un système, sans intervention humaine et la faculté d'adaptation face aux perturbations de son environnement[56] :

- **Architectures orientées-mission** : Dans ce type d'architecture de surveillance, les fonctions de pronostic et de prédiction sont généralement sous le contrôle d'un opérateur humain. Ce type d'architecture se décompose hiérarchiquement en trois niveau (voir la *Figure 1. 6*) : le niveau décisionnel, le niveau exécutif et le niveau fonctionnel. Le niveau décisionnel est le niveau le plus élevé, il génère le plan des actions à réaliser, et nécessite pour cela une connaissance globale de l'objectif et du contexte de la mission ; Le niveau exécutif est le niveau intermédiaire entre le niveau décisionnel et le niveau fonctionnel, il a pour rôle d'exécuter le plan de mission en utilisant les différentes fonctions disponibles sur le système sous surveillance ; Enfin, le niveau fonctionnel joue le rôle d'interface avec les systèmes sous surveillance.

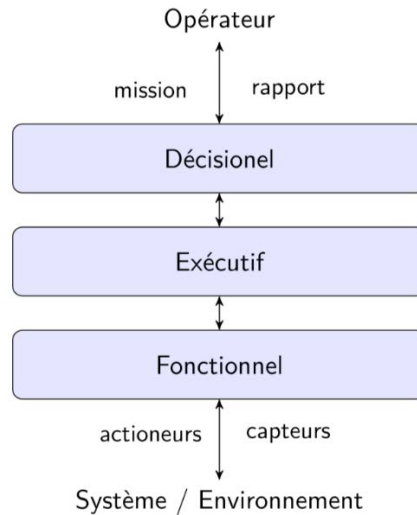


Figure 1. 6: Une architecture hiérarchique générique à trois niveaux : fonctionnel, exécutif et décisionnel[56].

- Architectures orientées-santé** : avec ce type d'architecture, la surveillance se démarque par la présence de fonctions de diagnostic et/ou de pronostic utilisant des connaissances globales sur le système sous surveillance et sur son contexte. Ces fonctions partagent des informations de surveillance avec les fonctions de décision, de contrôle et les acteurs de la maintenance. La Figure 1. 7 présente une forme générique de ce type d'architecture, dans laquelle les différentes fonctions sont toutes connectées deux à deux dans le cadre d'une aide à la décision pour la maintenance, la planification ou pour la mise à jour des objectifs de la stratégie de maintenance.

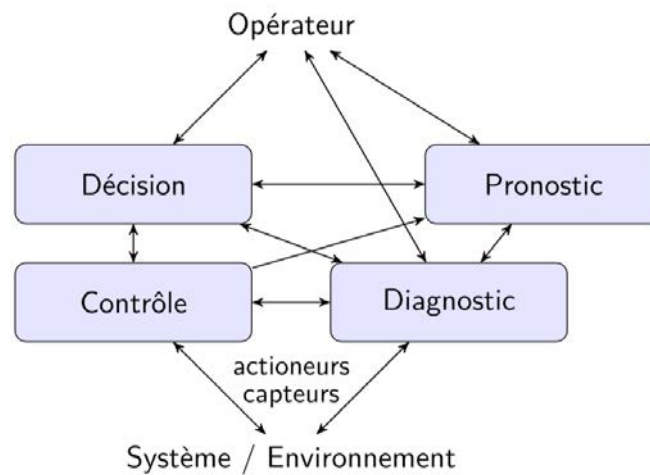


Figure 1. 7: Une architecture orientée-santé générique[56]

La seconde typologie est proposée par [7] . Elle consiste à différencier les architectures de surveillance selon le degré de centralisation des traitements, en partant des données brutes jusqu'à la construction des connaissances, de totalement centralisé à totalement distribué en passant par des degrés intermédiaires, hybrides. Nous considérons que cette typologie est fondamentale et, sous réserve d'une adaptation de notre part, elle servira avantagement le travail de

positionnement de la littérature présenté dans le chapitre suivant et nous aidera ainsi à identifier les éléments novateurs de notre contribution. C'est pourquoi nous ne détaillons pas plus à ce niveau cette typologie.

Maintenant que le cadre conceptuel de notre étude est posé, dans les sections suivantes, nous allons définir plus précisément le système sous surveillance considéré dans le cadre de nos travaux, puis identifier la problématique globale à laquelle nous nous intéressons. Nous en déduisons alors un ensemble de spécifications à respecter pour la conception de notre architecture de surveillance.

2.4. Systèmes sous surveillance considérés

Comme succinctement introduit, dans le cadre de ce manuscrit, les systèmes qui sont sous surveillance (« cibles ») sont supposés complexes, mobiles et immergés dans une flotte en interaction avec une infrastructure.

Ils sont, en addition et plus précisément, supposés être de nature homogène dans le sens où tous ces systèmes présentent la même mission (par exemple, transporter). Cette mission est en outre supposée être traitée par une même classe de solution technologique ou relevant du même secteur industriel (par exemple, le ferroviaire, l'aérien ou le routier). Cette homogénéité se traduit également par le fait que chaque système est supposé être structuré de manière similaire hiérarchiquement, du niveau supérieur (par exemple, un train) au niveau bas (par exemple, composant, une partie mécatronique), y compris des niveaux intermédiaires (par exemple, équipement: porte, clim ...). Il est enfin supposé que ces systèmes mobiles soient maintenus par un Centre de Maintenance au sol (CM) supervisant et connecté à la flotte (par exemple, en utilisant des systèmes de communication basés sur Internet, la téléphonie mobile ou des réseaux propriétaires). Le CM est responsable de la planification des interventions et de l'optimisation du coût global de maintenance. La *Figure 1. 8* schématise cette organisation.

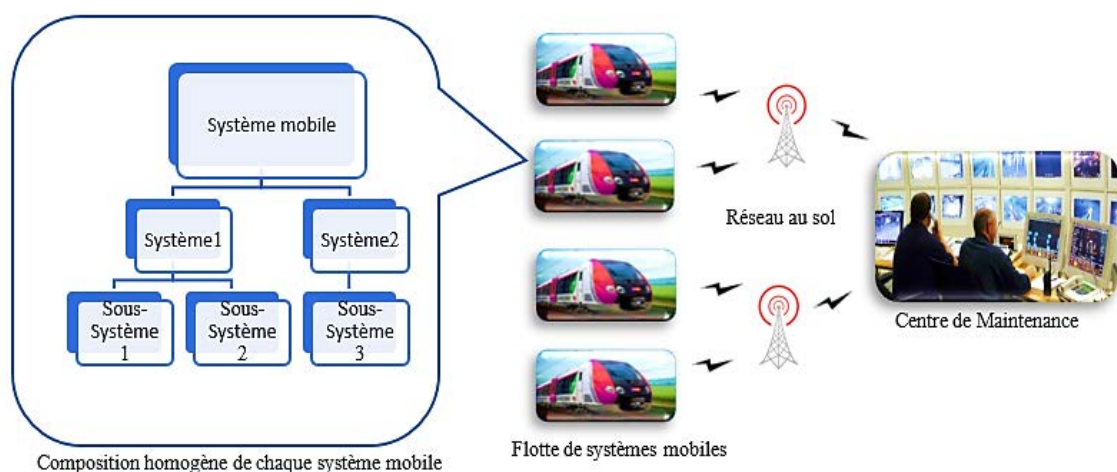


Figure 1. 8: Une flotte de systèmes complexes mobiles homogènes maintenue par un CM.

Dans la sous-section suivant, nous allons décrire parmi l'ensemble des problématiques de surveillance auxquelles doivent faire face aujourd'hui les industriels et les opérateurs de flottes de ces systèmes, celle à laquelle nous nous attachons. Nous traduisons alors ce problème en spécifications attendues concernant l'architecture de surveillance à concevoir dans l'objectif d'améliorer en continu la fiabilité, la maintenabilité, la disponibilité et la sécurité de la flotte de ces systèmes tout en assurant une optimisation maximale des coûts de maintenance.

2.5. Problématique globale de surveillance d'une flotte de systèmes complexes mobiles étudiée

Aujourd'hui, dans le contexte de la surveillance d'une flotte de systèmes complexes mobiles et homogènes, l'un des principaux problèmes rencontrés par les industriels et les opérateurs est de savoir *comment gérer et améliorer la surveillance de cette flotte où les processus d'information, de raisonnement, de décision et d'optimisation reposent sur des sources multiples d'événements et données dans un environnement distribué prenant en compte les spécificités, les emplacements et les différents usages de ces nombreux systèmes mobiles*. Cette question concerne des aspects importants tels que la gestion et l'optimisation des coûts et de la fiabilité de la maintenance, la réactivité du système de surveillance et sa capacité à apprendre et à s'adapter à des équipements neufs ou révisés.

Les mainteneurs de flotte doivent souvent faire face à d'énormes quantités de données brutes, d'informations et d'événements de surveillance liés au contexte de leurs systèmes [73]. De plus, ces événements, données et informations manquent souvent de précision et d'éléments contextuels. Ils sont même parfois contradictoires ou obsolètes. Enfin, leur degré d'urgence est rarement pris en compte [74]. Le mainteneur est fréquemment informé trop tard qu'une panne critique s'est produite, alors qu'elle aurait pu être évitée avec une gestion plus précise, pointilleuse et réactive des événements précurseurs générés par l'équipement critique.

Répondre à une telle question est donc complexe et concerne plusieurs dimensions: technologique, organisationnelle, psychologique et financière[75]. Comme présenté, le contenu de cette thèse couvre principalement la dimension *technologique* des architectures de surveillance dans le but d'améliorer le suivi et le diagnostic d'une flotte de ces systèmes.

2.6. Spécifications attendues pour la conception des architectures de surveillance

L'enjeu majeur lors de la conception d'une architecture de surveillance d'une flotte de systèmes est qu'elle doit être capable de répondre aux exigences des industriels et acteurs de la maintenance en terme de gestion et d'optimisation des coûts de maintenance, de la fiabilité des tâches de maintenance, de l'environnement, de la réactivité du système de surveillance, et de sa capacité à apprendre et à s'adapter à des équipements neufs ou révisés. Sa performance de gestion d'événements d'une flotte peut être décrite à travers différents indicateurs liés:

- **Au coût** (par exemple, le nombre de visites de maintenance par semaine)

- **Au temps** (par exemple, le temps nécessaire pour rechercher, diagnostiquer et classer un rapport relatif à une intervention ou le temps nécessaire à l'architecture de gestion des événements pour mettre à jour les données sur l'état de santé des équipements et du système mobile)
- **A la qualité** (par exemple, la qualité du diagnostic au niveau de la flotte, en évitant les fausses alarmes et les pannes d'équipements critiques)
- **A l'adaptabilité et l'apprentissage** (par exemple, le temps nécessaire pour caractériser, comprendre ou organiser le processus de surveillance d'un équipement ou d'un système nouveau ou révisé).

Nous avons traduit la problématique précédemment décrite en neuf spécifications qui traduisent les attentes industrielles quant aux architectures de surveillance:

1. **Big Data:** Le traitement correct des données massives est d'une importance majeure dans le secteur des transports. Par exemple, un seul train a plus de 100 000 événements à surveiller en temps réel, et les flottes de trains sont généralement composées de plus de 100 trains. Outre les problèmes connus et pertinents, en particulier le mode d'acquisition et la gestion du volume de données[76], une architecture de surveillance efficace nécessite des fonctionnalités supplémentaires en termes d'optimisation temporelle et de processus de surveillance récursifs en raison de la nature même des systèmes mobiles complexes à composants multiples. Les données et les événements doivent donc être analysés à l'aide de modèles (processus) similaires, quel que soit le niveau du système mobile considéré[4]. A l'inverse du big-data, nous estimons qu'il est possible d'être confronté à une problématique d'occurrence d'une donnée ou d'un événement extrêmement rare, parce que conséquence d'une chaîne d'événement (provoquée par une situation, un contexte, etc.) qui n'a jamais envisagée par les concepteurs. Ce type de problématique n'est pas considéré dans notre étude.
2. **Déploiement:** Déployer une nouvelle architecture de surveillance dans une flotte de systèmes mobile nécessite l'intégration de nouveaux systèmes embarqués spécifiques de surveillance matérielle et/ou logiciels, avec de nouveaux dispositifs de communication et des licences d'infrastructure. Ceci a un impact sur la disponibilité de systèmes mobiles actuellement ou bientôt en exploitation. Ce déploiement doit être mené avec soin afin d'assurer la prise en main et la maîtrise de l'architecture de surveillance;
3. **Gestion des coûts de maintenance:** une architecture de surveillance d'une flotte de systèmes complexes mobiles homogènes doit assurer une gestion précise des événements afin d'éviter des opérations de maintenance préventive surdimensionnées (inutiles), ainsi qu'un trop grand nombre d'opérations de maintenance curative (après panne). Cela doit être fait à court terme et à long terme pour optimiser en permanence la performance de la flotte exprimée en termes de coûts de maintenance et de disponibilité globale;
4. **Environnement hétérogène:** malgré le fait qu'une flotte soit composée de systèmes mobiles homogènes, chacun d'entre eux a des vies différentes. Par exemple, la livraison de systèmes mobiles aux exploitants de flottes n'est pas instantanée et dépend des capacités de fabrication. De plus, chaque système mobile évolue différemment, rencontre des contextes différents et est maintenu et révisé différemment. Les différences entre eux augmentent ainsi avec le temps. En plus de ces aspects, une architecture de surveillance d'une flotte doit prendre en compte d'autres applications embarquées et non embarquées existantes qui traitent des fonctions liées à la surveillance (par exemple, avec des microprogrammes(firmwares) différents). L'architecture doit aussi interopérer et partager

ses résultats avec un ensemble d'applications variées en tenant compte de diverses contraintes existantes (possibilité d'embarquer des systèmes ou non, capacité de calcul, etc.)[77];

5. **Modularité:** une telle architecture de surveillance doit être suffisamment modulaire et universelle pour faire face aux évolutions rapides des technologies d'information et de communication (par exemple, Internet des objets). Elle doit également supporter cette évolution dans les systèmes mobiles eux-mêmes (par exemple, remplacement de composants, mises à jour de firmwares, mises à niveau d'équipements technologiques – « overhaul »);
6. **Mise à jour des connaissances:** En lien avec le point précédent, cette spécification traduit la nécessité pour une telle architecture de surveillance, de faciliter les mises à jour des connaissances lorsqu'un équipement neuf ou révisé est intégré dans un système mobile. Pour chacun, le comportement de cet équipement nouveau / mis à jour et son interaction avec les autres doivent être correctement caractérisés pour ensuite assurer un suivi et un diagnostic corrects. Ce processus peut être réalisé par des humains et / ou des processus informatisés fusionnés avec l'architecture de surveillance;
7. **Réactivité et fiabilité des dispositifs de surveillance du système:** Les informations de surveillance en temps réel des (sous-) systèmes, quel que soit le niveau, doivent être fiables, robustes, auto-explicatives et transmises au CM par le biais de l'architecture de surveillance pour améliorer les opérations maintenance [4]. Les temps de réaction doivent être aussi rapides que possible, car un retard de réaction élevé entraînera un diagnostic tardif ou des décisions incohérentes;
8. **Sécurité:** les données et les événements générés par une architecture de surveillance devraient garantir l'authenticité, la confidentialité, l'intégrité et la disponibilité[78]; Une communication sécurisée est nécessaire pour éviter les attaques externes et les tentatives de piratage pouvant nuire aux comportements des (sous-) systèmes et la sécurité des usagers.
9. **Impact organisationnel:** les objectifs de déploiement d'une telle architecture de surveillance ne pourront jamais être atteints sans une profonde réorganisation et une adaptation de la planification et de la gestion des opérations de maintenance. L'organisation se doit de devenir aussi réactive que l'architecture de surveillance elle-même, sinon le déploiement d'une telle architecture est inutile.

3. Conclusion

Ce premier chapitre a permis d'introduire les enjeux actuels que rencontrent les industriels et les exploitants dans le domaine de la gestion de la maintenance d'une flotte de systèmes complexes, mobiles et homogènes et de spécifier un ensemble de besoins à respecter lors de la conception d'une architecture de surveillance adaptée à ces systèmes, et en réponse aux enjeux identifiés.

Pour ce faire, nous avons discuté des différents aspects scientifiques relevant de cette thèse. Nous avons également borné le cadre de notre étude dans l'objectif d'améliorer la maintenance durant la phases d'exploitation de flotte des systèmes mobile. Nous avons notamment limité notre champs d'intervention sur une stratégie de maintenance de type proactive. Nous avons également présenté les fonctions et architectures de la surveillance qui constitue une étape clé lors de la maintenance proactive d'une flotte.

Nous avons alors présenté deux typologies d'architecture de surveillance. Ces typologies nous serviront dans le chapitre suivant à positionner la littérature dans le domaine des architectures de surveillances et ainsi nous permettre de proposer dans le chapitre III une contribution claire dans ce domaine.

Nous avons finalement présenté la problématique globale à laquelle nous nous intéressons, problématique à partir de laquelle nous avons déduit un certain nombre de spécifications à respecter lors de la conception d'une architecture de surveillance.

Ch. II Etat de l'art dans le domaine des architectures de surveillance d'une flotte de systèmes mobiles

Introduction

Le précédent chapitre a mis en évidence un ensemble de spécifications à considérer pour concevoir une architecture de surveillance associée aux systèmes cibles considérés dans nos travaux. Il a permis de présenter les enjeux principaux relatifs à l'amélioration en continu de la fiabilité, maintenabilité, disponibilité et sécurité d'une flotte de systèmes complexes, mobiles et homogènes tout en assurant l'optimisation des coûts de maintenance.

Ce chapitre fournit une analyse de la littérature existante pour la surveillance d'une flotte de ce type de systèmes. Cette analyse contient une étude de l'état de l'art des architectures de surveillance et propose une typologie des architectures qui permet de positionner les différentes contributions. Les contributions citées ont été choisies en fonction des **trois critères** suivants. **Tout d'abord**, une référence est examinée si les auteurs suggèrent une contribution possible à la définition d'architectures de surveillance remplissant au moins l'une des neuf spécifications présentées dans la partie (2.6) du chapitre I. **Deuxièmement**, chacune des architectures étudiées se doit d'une part de disposer d'une interface de haut niveau (niveau flotte) avec le CM de la flotte où sont prises les décisions de maintenance et d'autre part disposer d'interfaces de niveau inférieur reliées aux équipements, systèmes ou composants d'un système mobile de la flotte, en fonction de la granularité de l'architecture conçue. **Troisièmement**, les contributions appliquées au domaine transport ont été favorisées, mais des applications dans d'autres domaines (par exemple, la fabrication ou la logistique) ont été considérées si elles fournissaient ou pourraient apporter des idées, expériences et des perspectives intéressantes concernant notre cas. Les meilleures pratiques, ainsi que les lacunes et les limites de la littérature existante, ont été analysées, ce qui nous permettra de proposer dans le chapitre III une contribution positionnée sur la base de cette revue de la littérature.

Comme indiqué dans la partie 2.3 du chapitre I, une architecture de surveillance est définie comme un système informatisé intégrant les fonctions de surveillance, et les algorithmes (logiciels) à supporter, ainsi que la localisation des différents éléments de calcul (hardware) en charge du flux d'événements, d'informations, et connaissances numérisées d'une flotte des systèmes afin de suivre leurs évolutions et impacts sur l'exploitation de cette flotte. Cette architecture de surveillance se doit d'être intégrée avec d'autres systèmes d'information existants[18].

Comme introduit, notre typologie est adaptée de celle de [7]. Elle est composée de quatre catégories d'architecture de surveillance: « centralisée », « edge-centralisée », « décentralisée » et « décentralisée et coopérative », qui définissent quatre types d'architecture de surveillance pour l'aide à la maintenance d'une flotte de systèmes complexes, mobiles et homogènes. Ces différents types sont décrits ci-dessous et des exemples tirés de la littérature sont fournis. Les éléments relatifs à la spécification cibles introduites sont également discutés.

1. Architectures de surveillance centralisées d'une flotte

Dans ce type d'architecture (voir la Figure 2. 1), tous les activités de surveillance sont centralisés dans un CM, qui est chargé de la collecte, du traitement, de surveillance et de l'optimisation de tous les événements bruts issus de la flotte [18].

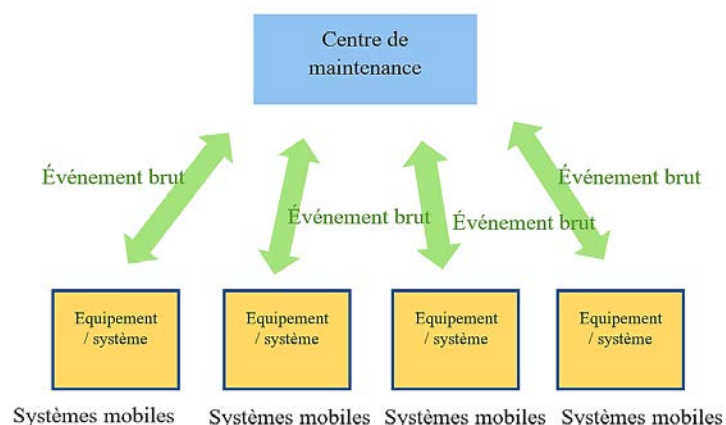


Figure 2. 1: Architecture de surveillance centralisée

Un nombre croissant d'auteurs ont abordé ce type d'architecture[18] et diverses applications dans différents secteurs du transport peuvent être identifiées. La plupart d'entre elles étudient principalement la question du Big Data qui est essentielle dans ce type d'architecture. Dans le secteur de l'aviation par exemple, on peut citer le développement d'une plate-forme d'application et d'analyse de données massives pour la gestion de la santé des avions civils[79].

Dans le secteur ferroviaire, les chemins de fer néerlandais utilisent le big data pour faciliter les décisions de maintenance[80]. On peut également citer la plateforme de gestion des flottes centralisée TrainTracer[81] conçue par le constructeur Alstom pour collecter et traiter les données en temps réel issues des trains, équipés de capteurs embarqués surveillant divers sous-systèmes tels que le convertisseur auxiliaire, les portes, freins, circuit de puissance et l'inclinaison. La Figure 2. 2 contient une illustration graphique de TrainTracer d'Alstom. Les trains commerciaux sont équipés de systèmes de positionnement (GPS) et de communication ainsi que de capteurs embarqués qui surveillent l'état de divers sous-systèmes du train et fournissent un flux de données en temps réel. Ces données sont transférées via un réseau sans-fil vers des serveurs centralisés où elles sont stockées, exploitées et analysées par l'équipe de support, les mainteneurs et les opérateurs via un accès intranet / internet sécurisé.

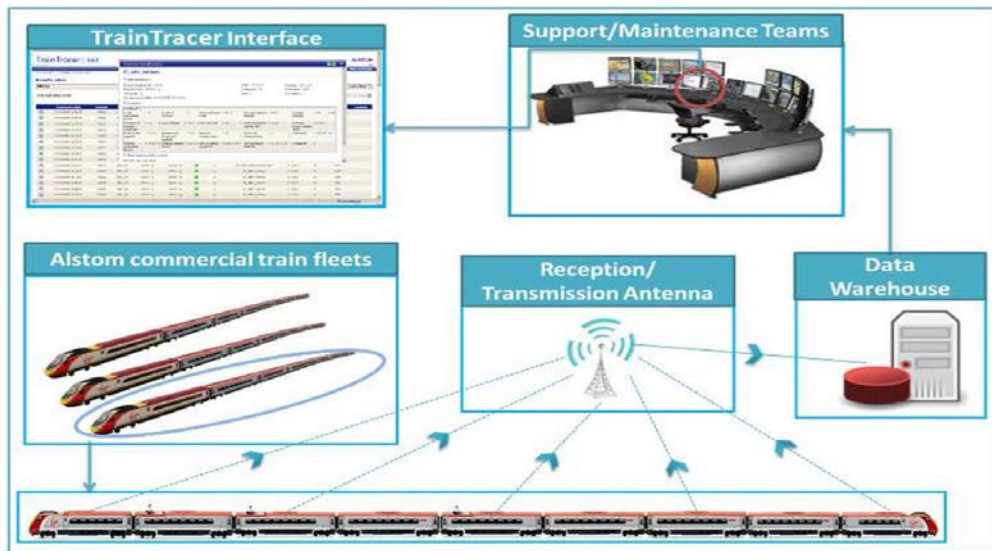


Figure 2. 2: Une illustration des services centralisés de surveillance dans le transport ferroviaire, l'Alstom Traintracer[81]

Dans le secteur automobile, des applications Big Data ont été développées pour le suivi en temps réel des opérations de trafic et de la sécurité sur les autoroutes urbaines[82]. On peut citer également le projet de recherche européen EBSF_2(European Bus System of the Future -(2015 – 2018))[83], qui vise à développer une nouvelle génération de bus en associant de nouvelles technologies et infrastructures de véhicules aux meilleures pratiques opérationnelles. Le système de surveillance développé a pour but d'analyser les données brutes provenant des capteurs de lubrifiants afin d'évaluer la qualité de l'huile, détectant ainsi les pannes potentielles ou prospectives et remplaçant à l'avance les pièces de rechange (voir la Figure 2. 3). L'intérêt majeur pour les opérateurs au travers de ce type d'architecture est d'optimiser les opérations de maintenance principalement au travers de la réduction des coûts d'exploitation et en innovant sur des nouveaux composants tels que les filtres à huile et les capteurs, contribuant ainsi à améliorer l'efficacité globale du système de bus et réduire les effets de la pollution due aux émissions.

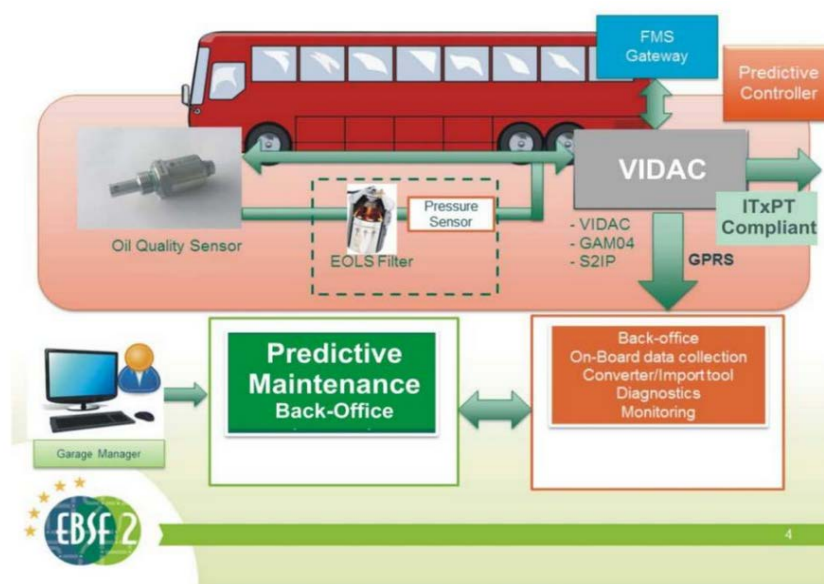


Figure 2. 3: Le processus de surveillance centralisé des systèmes mobile[83]

Ce type d'architecture a été utilisé dans le secteur maritime en Norvège[84]. On peut citer également l'architecture de gestion de la santé à l'échelle de la flotte proposé dans[20] qui gère les connaissances correspondantes pertinentes découlant à la fois de la modélisation et de la surveillance des systèmes de la flotte. L'architecture propose un modèle de connaissance où les informations sont structurées au moyen de contextes spécifiques. Ces contextes permettent de prendre en compte les similitudes et les hétérogénéités des composants de la flotte. Les données de l'équipement surveillé sont ainsi naturellement considérées dans leur contexte et améliorent l'identification de l'état de santé correspondant. Le modèle de connaissance de la flotte a été réalisé et intégrée à la plate-forme PHM industrielle KASEM (Knowledge and Advanced Services for Execution and Monitoring) (voir Figure 2. 4).

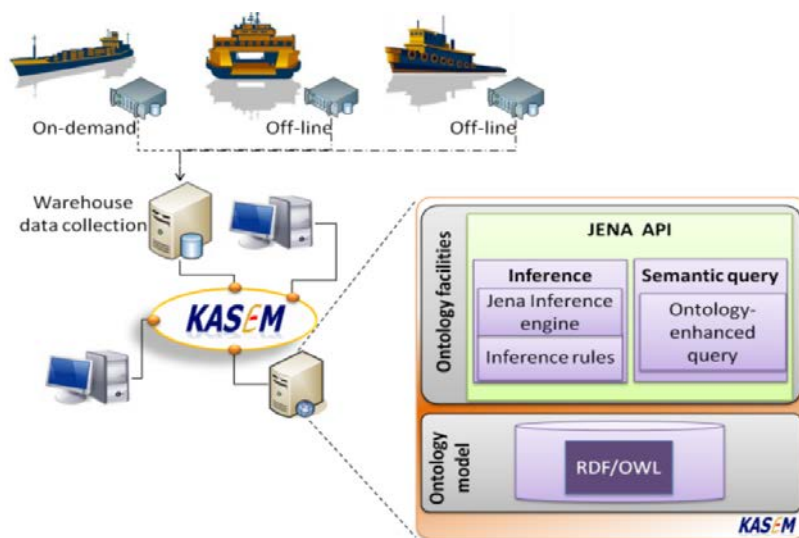


Figure 2. 4 : Architecture typique d'un système PHM à l'échelle de la flotte

La technologie Cloud, en plus de l'Internet des objets, est souvent utilisée pour implémenter des architectures de ce type, voir par exemple [85].

Les avantages de ce type d'architecture sont relatifs à leur facilité de déploiement, à leur capacité à faciliter la collecte et le stockage des données, et à leur adaptation facile aux exigences introduites pour les systèmes embarqués de faible complexité. Ils offrent également de puissantes solutions d'analyse de données, y compris le traitement de gros volumes de données pour faciliter les fonctions de surveillance de l'ensemble d'une flotte. Enfin, ils sont compatibles avec les solutions technologiques modernes (technologie cloud, Internet des objets, etc.). Le premier inconvénient concerne certaines limitations en termes de fiabilité et de précision du diagnostic car ces architectures prennent rarement en compte l'environnement physique des systèmes mobiles et les conditions environnementales dans lesquelles les sous-systèmes opèrent. Un deuxième problème est relatif à la difficulté de contrôler le temps, ce qui est nécessaire pour traiter les données et générer des conseils et des connaissances au bon moment pour le CM. De plus, l'acquisition des données n'est pas systématique en temps réel et dépend de plusieurs facteurs externes (qualité du réseau de transmission, capacité locale de stockage et mémorisation des données, etc.). Enfin, ces architectures sont relativement rigides car la configuration des systèmes (et flottes) surveillés nécessite souvent une stabilité à long terme: la modification de l'équipement ou l'insertion de

nouveaux systèmes est rarement abordée, car l'une ou l'autre de celles-ci peut nécessiter beaucoup de temps pour adapter les logiciels.

2. Architectures de surveillance Edge-centralisées d'une flotte

Ce type d'architecture est plus récent et animé par l'émergence de certaines technologies. Il est caractérisé par l'introduction de « réseaux de nœuds de nuage » intermédiaires pour le calcul et la communication, en utilisant une technologie appelée « edge computing » qui favorise la création d'un niveau intermédiaire entre les systèmes mobiles et le CM. L'« edge computing » pousse les applications, les services, les fonctions et les données informatiques des clouds centralisés ou des périphériques contraints (capteurs, objets intelligents par exemple) vers des nœuds distribués au bord des réseaux pour améliorer les performances des systèmes sous surveillance[86]. Ainsi, les fonctions de surveillance bénéficient du déchargement du calcul, du stockage et de l'accélération vers de tels nœuds de nuage grâce à la communication entre nœuds et au partage des ressources[86]. La Figure 2. 5 illustre ce type d'architecture qui peut être considérée comme une évolution de l'architecture de surveillance centralisé.

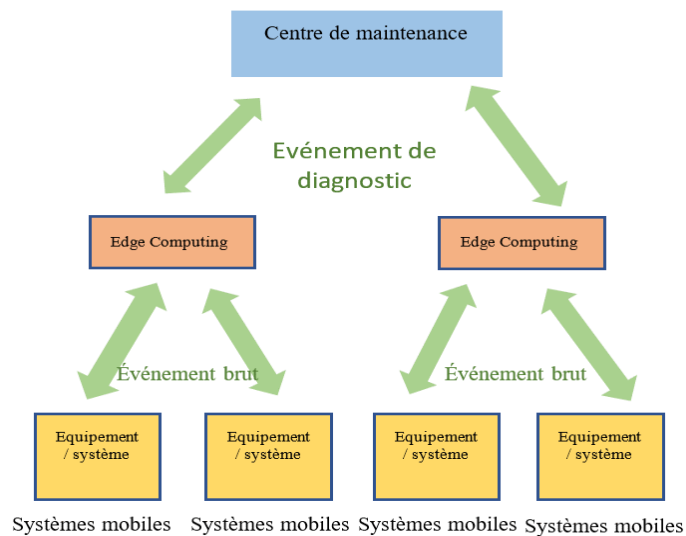


Figure 2. 5: Architecture de surveillance Edge-centralisé

Ce type d'architecture est ainsi assez récent. Il existe quelques applications dans le secteur des services, notamment AWS Lambda @ Edge¹, Google Cloud², IBM Watson³ Microsoft Azure IoT Edge⁴, EdgeConnex⁵, et Equinix⁶. Ces applications peuvent prendre en charge tous les systèmes mobiles tels que les drones, les systèmes fixes et les applications logicielles utilisant l'Internet des objets et la réalité augmentée.

¹ docs.aws.amazon.com/lambda/latest/dg/lambda-edge.html

² <https://cloud.google.com/>

³ <https://www.ibm.com/internet-of-things>

⁴ azure.microsoft.com/en-us/services/iot-edge/

⁵ www.edgeconnex.com

⁶ www.equinix.com

Dans le secteur des transports, on recense le déploiement des technologies de l'application IBM Watson par l'opérateur ferroviaire français SNCF⁷. Ce déploiement concerne des milliers de capteurs sur les trains et les voies qui envoient en toute sécurité des dizaines de milliers de points de données vers le cloud, le tout en temps réel (voir la Figure 2. 6). Leurs analyses permettent à la SNCF d'anticiper lorsqu'un équipement spécifique, par exemple composant de signal défectueux, a besoin d'être réparé. En prédisant quand la maintenance est nécessaire, la SNCF peut empêcher la mise hors service des trains tout en évitant des réparations plus coûteuses.



Figure 2. 6: Architecture de capteurs d'un train de la SNCF⁸

Les avantages de ce type d'architecture comprennent leur capacité à limiter et contrôler la transmission de gros volumes de données et d'événements au sol vers le Centre de Maintenance. Ils sont également compatibles avec les solutions technologiques modernes et permettent l'intégration d'un niveau de modularité de base par le découplage des fonctions et des algorithmes. Transposés au secteur des transports, les inconvénients peuvent être les mêmes que ceux identifiés pour le type centralisé, sauf en ce qui concerne la modularité.

3. Architectures de surveillance décentralisées d'une flotte

La Figure 2. 7 schématise ce type d'architecture où les unités de surveillance embarquées au sein des systèmes mobiles fonctionnent indépendamment. Ils ne communiquent pas entre eux et n'utilisent que les observations limitées de leurs sous-systèmes. Les résultats de surveillance sont transmis au centre de maintenance sol[87].

⁷ <https://www.ibm.com/blogs/internet-of-things/sncf-iot-french-railways/>

⁸ <https://www.carnets-eolane.com/lora-picogateway-msm-iot/>

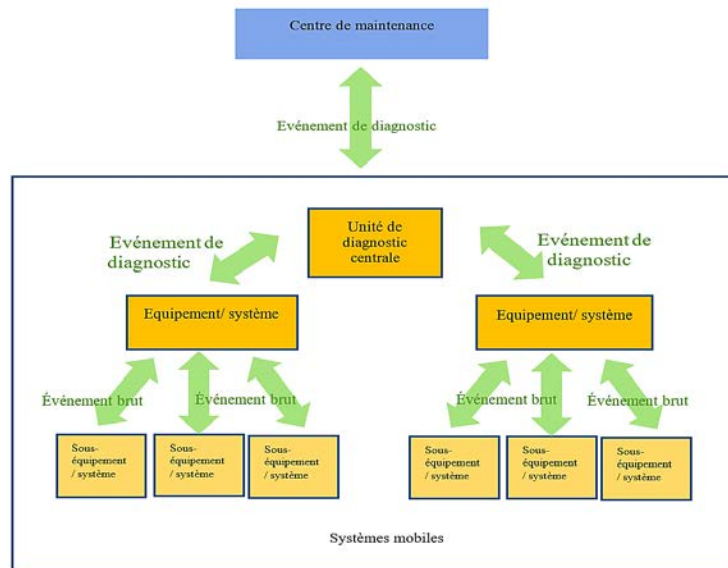


Figure 2. 7: Architecture de surveillance décentralisée

Le CMS(Central Maintenance System)[88] développé dans le secteur du transport aérien est un bon exemple de ce type d'architecture. L'architecture proposée fournit un système de maintenance centralisé au bord des avions, tel que décrit dans la norme ARINC 624 (Aeronautical Radio, Incorporated) [89]. Cette architecture est composée d'une unité de de maintenance centrale qui reçoit des rapports BIT (built-in test) et des données provenant de capteurs ACMS (Airplane Condition Management System) en continu et qui effectue une analyse d'isolement de défauts et de localisation pour trouver la cause profonde parmi les défauts signalés ou détecter un défaut avant son apparition (pronostic). L'état de l'aéronef peut être vu sur les écrans MAT (Maintenance Access Terminal) dans le cockpit ou sur les MAT portables des techniciens (voir la Figure 2. 8).

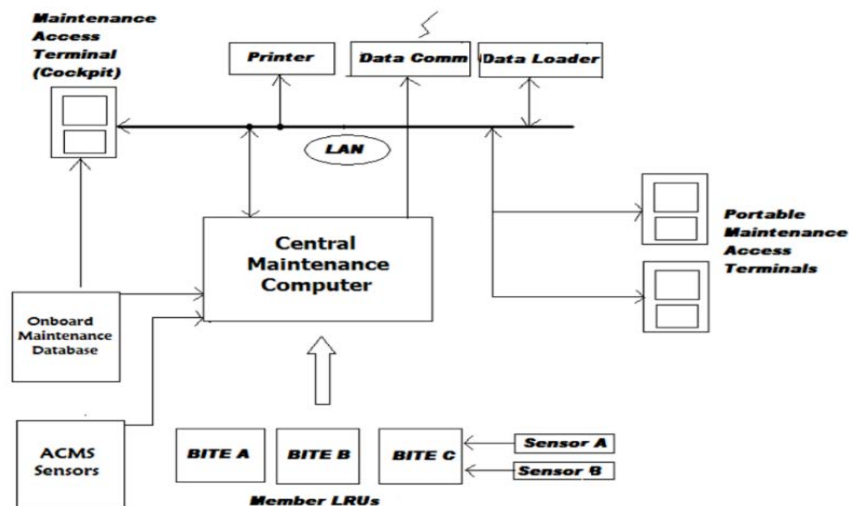


Figure 2. 8: Vue d'ensemble du CMS[88]

Un autre exemple est celui proposé dans le cadre du diagnostic de véhicule routiers[90]. Cette architecture déploie une stratégie de diagnostic de pannes dans un système SDK (Smart Distance Keeping). Un système SDK est un système de régulateur de vitesse adaptatif et évolutif mis en œuvre dans des véhicules Renault-Volvo pour augmenter la sécurité entre les véhicules et l'obstacle

devant. Les auteurs ont proposé une approche décentralisée permettant de réaliser un diagnostic en ligne sans calculer le modèle global et d'accomplir la plus grande partie du travail localement en évitant d'énormes trafics d'informations extra diagnostics entre composants (voir la *Figure 2. 9*). L'objectif principal visé est l'analyse des mesures, issues des éléments du SDK, afin de détecter, de localiser et d'identifier certaines défaillances pouvant survenir.

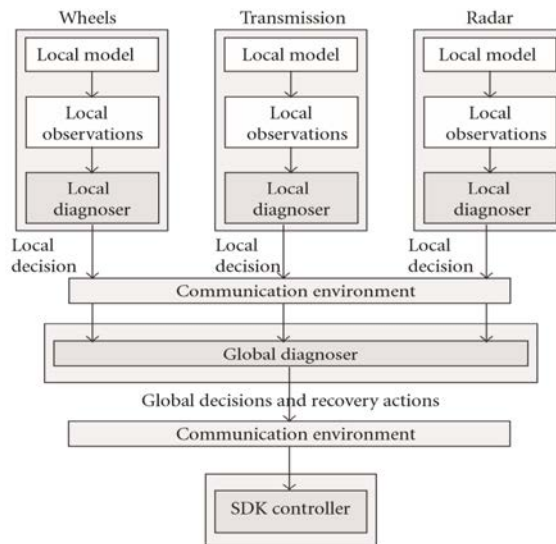


Figure 2. 9: Diagnostic de pannes décentralisé basé sur un modèle du système SDK[90].

Dans le secteur maritime, on peut citer une architecture extensible CBM pour la maintenance de la flotte navale[91]. Cette architecture repose sur les standards DDS (Data Distribution Service) et OSA-CBM (Open System Architecture for Condition-Based Maintenance) (voir Figure 2. 10) pour répondre aux principaux défis architecturaux dans le domaine maritime. Elle fournit une analyse prédictive avancée des équipements mécaniques, électriques et informatiques embarqués de la Marine. Le défi de l'interopérabilité est traité à l'aide de l'architecture pilotée par le modèle MDA (Model-Driven Architecture) en transformant les données des capteurs en modèles de données standard rigoureusement spécifiés.

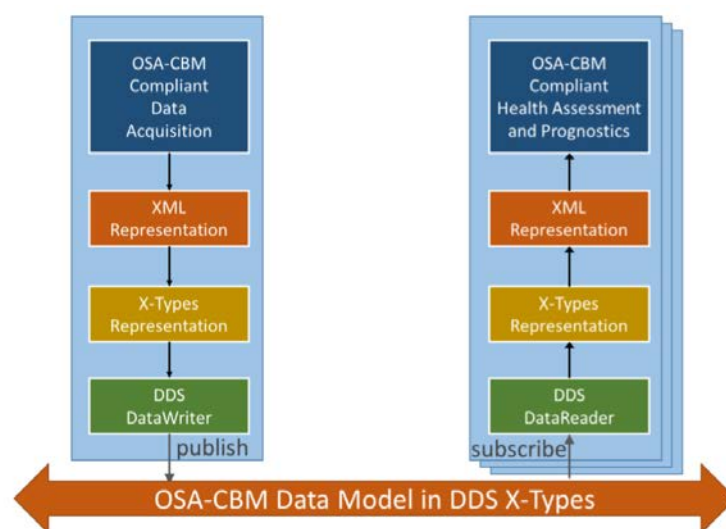


Figure 2. 10: Architecture de bus de données pour la maintenance conditionnelle utilisant DDS et OSA-CBM[91]

Les avantages de ce type d'architecture comprennent sa capacité à limiter et contrôler la transmission de gros volumes de données et d'événements au sol vers le centre de maintenance. Le diagnostic contextualisé en temps réel est également possible. En outre, il bénéficie d'une grande précision et est compatible avec le besoin en modularité des capacités de diagnostic de systèmes complexes composés de sous-systèmes. Cependant, de telles architectures intègrent difficilement des mécanismes permettant des analyses croisées et des redondances pour discriminer les erreurs. Par conséquent, elles peuvent conduire à des diagnostics erronés difficilement identifiables lorsque les sous-systèmes sont soumis à des interférences provenant d'autres sous-systèmes ou s'ils sont en interactions entre eux.

4. Architectures de surveillance décentralisées et coopératives d'une flotte

Dans ce type d'architecture, la surveillance est associée à un sous-système du système mobile selon un niveau de granularité souhaité, allant des capteurs, composants, équipements au système mobile lui-même ou même à sa flotte. La décomposition hiérarchique d'un système en sous-systèmes s'applique: un système à un niveau supérieur contient le système des niveaux inférieurs, comme représenté sur la Figure 2. 1. Ce type d'architecture permet aux unités de même niveau de coopérer et d'obtenir ainsi des informations supplémentaires pour enrichir leurs observations locales et fournir une surveillance plus robuste[57]. Les échanges impliquent des flux d'informations supplémentaires à bord du système mobile (mémorisation de données brutes, suivi, diagnostic coopératif, estimation de RUL, par exemple), ce qui nécessite des fonctions de calcul locales plus puissantes[92]. La Figure 2. 11 schématise ce type d'architecture.

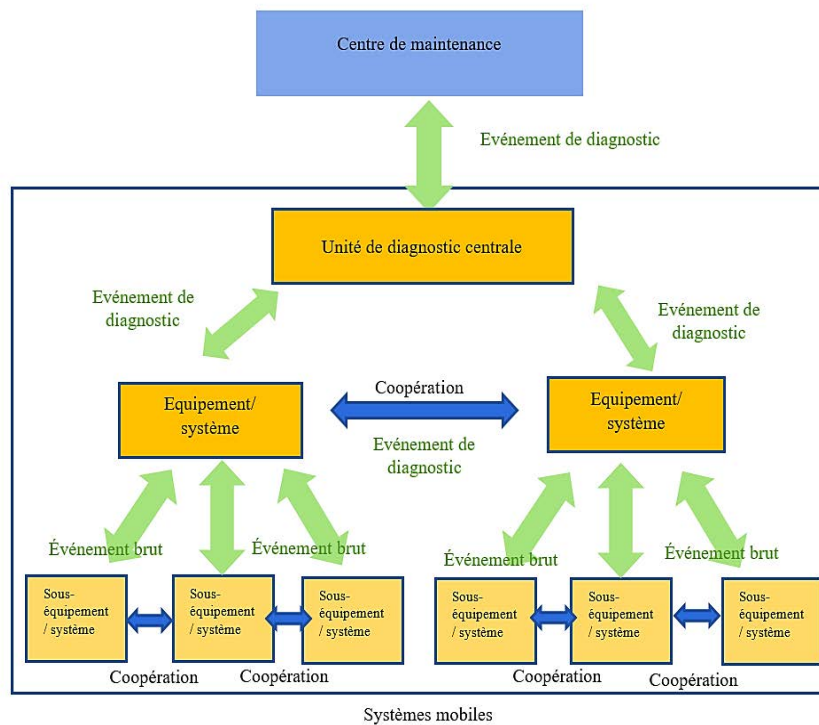


Figure 2. 11: Architecture de surveillance décentralisé et coopératif

Peu de contributions illustrent ce genre d'architecture, on peut cependant identifier quelques exemples, dans le domaine de la fabrication, la logistique, aérospatial et le transport.

Dans Les secteurs de la fabrication et de la logistique: cette catégorie d'architecture a été largement étudiée avec les approches de modélisation multi-agents et holoniques[93]. Ces approches offrent des mécanismes typiques construites à partir du paradigme holonique qui est relatif à la récursivité (un holon est à la fois un ensemble, composé d'holons et à la fois une partie, composant d'un holon plus grand) et à la vision Cyber-Physique(un holon est composé d'une partie physique et d'une partie informationnelle)[94].

Un exemple dans le domaine de la fabrication est fourni dans[95]. L'architecture proposée traite des comportements myopes et propose des mécanismes pour contrôler ce comportement. Cette architecture vise à concevoir une solution capable de résoudre le problème d'allocation dans les systèmes de fabrication flexibles, hiérarchiques et autonomes (voir la Figure 2. 12). Cependant cette architecture est destinée à un autre domaine que le secteur de transport, les propriétés spécifiques, comme celles de la modularité et la fiabilité, propres aux systèmes concernés dans cette thèse ne sont pas abordées.

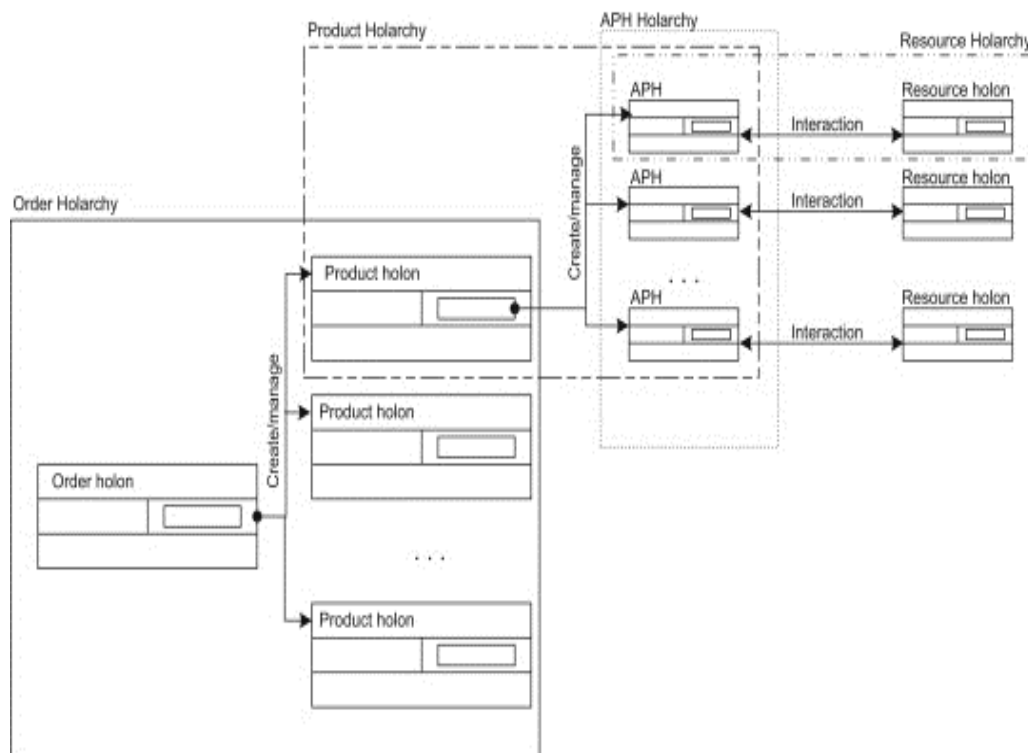


Figure 2. 12: Le contrôle du comportement myope dans les systèmes de production: un cadre holonique[95]

Des exemples dans le domaine de la logistique peuvent être mentionnés, par exemple[96] et [97]. Dans [96], les auteurs proposent une architecture holonique d'un système de gestion du trafic routier en mettant l'accent sur la perte de temps et les accidents provoqués par des facteurs humains tels que la fatigue, la perte de contrôle, la lenteur du temps de réaction, le champ de vision limité, etc. La Figure 2. 13 décrit la structure décentralisée et coopérative des holons, où chaque holon peut demander à d'autres holons des informations sur les distances.

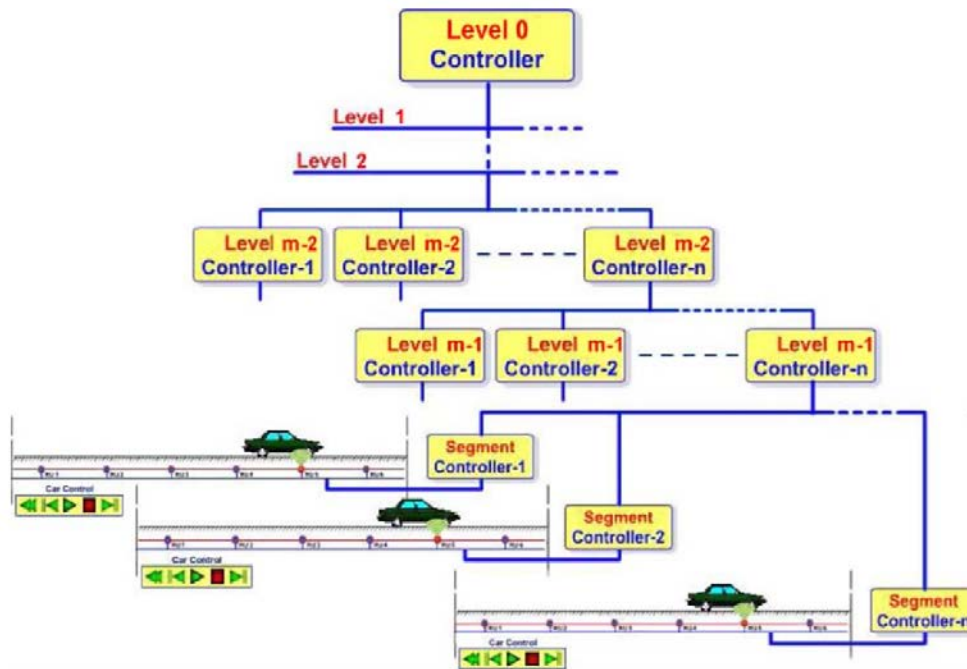


Figure 2. 13: Structure hiérarchique du système de gestion de la route

Dans [97], une architecture holonique est également proposée pour calculer le chemin le plus court dans les réseaux complexes de transport routier. Le concept proposé vise à accroître la flexibilité et l'adaptabilité à autant de niveaux hiérarchiques que nécessaire, il est donc utilisable aussi bien pour les petites zones que pour les zones étendues. Néanmoins cette architecture implémente des algorithmes spécifiques ne prenant pas en compte les enjeux liés à la santé des équipements et leurs contextes associés, limitant son applicabilité dans notre contexte.

Dans Les secteurs aérospatial et transport : les architectures de surveillance décentralisées et coopératives développées pour la surveillance de systèmes mobiles sont relativement peu nombreuses. Cependant, on peut identifier quelques contributions comprenant ce type d'architecture. Par exemple dans [98], les auteurs ont implémenté dans des systèmes mobiles des entités de diagnostic reliées à une base de données commune et pouvant s'échanger des informations par le biais d'une mémoire partagée. Une méthode de diagnostic spécifique est exécutée par les différentes entités qui composent cette architecture. Cependant, les informations contextuelles des sous-systèmes surveillés ne sont pas considérées pour améliorer la performance du diagnostic. L'architecture « Surfer » proposée dans [99] constitue un autre exemple applicatif dans le domaine du transport. Sans aller jusqu'à proposer une architecture opérationnelle, la contribution contient un ensemble de spécifications (méthode Surfer) qui traduisent une manière originale d'envisager les problématiques liées au Big Data, dans le but de transformer les données brutes en connaissances de haut niveau utilisables et exploitables par les acteurs de la maintenance. Cet ensemble de spécification repose sur le concept d'agents de surveillance autonomes qui a été breveté [100]. Le projet VIPR (Vehicle Integrated Prognostic Reasoner) peut également être mentionné. Il a permis de définir un modèle d'architecture pour la surveillance de véhicules aérospatiaux [101]. Les entités de l'architecture VIPR, nommées HM (Health Manager), sont réparties suivant trois niveaux hiérarchiques (niveau équipement, niveau régional et niveau véhicule) et attachés à la structure du système sous surveillance. Ils communiquent et coopèrent pour établir un diagnostic global d'un système (voir la Figure 2. 14). Néanmoins, Les entités HM sont

généralement associés à des technologies spécifiques et les informations contextuelles ne sont pas exploités.

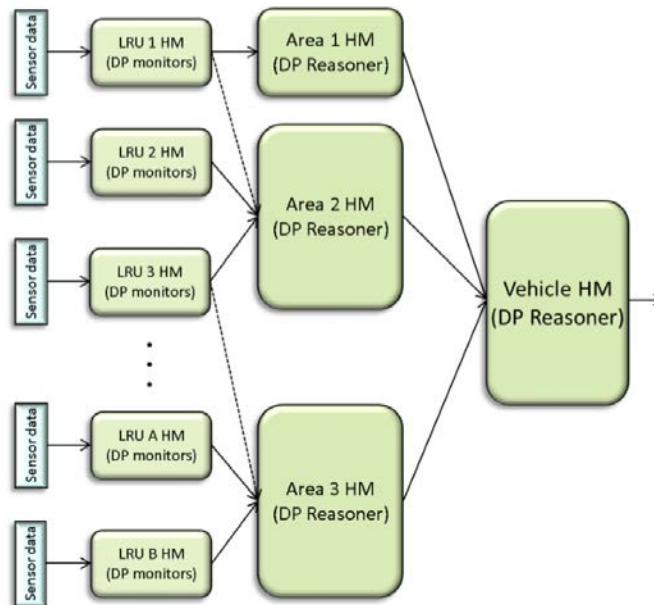


Figure 2. 14: Architecture VIPR, adaptée de[102].

Un autre exemple est l'architecture EMH (Embedded Monitoring Hierarchy) proposée dans[4]. Le paradigme holonique a été utilisé pour organiser l'architecture dans laquelle les unités sont des holons intégrant à la fois la partie physique du système surveillé et son unité de surveillance. Cette approche permet la récurrence des processus de surveillance (un holon étant composé de holons plus petits et plus localisés surveillant des composants plus petits). A chaque niveau de la décomposition holonique du système mobile, un holon est formé pour chaque système soumis à des fonctions de surveillance (voir la Figure 2. 15).

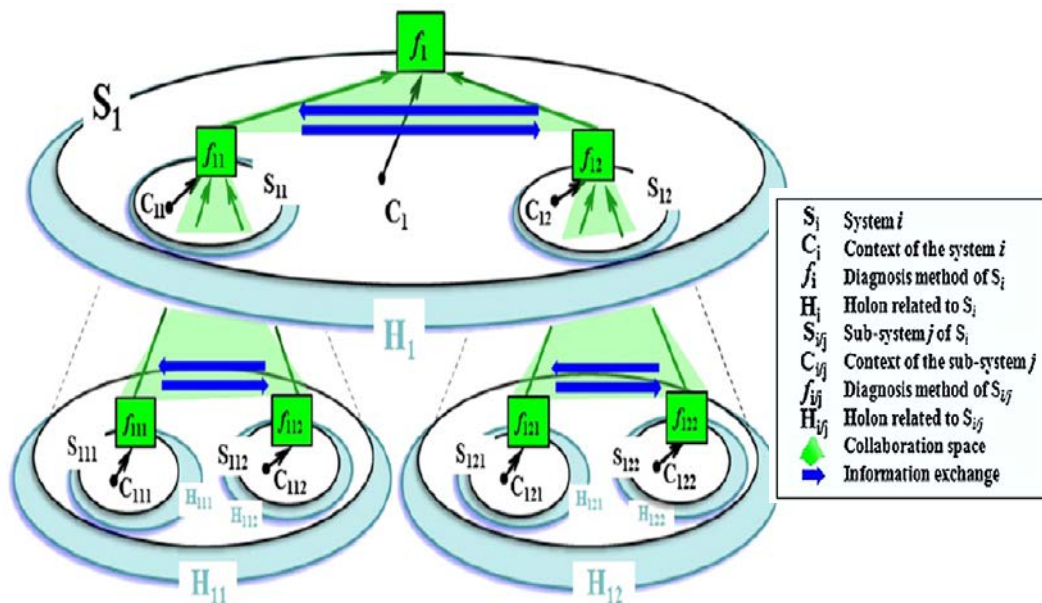


Figure 2. 15: Architecture EMH (Embedded Monitoring Hierarchy) [4]

Cette architecture est basée sur la norme ISO 13374 [41], [103] voir la Figure 2. 16. Les processus associés aux couches fonctionnelles inférieures de la norme ISO 13374 sont intégrés dans les holons des systèmes mobiles (couches #1 acquisition de données, #2 manipulation de données, #3 détection et #4 évaluation de santé) tandis que les couches supérieures (couches #5 pronostic et #6 élaboration de conseils) sont traitées au niveau du CM.

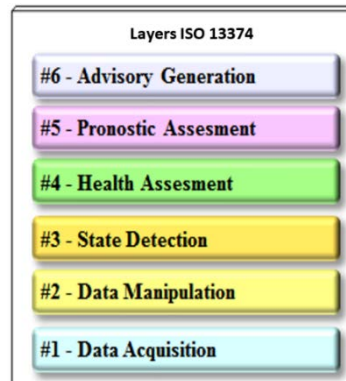


Figure 2. 16: Modèle d'architecture ISO 13374[104]

L'architecture EMH a été spécifiquement conçue pour surveiller l'accès aux portes des trains sans aborder le niveau de la flotte. Elle contient des algorithmes non paramétriques, limitant ainsi sa généralité. Cette architecture ne répondait pas aux spécifications d'hétérogénéité et de modularité car il n'y avait qu'une seule distribution verticale possible des couches ISO et était limitée en terme de déploiement.

Enfin, pour terminer, mentionnons l'architecture fonctionnelle IEDS (Intelligent Embedded Diagnosis System) proposé dans[75] qui se place dans le contexte de DDIR (Détection de Défauts, Isolement et Reconfiguration) et de la maintenance prédictive. L'architecture intègre des fonctions de surveillance dynamique basées sur les pratiques d'ingénierie système en considérant principalement l'étape de spécification (vue des besoins et des exigences; vue fonctionnelle et scénarios). Cependant, les procédures de validation et d'instanciation nécessaire à la construction d'un IEDS spécifique à partir de l'architecture fonctionnelle générique ainsi que l'intégration des contraintes lié au système sous surveillance n'ont pas été abordées. La Figure 2. 17 illustre cette architecture en présentant comment les fonctions / sous-fonctions peuvent être allouées aux composants embarqués et débarqués et qui sont connectés ensemble.

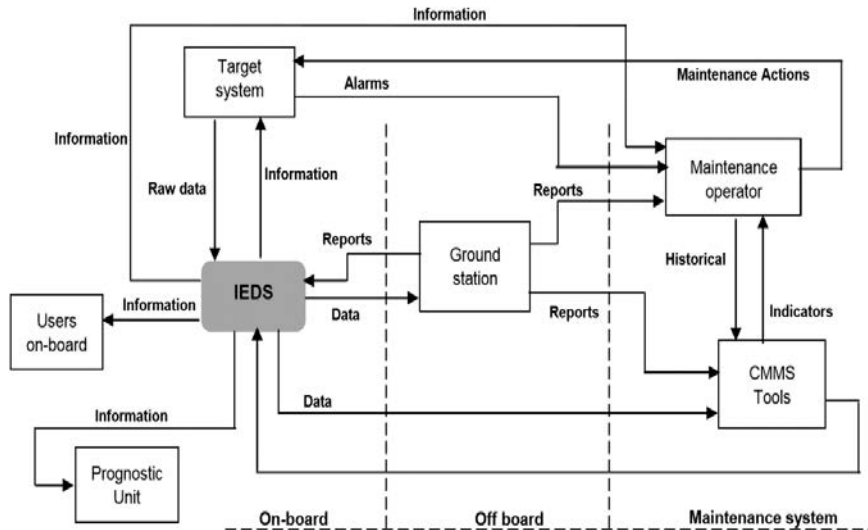


Figure 2. 17: Architecture organisationnelle IEDS en lien avec le système surveillé[75]

Dans cette section, nous avons présenté et illustré quatre catégories d'architecture de surveillance: « centralisée », « edge-centralisée », « décentralisée » et « décentralisée et coopérative ». Nous avons souligné leurs forces et leurs faiblesses par rapport aux spécifications cibles introduites dans le chapitre I. Dans la sous-section suivante nous allons synthétiser cette analyse de l'état de l'art puis nous justifierons le choix fondamental qui aura gouverné cette thèse : le choix d'une ou plusieurs architecture(s) de surveillance parmi les catégories identifiées.

5. Synthèse

Le Tableau 2. 1 résume les résultats de notre analyse. La capacité de chaque catégorie d'architecture de surveillance d'une flotte de systèmes mobile à proposer une réponse pour chacune des neuf spécifications introduites est évaluée sur une échelle allant de + (réponse partielle mais non complètement satisfaisante) à +++ (réponse totalement satisfaisante). Ces évaluations sont notamment tirées des discussions précédentes menées pour chacune des architectures de surveillance d'une flotte de systèmes mobile.

Catégories d'architecture vs. Spécifications demandées	Centralisée	Edge-centralisée	Décentralisée	Décentralisée et Coopérative
Big data	+++	+++	+++	+++
Déploiement	+++	+++	+	+
Gestion des coûts de maintenance	+	++	++	+++
Environnement hétérogène	+	+++	++	++
Modularité	+	+	++	+++
Mise à jour des connaissances	++	++	++	++
Réactivité et fiabilité des dispositifs de surveillance et de diagnostic du système	+	+	++	+++
Sécurité	+++	+++	++	++
Impact organisationnel	++	+++	++	++

Tableau 2. 1 : Évaluation des types d'architecture de surveillance par rapport aux spécifications demandées

À partir de ce tableau, nous pouvons conclure qu'aucun des types d'architecture existants de surveillance d'une flotte de systèmes mobile n'offre une réponse complète aux spécifications introduites précédemment. Cependant, par exemple, des catégories architectures telles que « Décentralisée et coopérative », et « Edge-centralisée » sont en fait complémentaires : le type « Edge-centralisée » nous permet de limiter et contrôler la transmission de gros volumes de données et d'événements au sol et offrir la capacité d'intégration des solutions technologiques modernes et de modularité dans des fonctions de surveillance alors que le type « Décentralisée et coopérative » facilite la surveillance complète du système cible à partir des informations fournies par différentes entités de surveillance situées à bord du système cible et pouvant interagir les unes avec les autres. Elle offre également la précision, l'adaptabilité, la réactivité et la robustesse des connaissances générées. Ce type de complémentarité a guidé nos choix de conception lors de la construction de notre contribution présentée dans le chapitre suivant.

Même si les architectures de la catégorie « Décentralisée et coopérative » (incluant l'architecture EMH) ne sont pas entièrement conformes à nos exigences, ce type d'architecture reste l'une des plus cohérentes avec les spécifications introduites. De notre point de vue, cette catégorie peut être généralisée et adaptée pour être intégrée avec une approche de type « Edge-centralisée » (voir la Figure 2. 18). Plus précisément, grâce à des extensions adéquats, l'architecture EMH pourrait gérer certaines des nouvelles spécifications introduites (environnement hétérogène, modularité et coûts de maintenance). Elle est par conséquent retenue dans la suite de nos travaux pour être généralisée

et articulée avec une architecture de catégorie Edge-centralisé pour fournir une surveillance dynamique à tout type de système ou d'équipement jusqu'au niveau de la flotte.

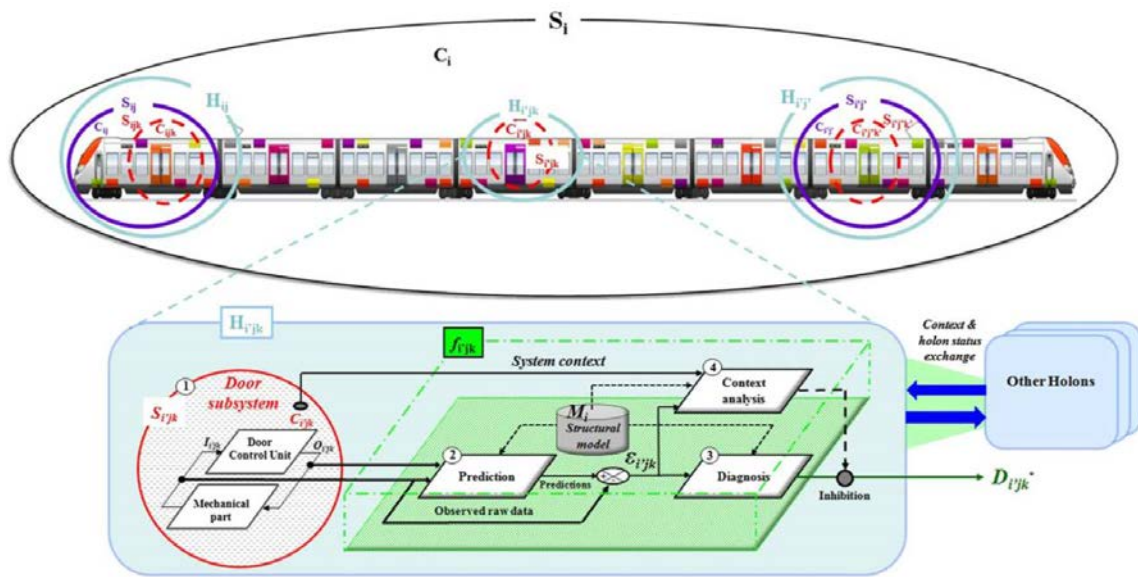


Figure 2. 18: Un modèle holonique d'un train avec application à la surveillance de porte[4]

6. Conclusion

Nous avons présenté dans ce chapitre différentes architectures pour la surveillance d'une flotte des systèmes mobiles, complexes et homogènes. En se basant sur la littérature, nous avons défini quatre types d'architectures de surveillance qui ont été présentés et discutés par rapport aux exigences exprimées dans le précédent chapitre. Aucune de ces typologies existantes d'architectures de surveillance de flotte des systèmes mobile n'offre une réponse complète aux spécifications introduites dans la chapitre I.

Cependant, certains types d'architecture sont complémentaires. Nous avons ainsi opté pour une choix hybride d'architecture de type « Décentralisée et coopérative », et « Edge-centralisée ». Dans ce cadre, nous nous proposons d'étendre et de généraliser l'architecture EMH. Ce choix nous permet, dans la suite de nos travaux, d'ambitionner la création d'une architecture de surveillance originale et efficace. Cette architecture devrait disposer de capacité de limitation et de contrôle de la transmission de gros volumes de données et d'événements au sol et offrirait une capacité d'intégration des solutions technologiques modernes et de modularité dans les fonctions de surveillance.

Le chapitre suivant décrit ainsi une telle architecture de surveillance. Cette architecture, appelée EMH², généralise EMH au niveau de l'utilisation des principes holoniques jusqu'au niveau flotte. La définition d'une approche de modélisation et de gestion d'événements génériques est un autre élément clé de la contribution. EMH² a enfin été conçu pour être applicable à tout type de système mobile ou d'équipement jusqu'au niveau de la flotte et ne fait aucune hypothèse sur la localisation des traitements, embarqués ou au sol.

Ch. III Proposition d'une architecture de surveillance

Introduction

Dans le précédent chapitre, une étude comparative des architectures de surveillance pour l'aide à la maintenance d'une flotte de systèmes mobiles a été présentée. Cette analyse a permis de retenir deux catégories d'architecture complémentaires. Nous avons opté pour un choix hybride d'architecture de type « Décentralisée –coopérative » (via l'architecture EMH) et de type « Edge-centralisé ». Ce choix devrait nous permettre de concevoir à la fois d'une surveillance précise, adaptée, réactive et coopérative à bord du système cible, permettre une limitation et un contrôle de la transmission de gros volumes de données et d'événements au sol et d'offrir une capacité d'intégration des solutions technologiques modernes et de modularité dans les fonctions de surveillance.

Ce chapitre présente les trois éléments clés de notre contribution dénommée EMH² :

1. La définition d'une approche de modélisation et de gestion d'événements génériques appelée « SurfEvent ». Cette approche est considérée dans nos travaux à la fois comme une approche générique de représentations hiérarchique des événements des systèmes mobiles et comme un langage d'interface commun pour l'échanges et la coopération ;
2. La modélisation d'une architecture Holonique appelée « Niveau Holonique » de EMH², inspirée de la norme ISO 13374 qui offre une capacité de déploiement automatisé et qui ne fait aucune hypothèse sur la localisation des traitements, embarqués ou au sol, du système générateur des événements jusqu'au niveau de la flotte;
3. Et enfin, une conception particularisée en fonction des niveaux holoniques afin d'octroyer à l'architecture une capacité de surveillance plus modulaire, réactive et adaptative, tout en facilitant la gestion et l'optimisation du coût global de la maintenance.

Il est important de noter que notre contribution repose sur deux hypothèses. Tout d'abord, l'architecture EMH² ne travaille que sur des événements observés, ce qui revient à dire que tous les événements générés par les systèmes surveillés sont supposés être observables, du moins, par ceux qui conçoivent l'application de notre architecture sur leur système mobile cible. Ensuite, la granularité de l'architecture EMH² est celle des émetteurs d'événements, ce qui revient à dire que l'architecture considère ces événements comme des éléments en entrée, elle ne « descend » ainsi pas au niveau inférieur (par exemple, les composants de ces émetteurs, sauf s'ils émettent de manière récursive des événements).

1. Conception de l'architecture de surveillance EMH²

Dans le cadre de nos travaux, L'approche Holonique a été privilégiée car elle permet de considérer à la fois la partie physique d'un système cible sous surveillance et l'entité de surveillance « active » associée à celle-ci tout en permettant la définition d'une organisation potentiellement réursive, systémique construite à partir du principe « tout et partie »[94]. Cette approche permet ainsi de décrire le système cible de manière réursive et permet de caler par effet miroir (« mirroring ») une surveillance basée sur des entités autonomes, réursives et coopératives que sont les Holons[4].

Dans la suite de ce chapitre, les principaux points de l'EMH² sont présentés. Trois éléments clés de notre contribution sont ainsi mis en évidence: La modélisation d'une forme générique d'évènements appelée « SurfEvent »; L'architecture Holonique de « EMH² » ; et la méthodologie de particularisation des niveaux holoniques de « EMH² » aux différents contextes des systèmes mobiles composant la flotte. Ces éléments clés sont présentés comme suit (voir *Figure 3. 1*) selon une logique itérative où la proposition suivante exploite les éléments précédemment introduits.

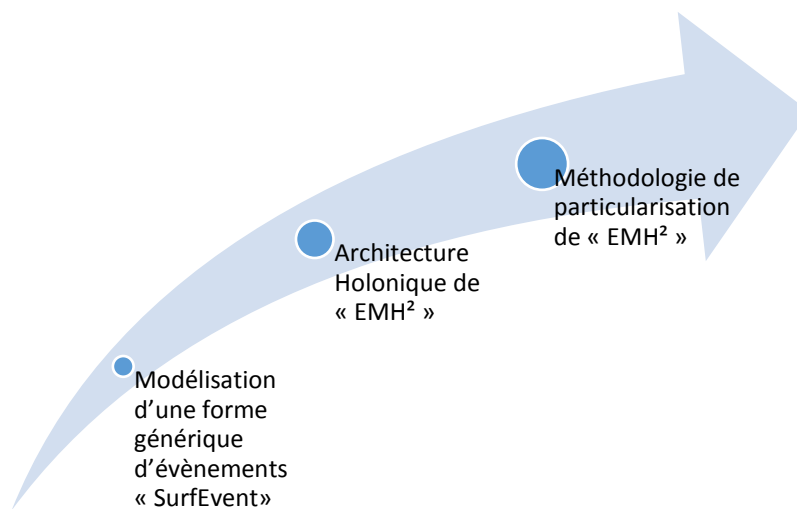


Figure 3. 1: Conception de l'architecture de surveillance

La conception globale de notre proposition est construite sur trois principes fondateurs.

Le premier principe fondateur de notre démarche est **l'alignement des niveaux d'abstraction de la supervision avec les niveaux holoniques**: les fonctions de surveillance des Holons de EMH² sont alignées avec la mission de chaque Niveau Holonique, Les missions de chaque niveau sont inspirées de la norme ISO 13374 : pour chaque couche, un Niveau Holonique est défini (ceci sera détaillé dans la partie « Niveau Holonique de EMH² »). Chacun de ces niveaux Holoniques, bien que spécialisé, dispose de sa propre interface d'élaboration et recommandation de conseils (que nous désignons sous l'acronyme AG : advisory generation). L'ambition est de faciliter la gestion dynamique d'un grand volume de données (logique « big data ») dans un environnement distribué (voir spécification correspondante). La Figure 3. 2, présente, dans sa partie gauche, ces différents niveaux.

Le second principe fondateur est le principe **de décomposition systémique** : par exemple, un holon « train » surveille tous ses accès voyageurs via tous ses holons « portes » qui surveillent chacun leur motorisation via leur holon « moteur ». Un Niveau Holonique fait ainsi référence à un groupe d'Holons appartenant à la même famille. Ce principe de découpage en niveaux systémique, illustré sur la Figure 3. 3, a été validé dans le cadre du développement de l'architecture EMH.

Cette figure met en évidence également notre troisième principe fondateur de **coopérations homogène et hétérogène**: au sein de chaque niveau, les Holons coopèrent en classe homogène, c'est-à-dire pour un ensemble de sous-systèmes surveillés de même nature. Par exemple, tous les holons « porte » qui surveillent un accès voyageur peuvent coopérer pour discriminer des fausses alarmes. Dans un second temps, ils coopèrent avec d'autres systèmes de nature hétérogène (par exemple, une application de type « planification des opérations de maintenance », les systèmes d'information de l'entreprise ou les autres holons « clim », « bogie » d'une voiture dans un train). Ce type de coopération homogène/hétérogène, facilite le suivi, la discrimination, les comportements suspects par les holons pour discriminer les besoins de maintenance préventive ou proactive[105].

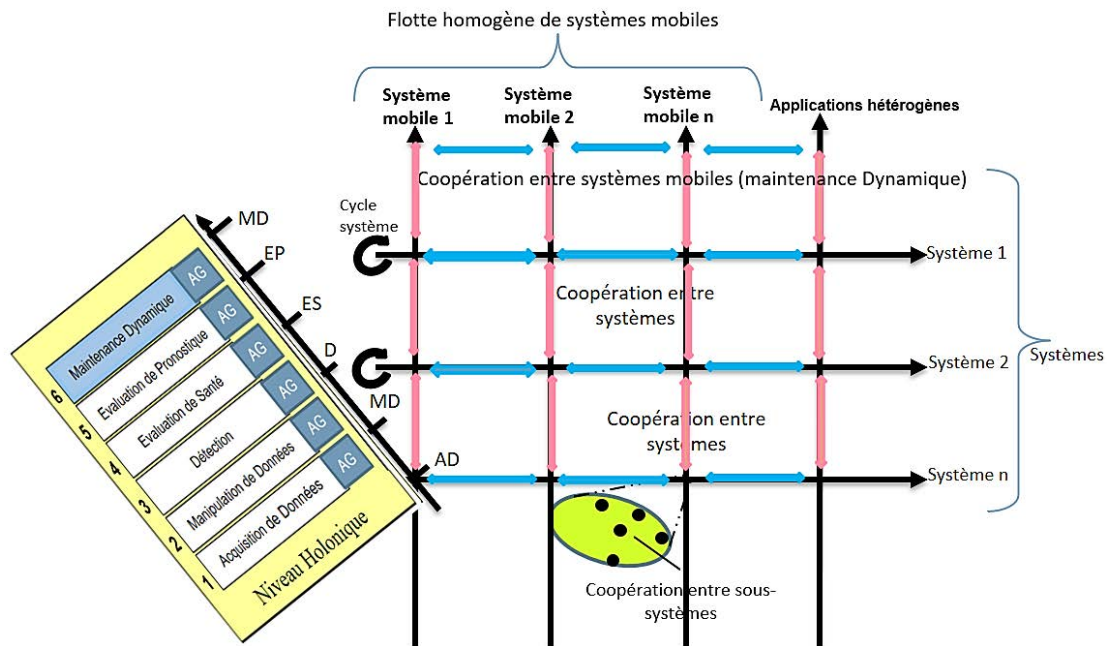


Figure 3. 2: Surveillance coopérative des Holons de l'architecture

Du point de vue de la décomposition d'un système mobile, le résultat d'évaluations récursives de la surveillance des Holons de différents niveaux Holoniques peut être modélisée sous la forme d'une conséquence logique[106], présentée comme suit :

Soit un système « S_i » composé de sous-systèmes « SS_{ij} ». On considère l'évaluation de sa fonction de surveillance $I(S_i)$ comme une conséquence logique « $| =$ » pour l'ensemble de ses sous-systèmes « Γ » si et seulement si il est nécessaire que l'évaluation de chacun de ses sous-systèmes $I(SS_{ij})$ soit valide. On note :

$$\Gamma | = I(S_i) \quad \text{---- Formule 3. 1}$$

Cela revient à dire alors

$$\Gamma = \{ I(SS_{i1}), I(SS_{i2}), I(SS_{i3}), I(SS_{i4}), \dots \dots I(SS_{ij}) \} | = I(S_i) \quad \text{--- Formule 3. 2}$$

La relation entre les sous-systèmes est régie par la structure hiérarchique spécifique à chaque système, tenant compte de sa décomposition récursive, de son contexte, de son hétérogénéité technologique et de son environnement de fonctionnement. De ce fait, la formule résultante de l'évaluation d'une flotte F composée d'un ensemble de systèmes mobiles est alors définie par la conséquence logique suivante :

$$\{I(S_1), I(S_2), I(S_3), I(S_4), I(S_5), I(S_6), \dots, I(S_n)\} | = I(F) \text{ -----Formule 3. 3}$$

A partir des *Formule 3.1*, *Formule 3.2* et *Formule 3.3*, on remarque que la logique formelle d'une conséquence logique intègre une faculté importante de l'architecture de surveillance Holonique, qui est la surveillance d'un système doit passer par la surveillance de ses sous-systèmes qui le compose (voir la Figure 3. 3). L'évolution de la surveillance de systèmes à un instant de temps « t » durant leur cycle de vie, doit prendre en compte les spécificités de leurs sous-systèmes, leurs contextes, leurs états historiques, et leurs évolutions dans le futur (prévision). Dès lors, les formules *Formule 3.1*, *Formule 3.2* et *Formule 3.3* de conséquences logique sont ré-écrites comme suit (voir la *Formule 3.4*) :

$$\Gamma | = I(S_i, t, c_i) \text{ ----- Formule 3. 4}$$

$$\Gamma = \{I(SS_{i1}, t, c_{i1}), I(SS_{i2}, t, c_{i2}), I(SS_{i3}, t, c_{i3}), \dots, I(SS_{ij}, t, c_{ij})\} | = I(S_i, t, c_i) \text{ ---- Formule 3. 5}$$

$$\{I(S_1, t, c_1), I(S_2, t, c_2), I(S_3, t, c_3), I(S_4, t, c_4), \dots, I(S_n, t, c_n)\} | = I(F, t, c) \text{ -----Formule 3. 6}$$

Les formules *Formule 3.4*, *Formule 3.5* et *Formule 3.6*, formalisent ainsi la capacité d'observer les systèmes « S_i » à travers l'observation de leurs sous-système « SS_{ij} », en fonction du temps « t » et en prenant en compte leur contexte « c_i » associées. La base de temps, universelle pour l'ensemble des systèmes de la flotte, exprime l'évolution des durées de vie de chacun des systèmes.

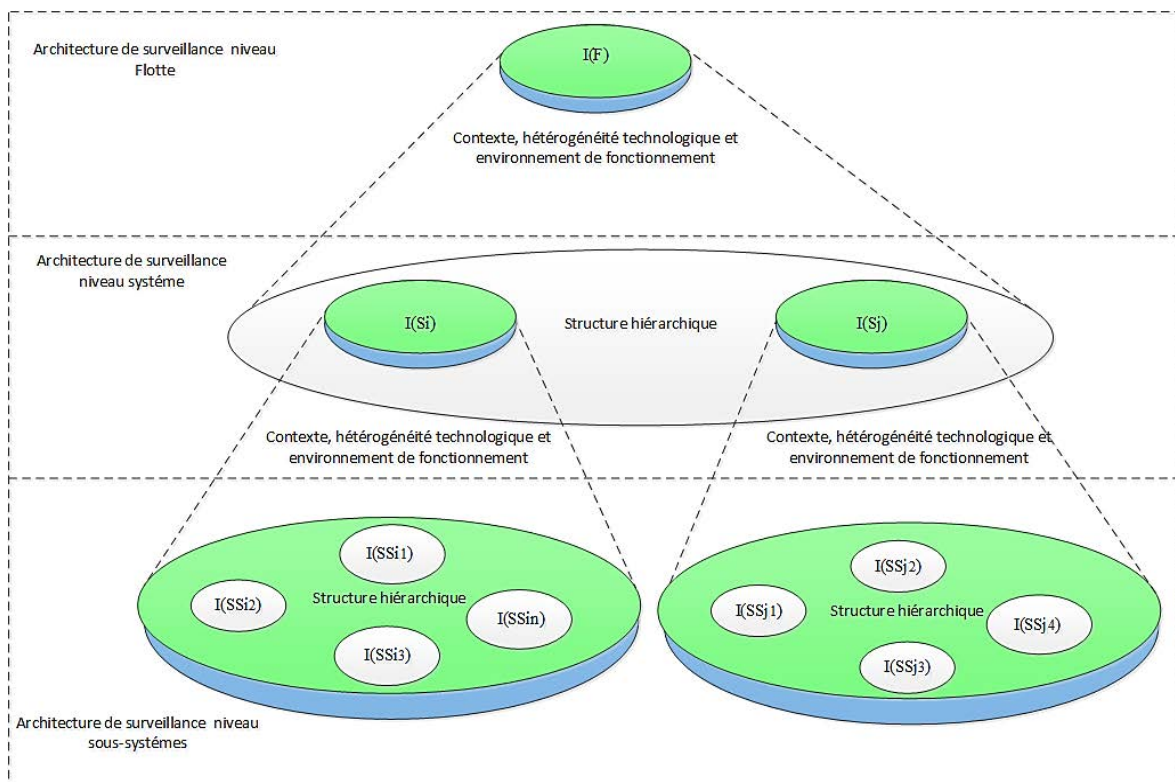


Figure 3. 3: Surveillance récursives des Holons de l'architecture

On peut donc résumer l'architecture EMH² d'une flotte de systèmes mobiles comme un système informatisé intégrant des fonctions de surveillance des niveaux Holoniques. Ce système est capable

de surveiller en continu (dans le temps), l'évolution de chacun des systèmes de la flotte à travers une observation hiérarchisée des sous-systèmes qui le compose.

Dans les sous-section suivants, nous allons détailler les trois éléments clés de notre contribution, à savoir, la modélisation d'une forme générique d'évènements appelée « SurfEvent » ; L'architecture par Niveau Holonique de « EMH² » ; et la méthode de particularisation des niveaux holoniques.

2. Modélisation d'une forme générique d'évènements « SurfEvent »

Dans EMH², chaque événement, qui traduit un changement d'état ou de variable horodaté dans un système, est modélisé et géré à l'aide d'un modèle unique et générique. De ce fait, avec cette approche et contrairement à l'approche originelle EMH, des règles expertes peuvent être conçues indépendamment du domaine applicatif de surveillance et des objectifs de performances de traitement de diagnostic en temps réel, contribuant ainsi à prendre en compte la spécification liée à la modularité précédemment présentée. Ce modèle générique d'un événement s'appelle un "SurfEvent". Il est considéré dans notre travaux à la fois comme une approche générique de représentations hiérarchique des évènements des systèmes mobiles et comme une langage d'interfaçage commun pour l'échanges et la coopération entre les Holons des différents niveaux Holoniques. Un SurfEvent est modélisé par un ensemble de paramètres qui sont:

- Une identification unique (son nom);
- Une association à un type de données parmi deux possibles: il peut être soit quantitatif (par exemple, le temps moyen pour le pantographe d'un train de se connecter ou la durée moyenne d'un cycle d'ouverture d'une porte dans un train) soit qualitatif (par exemple, l'état de santé d'un « système porte » décrit comme « normal », « dégradé », « critique », etc.);
- Un identifiant de niveau qui peut être modifié en fonction de la couche d'abstraction à laquelle il correspond. Ces niveaux d'abstraction d'un SurfEvent sont structurés de bas niveau à haut niveau et sont appelés respectivement Signal, Indicateur, Alarme, Signalement et Haut-Niveau. Ils sont inspirés de la norme ISO 13374 (voir la Figure 3. 4) en associant la couche #1 de la norme ISO 13374 avec le niveau « Signal », la couche #2 avec le niveau « Indicateur », la couche #3 avec le niveau « Alarme», la couche #4 avec le niveau « Signalement» et les couche #5 et #6 de la norme ISO 13374 avec le niveau « Haut-Niveau».
- Un statut parmi deux possibles: Le premier est appelé "test et développement", lorsqu'il est utilisé par les utilisateurs au cours d'un processus d'apprentissage pour un Holon spécifique (décrit ultérieurement), et le second est appelé "production", lorsqu'il est généré par un système mobile en cours d'utilisation;
- Une origine parmi deux possibles: Soit il est "calculé" (c'est-à-dire déduit et généré à partir d'autres événements) soit il est "original" (directement obtenu à partir d'un capteur).
- Une référence unique à la source qui est son "émetteur" : un émetteur est associé à la structure hiérarchique de chaque système mobile. Il dispose de son propre contexte (décrit ultérieurement). ;

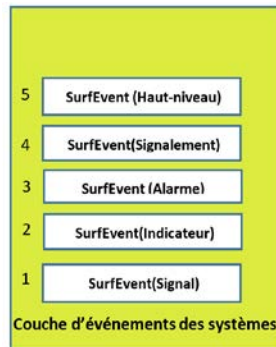


Figure 3. 4: Description des couches d'abstraction de SurfEvent (inspiré de la norme ISO 13374)

En supposant que ces paramètres constituent un ensemble d'objets soumis à des règles de fonctionnement (par exemple un évènement dispose toujours d'un émetteur quelle que soit sa source de génération), il est possible, en appliquant le modèle d'Entité Association (EA) [107](voir annexe A) qui permet de minimiser les redondances et anomalies entre les relations, d'obtenir le modèle générique d'un SurfEvent présenté Figure 3. 5). Ce modèle prend en compte les différents types, le statut et les différents emplacements (Emetteur) de SurfEvents et leurs contextes associés.

Dans nos travaux, ce modèle générique d'un SurfEvent est considéré comme un moyen de formaliser les connaissances des Holons (décrit sous la forme d'une base de connaissances et de retours d'expérience). Comme indiqué auparavant, cette approche permet aux processus de surveillance d'être indépendants des applications (systèmes cibles).

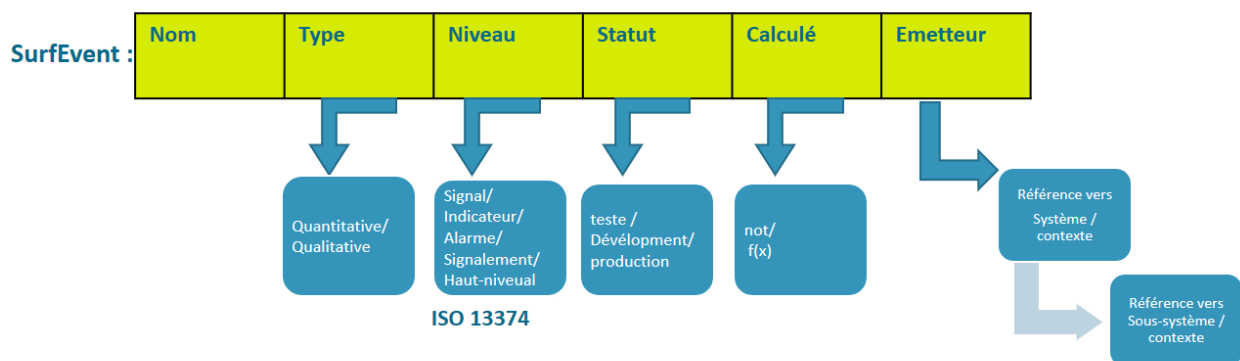


Figure 3. 5: Modèle générique d'un SurfEvent

La création, la gestion et la destruction d'un SurfEvent s'appelle "le cycle de vie d'un SurfEvent". Tout au long d'un cycle de vie, certains des paramètres d'un SurfEvent peuvent être modifiés. Son niveau, par exemple, peut évoluer au fur et à mesure que le SurfEvent traverse toute l'architecture EMH², depuis un capteur de bas niveau jusqu'à l'opérateur de maintenance. Cette approche originale facilite le suivi des SurfEvents et des activités pertinentes depuis leur entrée dans l'EMH² jusqu'à leur départ ou leur destruction. Le modèle d'un tel cycle de vie est proposé dans la Figure 3. 6 au travers de l'utilisation d'un formalisme de diagramme d'état.

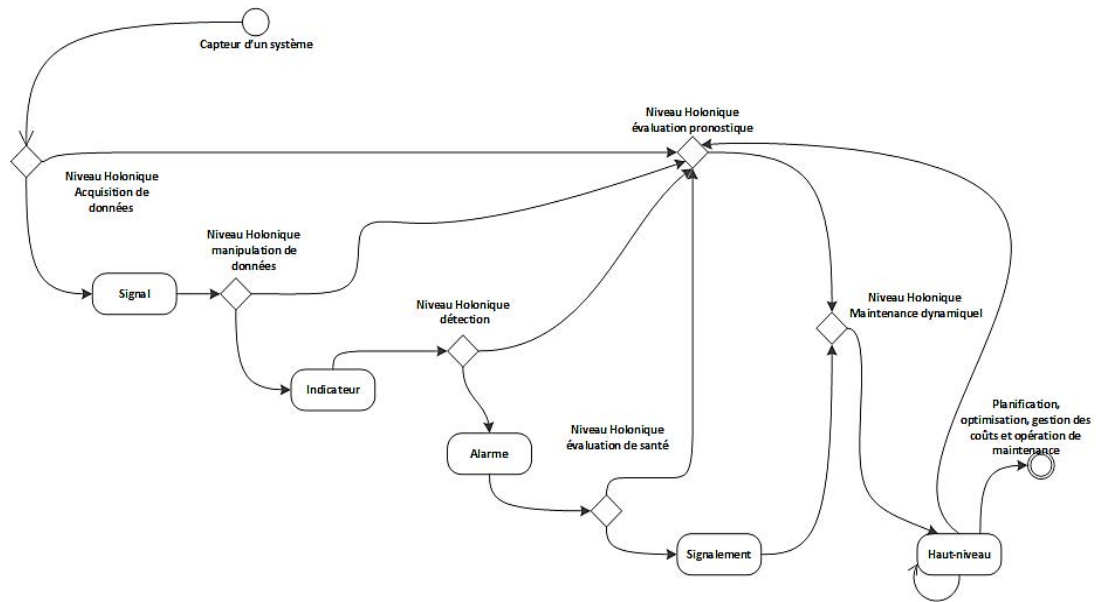


Figure 3. 6: Modèle d'un cycle de vie SurfEvent

Une des spécifications proposées requiert que l'architecture conçue soit interopérable avec les systèmes d'information de l'entreprise pour assurer une intégration efficace dans son organisation[77]. Le modèle générique de SurfEvent facilite la prise en compte de cette spécification. L'interopérabilité est définie comme « La capacité pour deux systèmes (ou plus) à communiquer, coopérer et échanger des données et services, et ce, malgré les différences dans les langages, les implémentations et les environnements d'exécution ou les modèles d'abstraction »[108]. D'après EIF(European Interoperability Framework)[109], l'interopérabilité d'un système doit répondre aux trois niveaux d'interopérabilité : Le niveau sémantique« savoir se comprendre »; Le niveau technique « pouvoir communiquer » et le niveau syntaxique « savoir communiquer ».

Plusieurs normes d'interopérabilité ouvertes, standard communautaires et propriétaires existent. Elles valident et aident les organismes à rendre leurs services et produits interopérables[109]. Dans le cadre de nos travaux, nous nous focalisons sur les normes liées aux formats d'échange de données dans le contexte du système d'information, par exemple, la normes ISO pour les TIC (Technologie d'Informations et de la Télécommunication).

Dans EMH², l'interopérabilité est permise au travers de la définition d'un convertisseur d'évènements appelée "SurfConvert" qui intègre les trois niveaux d'interopérabilité sémantique, technique et syntaxique des SurfEvents comme suit :

- Concernant le niveau d'interopérabilité sémantique, le convertisseur SurfConvert dispose d'un descripteur de méta données capable d'associer le modèle générique d'un SurfEvent et son cycle de vie avec une représentation universelle et standard de données (voir la Figure 3. 7). Cette représentation standard de données peut être de type XML (Extensible Markup Language), JSON (JavaScript Object Notation), YML (Yet Another Markup Language), CSV (Comma Separated Values), TRDP (Train Real Time Data Protocol), CAN (Controller Area Network), MVB (Multifunction Vehicle Bus), etc.
- Concernant le niveau d'interopérabilité technique, le convertisseur SurfConvert véhicule la représentation standard de données définie au niveau de l'interopérabilité sémantique au travers de différents interfaçages universels de communication tels que REST

(Representational State Transfer), MQTT (Message Queuing Telemetry Transport), OPC (Open Platform Communications), etc.

- Concernant le niveau d'interopérabilité syntaxique, le convertisseur SurfConvert dispose de capacité de renseignements sur le format de codage textuel des évènements, de cryptage, etc.

Le convertisseur d'évènements SurfConvert est alors intégré dans chaque niveau Holonique. Comme indiqué, il permet de mettre en œuvre les liaisons avec des systèmes externes, notamment ceux qui gèrent l'infrastructure, et avec les activités liées au cycle de vie des systèmes cibles, notamment leurs phases de design, développement, production et retrait. Ainsi lorsque l'architecture EMH² reçoit ou échange un évènement, son type de format et ses champs de représentation des données (méta données) doivent être au préalable renseignés et le convertisseur se base sur cette description pour convertir les données en SurfEvent et vice versa.

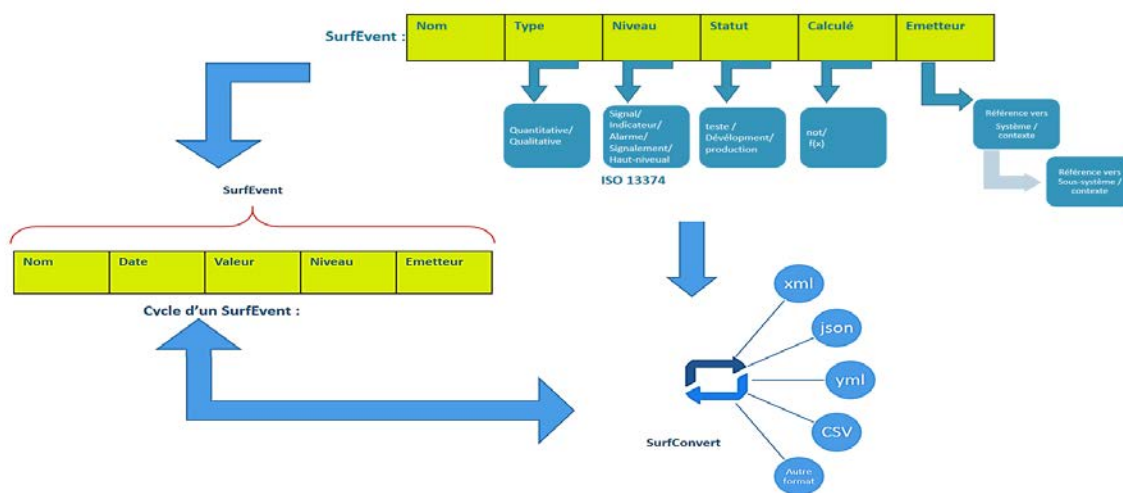


Figure 3. 7: Convertisseur d'évènements "SurfConvert"

Dans la sous-section suivante, nous allons présenter le deuxième élément clé de notre contribution, qui est « L'architecture par niveaux Holoniques de EMH² ».

3. L'architecture par Niveaux Holoniques de « EMH² »

L'architecture Holonique « EMH² » est composée d'un ensemble des niveaux Holoniques. Comme défini ci-dessus dans la section qui présente les principes fondateurs, un niveau Holonique fait référence à un groupe de holons appartenant à la même famille. Ce groupe hérite ainsi de la mission de surveillance de la couche qui le représente. Cette approche offre des avantages en terme de facilité de déploiement automatisé notamment. Au sein d'EMH², chaque niveau Holonique gère la mission de surveillance des systèmes et sous-systèmes à travers un ensemble de Holons. Un niveau holonique est défini par l'ensemble des éléments suivants:

- Un ensemble d'holons regroupés par famille d'équipements composant le système cible. Ces holons coopèrent dans un premier temps au niveau des sous-systèmes et des systèmes par famille d'équipements (par exemple, dans un Train, le système porte dispose de plusieurs émetteur, deux par véhicule), puis dans un second temps, ils coopèrent avec les autres systèmes hétérogènes et holons de surveillance correspondants. Ce type d'analyse

prioritairement homogène, facilite le suivi, la discrimination et l'identification de comportements suspects. Cela accroît par conséquent la robustesse des décisions de maintenance prévisionnelle ou proactive.

- Un module d'interface d'élaboration et de recommandation de conseils ainsi qu'un module d'interface interne et externe grâce à l'approche « SurfConvert ».
- Un système de gestion de flux (workflow)[110] qui gère la provenance et le cycle de vue des SurfEvents du niveau (provenant de capteurs et / ou d'autres niveaux Holoniques). Lorsqu'un SurfEvent arrive, le système identifie son origine (le système émetteur) et déclenche les holons correspondants chargés de sa prise en charge.

La Figure 3. 8 suivant schématise globalement l'articulation de tous les niveaux holoniques dans EMH² et propose un focus sur un niveau en particulier.

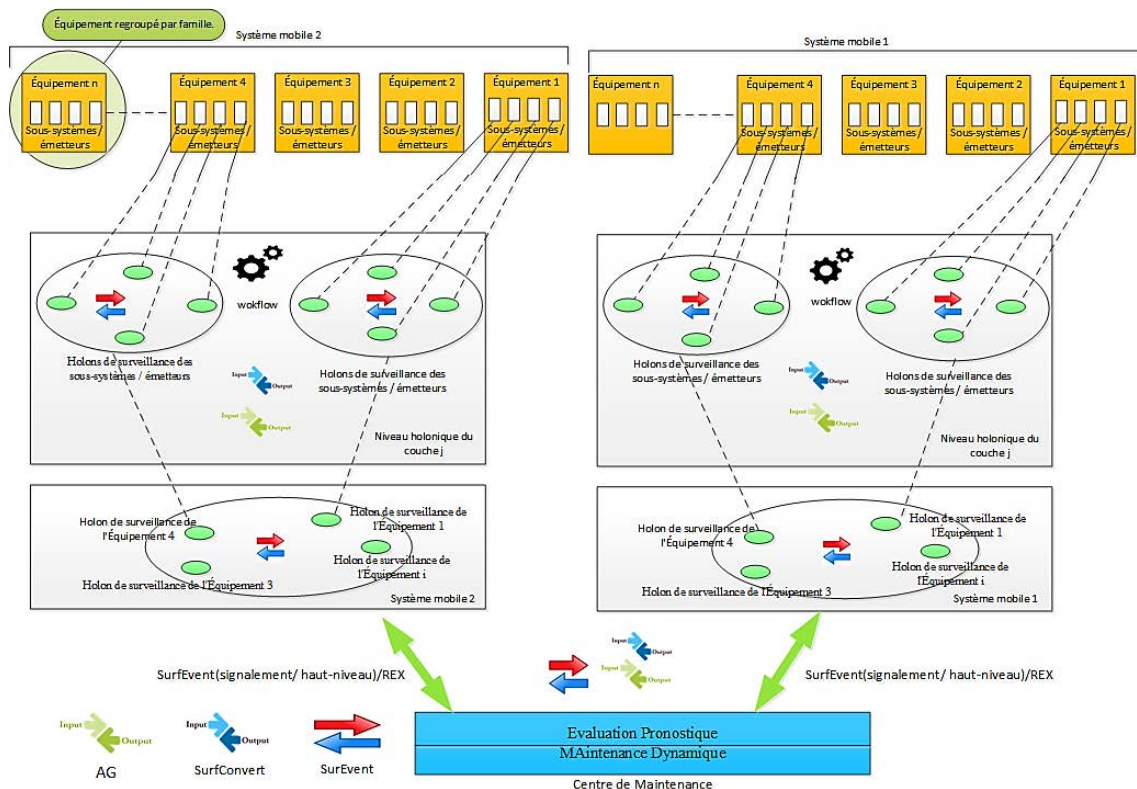


Figure 3. 8: Un niveau Holonique dans l'EMH²

Les différents niveaux Holoniques de EMH² sont détaillés ci-après. Les termes et missions associés aux différents niveaux Holoniques sont inspirés de la norme ISO 13374 :

- **Acquisition de données** : ce niveau Holonique permet de convertir (via le SurfConvert) un signal délivré par un capteur ou un transducteur en un SurfEvent de type « signal » représentant une donnée de type « quantitative » ou « qualitative », auquel est associé différentes informations nécessaires à son instanciation présentées dans la partie précédente (datation, source émettrice du signal, par exemple). Le résultat d'acquisition est transmis aux niveaux Holoniques supérieurs et/ou aux autres applications hétérogènes de l'entreprise avec lesquelles le niveau interopère.
- **Manipulation de données** : ce niveau Holonique analyse des SurfEvents de type signal, effectue un calcul de descripteurs significatifs (un descripteur représente différents

opérations ou fonctions intervenant sur les champs des cycles des SurfEvents, par exemple, les opérations d'agrégation sur les occurrences des événements durant un intervalle de temps), et réalise des traitements et des transformations de ces SurfEvents en SurfEvents de type « indicateur ». Les résultats de cette manipulation sont transmis aux niveaux Holoniques supérieurs et/ou aux applications hétérogènes de l'entreprise avec lesquelles le niveau interopère.

- **Détection** : ce niveau Holonique a pour mission de rechercher et d'identifier des anomalies portées par les SurfEvents de type « indicateur » et déterminer la zone d'anomalie correspondante (si elle existe) concernée par ces SurfEvents, dans le but de générer des SurfEvents de type « alarme ». Les résultats de cette détection sont transmis aux niveaux Holoniques supérieurs et/ou aux applications hétérogènes de l'entreprise avec lesquelles le niveau interopère.
- **Evaluation de Santé** : ce niveau Holonique a pour mission de réaliser le diagnostic des fautes détectées sur les SurfEvents de type « alarme » et déterminer l'état de santé courant de l'équipement ou du processus, en générant des SurfEvents de type « signalement » qui regroupent l'ensemble des informations d'états. Les résultats de cette évaluation de santé sont transmis aux niveaux Holoniques supérieurs et/ou des applications hétérogènes de l'entreprise avec lesquelles le niveau interopère.
- **Evaluation de Pronostic** : ce niveau Holonique a pour mission de déterminer les états de santé et les modes possibles de défaillances futurs à partir de l'évaluation de SurfEvents de type « signalement », « signal », « indicateur » et « alarme » et de la projection de l'utilisation de l'équipement ou du processus. Il génère des SurfEvents de type « Haut-niveau » contenant les prédictions de durée de vie résiduelle (RUL). Les résultats de cette évaluation de Pronostic sont transmis aux niveaux Holoniques « Maintenance Dynamique » et « Evaluation de santé » et/ou aux applications hétérogènes de l'entreprise avec lesquelles le niveau interopère.
- **Maintenance Dynamique** : ce niveau Holonique remplace la couche de génération de conseils de la norme ISO 13374. Il sera détaillé dans la section présentant la méthode d'adaptation de « EMH² » aux contextes de la flotte. Il s'appuie sur les SurfEvents de type « signalement » (de l'état de santé actuel) et « Haut-niveau » (état de santé prévisionnel) ainsi que sur les différents contextes opérationnels des systèmes mobiles de la flotte. Sa mission est de fournir une surveillance dynamique fondée sur l'optimisation et la gestion du coût de la maintenance de la flotte. Les résultats de ce niveau Holonique sont transmis aux niveaux Holoniques « Evaluation Pronostic », « Evaluation de santé » et/ou aux applications hétérogènes de l'entreprise avec lesquelles le niveau interopère.

La Figure 3. 9 formalise une vue de ces différents niveaux Holoniques.

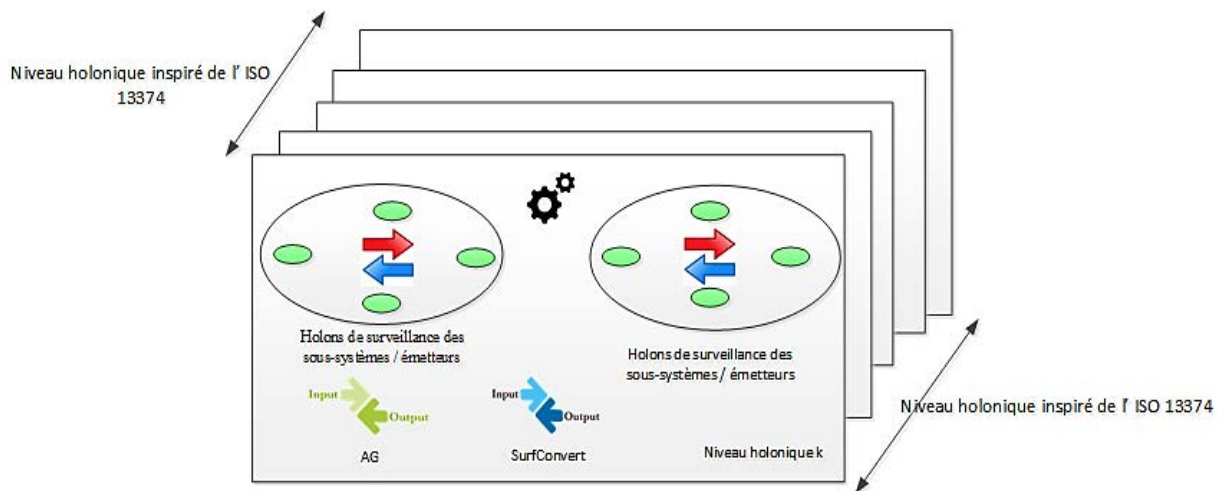


Figure 3. 9: Les niveaux Holoniques dans l'EMH²

4. La méthode de particularisation

La partie précédente ne fait que décrire globalement l'architecture holonique, c'est-à-dire la structure par laquelle les holons vont coopérer. Cependant, il reste à déterminer comment s'assurer que les Holons de différents niveaux Holoniques coopèrent et surveillent de manière modulaire, réactive et adaptative tous les systèmes, du plus bas niveau (par exemple, niveau équipement) au plus haut niveau (la flotte de systèmes mobiles) sachant qu'il existe des contextes d'utilisation différents pour les niveaux holoniques.

Pour ce faire, dans cette partie, nous allons particulariser les différents niveaux Holonique de l'EMH² afin de pouvoir les adapter aux différents contextes rencontrés lors de l'exploitation de la flotte de systèmes mobiles. L'objectif de ces particularisations est d'obtenir une surveillance plus modulaire, réactive et adaptative tout en gardant en ligne de mire l'optimisation global du coût de la maintenance de la flotte. Nous les présentons succinctement ci-dessous avant de les détailler par la suite:

- **Pour les niveaux Holoniques « Acquisition de données », « Manipulation de données », « Détection » et « Evaluation de Santé »**, la particularisation consiste à rendre la surveillance des Holons plus modulaire, réactive et adaptative avec les Holons des niveaux Holoniques de même niveau et des niveaux supérieurs au travers de la définition d'une fonction de traitement spécifiques des SurfEvents appelée "SurfProcessing".
- **Pour le niveau Holonique « Evaluation de Pronostic »**, la particularisation consiste à intégrer aux holons de ce niveau des mécanismes supportant l'intégration et la génération de nouvelles connaissances concernant le niveau de la flotte;
- **Pour le niveau Holonique Maintenance Dynamique**, la particularisation consiste à supprimer la couche de génération de conseils de la norme ISO 13374 (comme indiqué auparavant, cette couche de génération de conseils est en fait désormais dupliquée pour chaque niveau Holonique afin de leur permettre de générer leurs propres conseils et recommandations) et à la remplacer par une nouvelle couche appelée « maintenance dynamique ». L'objectif de cette particularisation est de permettre l'intégration de mécanisme en lien avec la gestion, la planification dynamique et l'optimisation des coûts de la maintenance d'une flotte de systèmes mobiles ;

Dans les sous-sections suivantes, nous allons détailler ces trois différentes particularisations.

4.1. Particularisation des niveaux Holoniques « Acquisition de données », « Manipulation de données », « Détection » et « Evaluation de Santé »

L'adaptation des niveaux Holoniques « Acquisition de données », « Manipulation de données », « Détection » et « Evaluation de Santé » consiste à rendre la surveillance de leurs Holons plus modulaire vis à vis de différents systèmes et composants et plus réactive en localisant, raisonnant et s'assurant que les décisions prises dans un contexte réparti soient prises de manière suffisamment coopérative avec les Holons des couches supérieures, tout en intégrant dynamiquement les connaissances et informations générées par les Holons de même niveau et des niveaux Holoniques « Evaluation de Pronostic » et « Maintenance Dynamique ».

Ce besoin en particularisation provient du fait que les holons de ces niveaux sont conçus pour permettre l'intégration du grand nombre de modèles et méthodes de diagnostic de systèmes existants (prêts à l'emploi, « sur étagère ») sous deux conditions 1) ils doivent discriminer de manière générique des comportements normaux et anormaux[111] et 2) ils peuvent être adaptés pour supporter le modèle de « SurfEvent ». Il est ainsi très important de noter que l'architecture EMH² est ainsi vue comme un intégrateur de modèles de diagnostic existants : ceci nous conforte bien dans le fait que notre contribution se situe au niveau architectural et événementiel, et non au niveau des modèles et méthodes de diagnostic eux-mêmes.

Comme évoqué dans la partie (2.2) du chapitre I, du point de vue de la modélisation, différentes méthodes de diagnostic existent. Elles sont classifiées en méthode basées sur des modèles, sur des règles (base de connaissances) et sur les historiques des événements. A ce niveau de la conception de notre proposition, nous devons choisir une méthode de diagnostic pour avancer dans le développement (il est possible d'envisager en perspective l'usage d'autres méthodes). Dans cette thèse, notre choix s'est porté sur les méthodes basées sur la connaissance, et plus particulièrement, les systèmes experts, car ils offrent une capacité de raisonnement dans l'incertitude et de capacité d'explication et de localisation des solutions fournies[59]. La technique est également largement éprouvée et les limites de ce type d'approche sont connues depuis très longtemps. Pour plus de détails sur le choix de cette méthode, voir l'annexe B-1).

Schématiquement, un système expert est un programme informatique reproduisant le comportement cognitif d'un expert humain pour résoudre certains problèmes d'un domaine particulier[7]. Il est classiquement constitué de deux parties indépendantes [112] (voir la Figure 3.10) : Une **base de connaissance**, élaborée à partir d'une expertise humaine, elle-même composée d'une base de règles qui modélisent les causes d'une défaillance en fonction des symptômes observés (la connaissance du système sous surveillance) et d'une base de faits qui contient la liste des fautes répertoriées et qui peut être actualisée; Un **moteur d'inférence** capable de raisonner à partir de la base de connaissance et d'émettre des conclusions à partir de faits. Ce moteur d'inférence raisonne selon différentes stratégies : chaînage avant, chaînage arrière ou chaînage mixte.

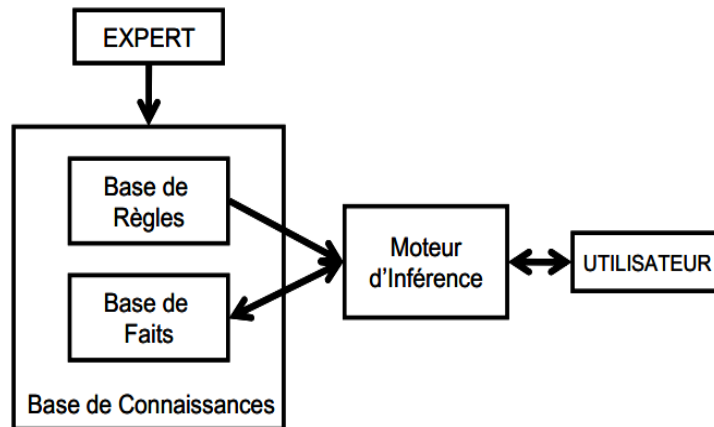


Figure 3. 10 : Architecture d'un system Expert[112]

Plusieurs systèmes experts dédiés au diagnostic ont vu le jour, par exemple CADUCEUS, MYCIN, PUFF, Mistral, Eydenet et Kaleidos[113]–[115]. Ils sont souvent dédiés à un domaine spécifique (applicatif). Ils sont très souvent conçus de manière centralisé, monolithique dans la mesure où peu d'entre eux considèrent la possibilité de coopérer avec d'autres systèmes experts dans un contexte décisionnel tel que celui qui nous concerne.

Nous proposons ainsi de développer un système expert dédié au diagnostic d'une flotte de systèmes mobiles sous la forme d'un modèle mettant en œuvre les fonctions de surveillance des Holons, afin de leur donner la capacité de localiser, raisonner et de prendre des décisions et de coopérer dans un environnement réparti. Chaque holon dispose ainsi de son système expert. Chaque système expert dispose, outre d'une stratégie classique de chaînage avant, d'une stratégie de raisonnement de chaînage arrière afin de disposer d'un mécanisme d'exploitation et d'explication guidé par les faits jusqu'à l'objectif tout en permettant une coopération avec les Niveaux Holoniques supérieurs.

Cette fonction de surveillance particularisée et spécifique à ces Holons est appelée "SurfProcessing". Elle est mise en œuvre au travers du système expert présenté. Celui-ci est logiquement construit sur le modèle de « SurfEvent » qui comme indiqué, permet de supprimer la dépendance des fonctions de surveillance vis-à-vis des systèmes applicatifs cibles et permet de répondre aux exigences des systèmes embarqués en terme de gestion de ressources critiques (limitant les capacités de traitement, de communication et de mémorisation). Cette approche offre enfin la possibilité de raisonner sous incertitude et d'expliquer les solutions données aux opérateurs de maintenance en utilisant des processus de backtracking via le chaînage arrière.

Chacun de ces systèmes experts est classiquement composé d'une base de connaissances et d'un moteur d'inférence:

- **La base de connaissances** proposée est basée sur une grammaire sans contexte[116]. Le choix et la définition de cette grammaire a fait l'objet d'une étude comparative décrite dans l'annexe C. Une grammaire de type 2 a été sélectionnée pour nous permettre de générer les expressions mathématiques, logiques, textuelles et temporelles associées au champ « calculé » précisant l'origine d'un SurfEvent. Etant donné que pour chaque SurfEvent, son statut peut être soit en " test et développement " (à l'aide d'un simulateur, par exemple) soit en "mode production" (validé et en cours d'utilisation), voir Figure 3. 4 , les connaissances en question relèvent de l'un ou de l'autre de ces statuts

- **Le moteur d'inférence** proposé est basé sur un automate à pile[116] pour permettre une reconnaissance grammaticale sans contexte. Le choix et la définition de cet automate a fait l'objet d'une étude comparative présentée en annexe C.

Comme indiqué, le moteur d'inférence utilise le chaînage arrière. Son mode de raisonnement peut être de deux types : soit il est basé sur cycle soit sur état:

- **Mode de raisonnement basé sur cycle:** un cycle est défini par sa condition de départ et un état final. Chaque condition est modélisée sous la forme de SurfEvents. Un cycle dépend fortement des caractéristiques physiques du système cible, cependant il présente les caractéristiques génériques suivantes:
 - Tous les SurfEvents dans un cycle sont uniques;
 - Un cycle a une date de début et une date de fin;
 - La durée d'un cycle est variable.

Par exemple, quand une porte de train s'ouvre, un cycle d'ouverture de porte et un cycle de fermeture de porte sont distingués. Chaque cycle a ses propres conditions de début et de fin. Le principal avantage du raisonnement par cycle est la réactivité car si l'on attend l'arrivée de tous les SurfEvents dans le cycle avant de lancer le processus de raisonnement du système expert, un temps précieux peut être perdu. Une description formelle de ce mode est la suivante:

$$cycle(s) \leftrightarrow \exists e_1, e_2 \in s, I(e_1, t) \neq I(e_2, t)$$

$$\forall SurfEvent \in e_1, e_2 SurfEvent_i \neq SurfEvent_j$$

où « s » est un système composé d'un ensemble de sous-systèmes et de composants; « e_1 et e_2 » sont des ensembles d'événements représentant respectivement les conditions de début et de fin d'un cycle (sa condition départ et son état final) de telle sorte que chaque SurfEvent d'un ensemble « e_1 et e_2 » soit unique à instant t .

- **Mode de raisonnement basé sur état:** dans ce mode, les processus de raisonnement du système expert commencent lorsqu'une modification de la valeur d'un SurfEvent est observée. Par défaut, les dernières valeurs observées sont conservées. L'avantage de ce mode est qu'aucune connaissance du système à surveiller n'est nécessaire. Cependant, une surveillance continue et un dispositif de mémoire sont obligatoires. Un exemple d'un équipement physique concerné par ce mode est typiquement une lampe (allumée / éteinte). Une description formelle de ce raisonnement est la suivante:

$$état(s, t) \leftrightarrow \forall SurfEvent \in s, (SurfEvent_t = SurfEvent_{t-1})$$

$$\vee (SurfEvent_t \neq SurfEvent_{t-1}), t \in [0, \infty[$$

où « s » est un système composé d'un ensemble de sous-systèmes et de composants; "SurfEvent_t" correspond au changement d'état d'un SurfEvent à un instant donné t

Le chaînage arrière permet au holon de surveillance d'avoir la possibilité de remonter en arrières la chaîne des faits jusqu'à l'objectif. Il est utilisé comme suit, pour les deux modes de raisonnement introduits (par cycle ou sur état) :

- Quand un SurfEvent est observé par un holon dans une des couches « acquisition de données », « manipulation de données », « détection » et « évaluation de santé », le système expert du holon le considère comme un fait. Toutes les règles de la base de connaissances où ce SurfEvent est identifié sont sélectionnées;
- L'ordre d'exécution des règles sélectionnées est établi en fonction du niveau d'abstraction (leur appartenance à un niveau Holonique) : le niveau signal est de la plus haute priorité, puis le niveau de l'indicateur, et ainsi de suite;
- Pour un niveau d'abstraction donné, un ordre d'exécution des règles est établi en commençant par des règles simples et en terminant par des règles complexes. Une règle est dite « simple » lorsque son évaluation ne nécessite aucune coopération ni échange avec d'autres Holons ni aucun système hétérogène, sinon elle est appelée « complexe ».
- Une règle est exécutée en mode cycle ou état;
- Le résultat d'évaluation sert de fait pour les autres règles, et ainsi de suite;
- Si le système expert d'un Holon n'arrive pas expliquer ou évaluer un fait, son moteur d'inférence demande aux Holons du niveau « Evaluation Pronostic » une explication ou une prédiction sur l'évènement (ce mécanisme sera détaillé dans la sous-section suivant).

Le Figure 3. 11 suivante transcrit sous forme algorithmique le raisonnement de type chaînage arrière du moteur d'inférence des Holon .

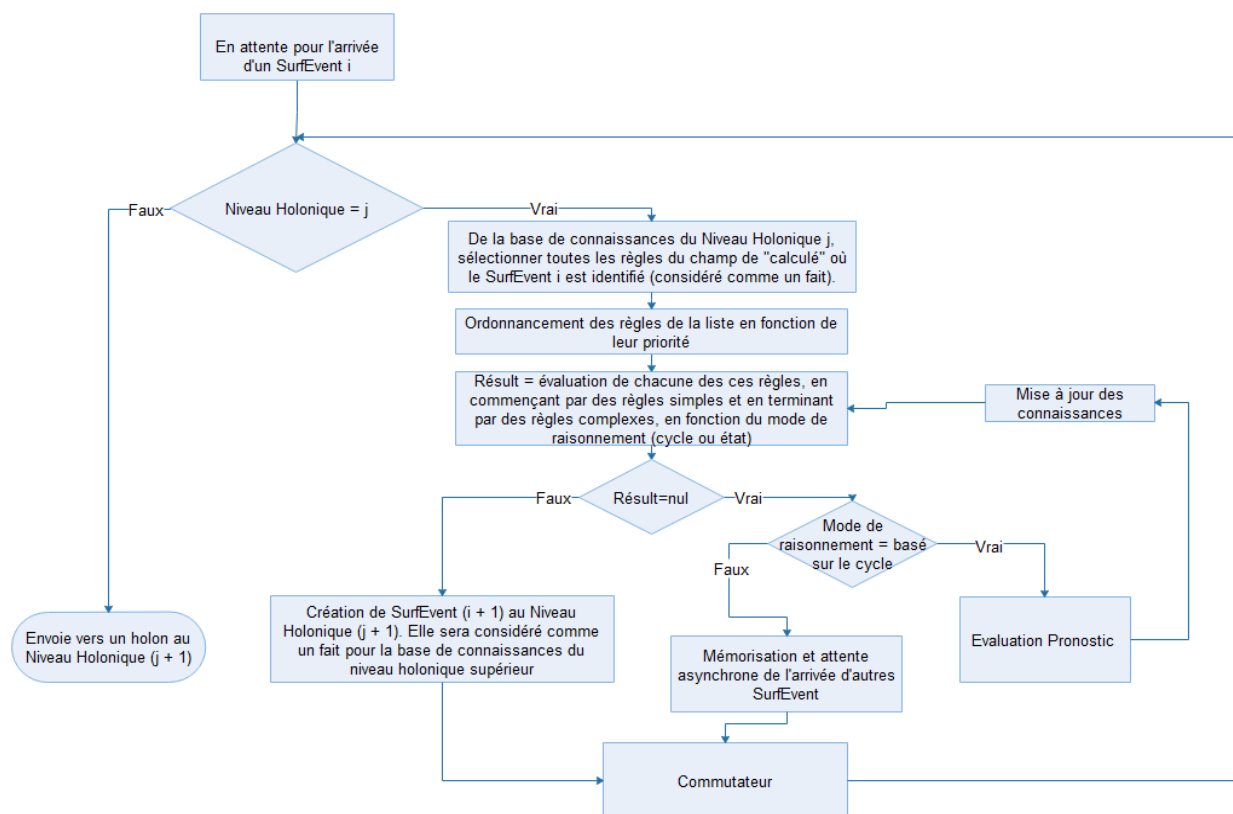


Figure 3. 11: Description algorithmique du chaînage arrière du système expert d'un holon.

4.2. Particularisation du Niveau Holonique « Evaluation de Pronostic »

La particularisation des holons de ce niveau consiste à spécifier leurs localisations, soit dans les systèmes mobiles soit dans le CM au sol en fonction de la globalité du pronostic. Les Holons concernés vont soit implémenter un mode de « **pronostic local** » s'ils sont embarqués dans les systèmes mobiles soit de « **pronostic global** » s'ils sont « débarqués » au sol :

- **Pronostic local**

Le pronostic local sert à classifier l'influence entre les SurfEvents, prédire de nouveaux SurfEvents et expliquer des situations qualitatives ou quantitatives, au sein d'un même système mobile. Ses résultats de recommandations et conseils seront transmis aux Holons du Niveau Holonique « Evaluation de Santé » et au sol vers l'Holon pronostic qui est en charge du « pronostic global ».

Comme évoqué dans la partie (2.2) du chapitre I, du point de vue de la modélisation, différentes méthodes de pronostic existent. Elles sont classifiées en méthodes basées sur des modèles, basées sur des règles (base de connaissances) et basée sur les historiques des événements. Notre choix s'est porté sur une méthode basée sur les historiques des événements, qui est celle des arbres de décision [117]. Elle a en effet l'avantage de ne pas nécessiter de connaissance préalable pour caractériser l'accumulation et la progression des défaillances de systèmes sous surveillance et elle offre également la capacité de classifier les influences entre les événements, de prédire et d'expliquer des événements quantitatifs et qualitatifs et ce, à partir des événements dont la diversité n'est pas contraignante. Elle présente enfin la capacité de générer directement des règles de prédictions à partir de ses « feuilles » (nœuds terminaux de l'arbre de décision). Ces règles peuvent exprimer les risques de défaillance future des systèmes et les durées de vie utile restante (RUL). Pour plus de détail sur le choix de cette méthode, voir l'annexe B-1. Un arbre de décision est un outil d'aide à la décision construit avec des algorithmes d'apprentissage supervisé tels que l'algorithme CART (Classification and Regression Tree)[118], C4.5[119] et C5.0[120]. Il représente un ensemble de choix sous la forme graphique d'un arbre (constitué d'une racine et des feuilles). Le chemin reliant une feuille à la racine de l'arbre peut être lu comme une règle de prédiction du type attribut-valeur : « Si prémisse « faits » ... alors Conclusion « objectif » ... ». Les algorithmes d'apprentissage C4.5 et C5.0 offrent d'excellentes performances pour la construction d'un arbre de décision.

L'algorithme construit un arbre de décision à partir d'un ensemble de données d'apprentissage « **H** ». Son objectif est de classer ces données en catégories en fonction de leurs critères de séparation du point de vue de la variable à prédire, et cela en utilisant le concept d'entropie de l'information (voir la Formule 3. 7). Dans la première étape, l'algorithme calcule l'entropie sur l'ensemble de données d'apprentissage « **H** » afin de choisir la variable qui sépare de manière la plus significative cet ensemble. Puis, le résultat de séparation de cet l'ensemble de données « **H** » est affecté à différentes feuilles, qui représentent les différents sous-ensembles de « **H** ». L'algorithme va continuer ainsi à segmenter ces feuilles afin de minimiser l'entropie et limiter le risque de sur-apprentissage. Une fois l'arbre terminé, l'algorithme affecte à chaque feuille la classe la plus représentée et les résultats peuvent alors être représentés visuellement, sous forme d'un arbre pour lequel le chemin reliant une feuille à la racine de l'arbre est lu comme une règle de Prédiction. L'Algorithme 3. 1 décrit cette méthode appliquée pour un holon embarqué qui implémente un « pronostic local » de notre architecture. Il est construit sur le concept de SurfEvent où l'ensemble « **H** » représente ici les données historiques des SurfEvents d'un système. A chaque nœud de l'arbre,

le C4.5 choisit le SurfEvent qui divise le plus efficacement son ensemble en sous-ensembles enrichis dans une classe ou dans l'autre. Le critère de division est le gain d'information normalisé (différence d'entropie). Le SurfEvent avec le gain d'information normalisé le plus élevé est choisi pour prendre la décision. L'algorithme C4.5 est ensuite appliqué de manière récurrente sur les plus petites sous-listes.

Arbres (H)
Début
Si tous les **H** appartiennent à une catégorie alors
 retourner une feuille avec cette catégorie comme étiquette
Sinon si SurfEvents est vide alors
 retourner une feuille dont l'étiquette est la catégorie majoritaire dans **H**
Sinon choisir un SurfEvent « **Surf** » depuis **H** pour la racine:
 pour toutes les valeurs possibles v_i de **Surf**
 soit H_i le sous-ensemble de **H** dont la valeur de **Surf** est v_i
 ajouter une branche à la racine pour le teste **Surf** = v_i
 si H_i est vide alors
 créer une feuille dont l'étiquette est la catégorie majoritaire dans **H**
 sinon appeler récursivement **Arbres** ($H_i - \{ \text{Surf} \}$)
Fin

Algorithme 3. 1: Algorithme de l'arbre de décision

Le calcul du choix de la racine est effectué avec la formule de l'entropie suivant, qui est une mesure du désordre d'un ensemble de « **H** » par rapport à une classification.

$$Entropie(H) = \sum_{i=1}^c - p_i \log_2 p_i \quad \text{Formule 3. 7}$$

Où p_i est la proportion de SurfEvents de catégorie i dans **H** et c est le nombre d'occurrences de SurfEvents dans **H**. Ainsi, le gain d'un SurfEvent « **Surf** » est calculé avec la Formule 3. 8 suivante :

$$Gain(H, surf) = Entropie(H) - \sum_{v \in \text{valeurs}(surf)} \frac{|H_v|}{|H|} Entropie(H_v) \quad \text{Formule 3. 8}$$

avec Valeur(surf) est l'ensemble des valeurs possibles de l'attribut **Surf**, H_v est le sous ensemble de **H** pour lequel **Surf** a valeur v , $|H|$ est la taille de **H**, $|H_v|$ est la taille de H_v

Les résultats de l'arbre de décision en terme de classification, d'influence, de prédiction ou d'explication de chaque SurfEvents au sein d'un système mobile sont alors transmis au sol vers l'Holon pronostic qui implémente un pronostic de type « pronostic global ». Ce mode est détaillé dans la sous-section suivante.

- **Pronostic global**

Un holon débarqué implémentant un « pronostic global » reçoit en entrée les résultats des études de classification, d'influence, prédiction ou d'explication de différents Holons « pronostic local ». Il reçoit également un retour d'expériences (REX) de la part des Holons du Niveau Holonique « maintenance dynamique ». Il établit ensuite une classification plus générale des SurfEvents pour

tous les systèmes mobiles de la flotte. Ses résultats d'évaluations servent à enrichir la base de connaissances des Holons de Niveau Holonique inférieurs.

Les échanges et les coopérations entre les holons de pronostic global et les holons de pronostic local sont modélisés dans la Figure 3. 12 .en utilisant un formalisme d'automate à états finis. Il faut noter que les résultats des Holons « pronostic local » sont considérés dans nos travaux comme des règles de classification, de prédiction ou d'explication de nature « peu robuste » dans le sens où ils dépendent directement des contextes de fonctionnement physique et des degrés de vieillissement des équipements émetteurs. Ainsi, Ces Holons « pronostic local » peuvent émettre des résultats d'études de classification et de diagnostic erroné lorsque le sous-système auquel ils sont associés est influencé par d'autres sous-systèmes. De ce fait, une étape plus générale « pronostic global » est nécessaire afin d'unifier, de valider ces résultats et donc de présenter des classification, prédiction ou d'explication plus robustes et fiables aux opérateurs de maintenance.

Pour ce faire, un holon de « pronostic global » applique un algorithme de fusion de niveau flotte par équipement (voir l'Algorithme 3. 2). Cet algorithme calcule les règles de surveillance générale sous la forme d'une « base de connaissance » qui représente les différentes spécificités et contextes fonctionnelles des différentes études de classification, d'influence, de prédiction ou d'explication des Holons « pronostic local ». Cet algorithme fournit une comparaison détaillée de divers cas d'études observées pour un type d'équipement au niveau flotte. Cet algorithme fonctionne comme suit : supposons que nous ayons une règle de prédiction « Rég » d'un équipement « Eq » du type attribut-valeur « Si « faits » ... alors « objectif » ... », où « faits » correspond à un ensemble de SurfEvents connectés en utilisant les opérateurs « et » / « ou » et « objectif » correspond à un SurfEvent représentant l'objectif (conclusion) de la prédiction. Cette règle est transmise de la part d'un Holon de type « pronostic local » vers l'Holon de « pronostic global » situé au CM au sol. Supposons également que vous avons une base de connaissance « Base_Eq » qui représente la connaissance générique niveau flotte de l'équipement « Eq ». Cette base de connaissance peut être vide ou non. Sur la base de ces suppositions, l'algorithme de fusion utilise la longueur de chemin reliant une feuille à la racine de l'arbre « long(Rég) » afin de déterminer la comparaison détaillée de diverses cas d'études observées d'un équipement « Eq ». Il est détaillé ci-dessous (Algorithme 3. 2).

Fusion (Base_Eq ,Rég)

Début

Si (Base_Eq est vide) ou (Rég n'existe pas dans Base_Eq) alors

Insertion de la Rég dans Base_Eq ;

Sinon

Rég_existant = recherche (Base_Eq ,Rég) ;

Si (long(Rég) == long(Rég_existant)) alors

Vérification et intégration des seuils dans les conditions des prémisses de la Rég_existant
si nécessaire ;

Sinon

Rég_existant = Rég_existant « ou » Rég ;

Mise jour de la base de connaissances (Base_Eq , Rég_existant) ;

Fin

Algorithme 3. 2: Algorithme de fusion de l'Holon « pronostic global »

La Figure 3. 12, introduite précédemment, qui illustre l'échange et la coopération entre les Holons de « pronostic local » et « pronostic global » met en évidence un point important de notre contribution : seuls les résultats des études de classification, d'influence, de prédiction ou d'explication sont partagés entre les systèmes mobiles et le CM au sol. Cette situation évite ainsi la transmission de grandes quantités de données historiques des équipements vers le CM, étape délicate à mener dans une logique architecturale fortement centralisée.

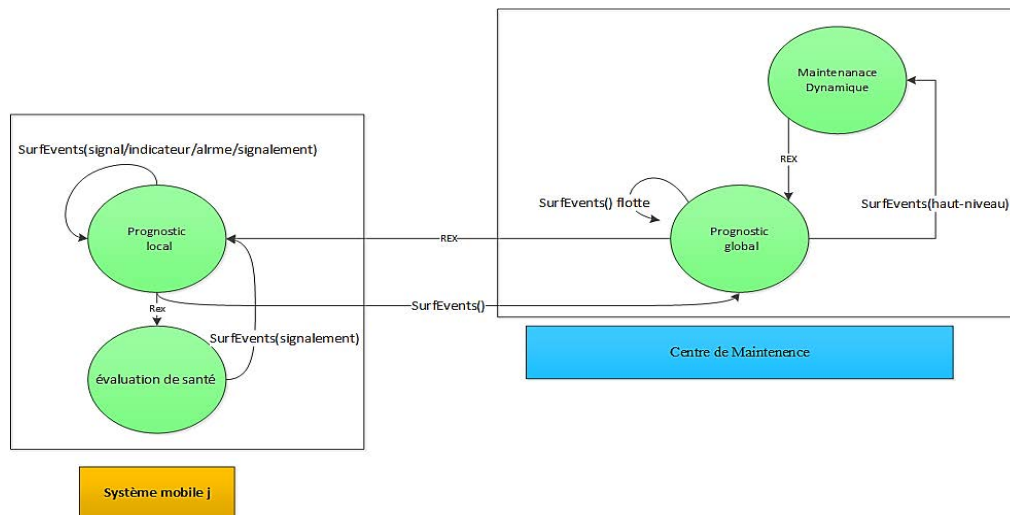


Figure 3. 12 : Echange et coopération entre Holons de « pronostic local » et « pronostic global »

À travers un tel modèle d'arbre de décision, les Holons « **pronostic local** » et « **pronostic global** » ont la capacité de jouer un rôle essentiel pour la génération de nouvelles connaissances, l'adaptation aux changements au sein des systèmes mobiles et pour le partage de connaissances entre les différentes entités de la flotte. Dans ce contexte, le chapitre suivant présentera une simulation visant d'une part à illustrer la modélisation des Holons « **pronostic local** » au sein d'une flotte de systèmes mobiles et d'autre part, estimer la durée moyenne de génération de connaissances et d'intégration des retours d'expérience sur le processus de surveillance et comparer cette durée avec une situation de référence.

Dans la sous-section suivante, nous allons présenter la particularisation du niveau Holonique « Maintenance Dynamique » et présenter les échanges entre holons de ce niveau avec ceux des Niveaux Holoniques « Evaluation Pronostic » et « Evaluation de Santé ».

4.3. Particularisation du Niveau Holonique « Maintenance Dynamique »

Afin d'élaborer des conseils issus des fonctions de diagnostic et pronostic d'un système mobile immergé au sein d'une flotte, la stratégie ayant pour objet d'optimiser la disponibilité des équipements roulants et de réduire l'impact de la maintenance sur les coûts d'exploitation à partir de l'état de la flotte seule ne suffit pas. Il est essentiel de mener une réflexion plus large associant l'ensemble des actions et données environnant le niveau flotte. Ces actions peuvent en effet être influencées en continu non seulement par les comportements des sous-systèmes et systèmes mobiles, mais aussi par différents contextes opérationnels [20] tels que l'impact de la température extérieure sur les systèmes clim, voire même par l'infrastructure (par exemple : problème de balise d'arrêt en gare empêchant l'ouverture des portes), etc. L'idée est ainsi de proposer une

particularisation de ce Niveau Holonique afin de permettre une élaboration de conseils au niveau flotte basés sur une optimisation globale des coûts d'exploitation des coûts d'opérations et de politique de maintenance.

Ce Niveau Holonique remplace la couche de génération de conseils de la norme ISO 13374 (qui est destiné initialement à élaborer les conseille d'une système isolé[104]) par une nouvelle couche appelée « maintenance dynamique ». Rappelons que selon notre vision, la fonction de « génération de conseils » de cette norme est répartie entre les différentes Niveaux Holoniques, chaque Niveau Holonique pouvant élaborer ses propres conseils et recommandations et les envoyer aux Niveaux Holoniques supérieurs. Cette évolution offre également l'opportunité d'élaborer un déploiement distribué de ces niveaux holoniques (voir chapitre V).

L'élaboration des conseils liés aux fonctions de diagnostic et pronostic d'un système sont estampés sur les SurfEvents selon l'abstraction d'un Niveau Holonique « NH » comme suit(voir la Formule 3.9) :

$$|l'élaborations x| = \begin{cases} x \text{ estampé sur le SurfEvent(signal) , si } x \in NH \# 1 \\ x \text{ estampé sur le SurfEvent(indicateur) , si } x \in NH \# 2 \\ x \text{ estampé sur le SurfEvent(alarme) , si } x \in NH \# 3 \\ x \text{ estampé sur le SurfEvent(signalement) , si } x \in NH \# 4 \\ x \text{ estampé sur le SurfEvent(haut - niveau) , si } x \in NH \# 5 \text{ et } \# 6 \end{cases} \quad \text{Formule 3.9}$$

La particularisation de ce niveau porte sur le choix de la stratégie de maintenance à mettre en œuvre au sein d'un holon « maintenance dynamique ». Pour nous éclairer dans ce choix, nous avons mené une revue de la littérature. Nous avons identifié différents modèles de stratégie de maintenance voir par exemple [121], [122] et [123]. Ces modèles spécifient tous les étapes de définition des objectifs stratégiques pour le service de maintenance, d'identification des indicateurs de performance (KPI) pertinents, d'évaluation de l'état actuel de la maintenance, de définition des objectifs pour chaque indicateur de performance (KPI) et d'élaboration d'un plan d'action. Cependant nous notons que les résultats issus de la surveillance ne sont pas intégrés, ce qui rend la politique de maintenance difficile à adapter à l'évolution des états réels des systèmes mobiles sous surveillance. Dans [21], [124]–[127], les auteurs proposent des politiques de maintenance appliquées aux systèmes multi-composants (système composés de ses sous-systèmes). Ces politiques sont basées principalement sur le regroupement des interventions visant à réduire le coût de maintenance global d'un système[21]. Leurs processus de décision pour les déclenchements des opérations de maintenance se distinguent principalement en deux catégories: statique ou dynamique.

Notre choix s'est porté sur le modèle de politique de regroupement dynamique, car il permet de prendre en compte l'élaboration de conseils de chaque composant contrairement aux politiques statiques qui ne permettent pas de s'adapter à l'état de santé réel des composants. L'approche dynamique permet aussi de développer une méthode de planification adaptée à l'utilisation réelle des composants et de profiter des opportunités (remplacements, des inspections, des pannes, etc.) de maintenance offertes pour regrouper les opérations de maintenance et ainsi économiser des coûts logistiques. Pour plus de détail sur le choix de ce modèle, voir l'annexe B-2.

Parmi les modèles de politiques de regroupements dynamiques, nous nous intéressons plus particulièrement au modèle proposé dans[21]. En effet, il offre beaucoup d'autonomie aux systèmes sous surveillance pour fournir des opportunités de maintenance, contrairement à la plupart des politiques de regroupement dynamiques où le regroupement est décidé indirectement par les dates de maintenance des composants[21]. Le modèle retenu dans le cadre de nos travaux est basé sur la

MFOP « Maintenance Free Operating Period » qui permet de définir des périodes à la fin de chaque mission où le système peut être maintenu (Par exemple, pour des systèmes comprenant une structure parallèle, certains sous-systèmes redondants peuvent être défaillants sans impacter la réalisation de la mission). L'auteur a fait évoluer la politique de maintenance de la MFOP au niveau des processus de décision, d'évaluation et d'optimisation de la maintenance pour qu'elle soit capable d'intégrer les informations disponibles en ligne issues des systèmes pour adapter la décision de maintenance.

La MFOP est définie comme une période de fonctionnement pendant laquelle le système doit être en mesure de mener à bien ses missions sans action de maintenance et sans que l'opérateur ne soit limité en aucune manière en raison de défaillances du système[128]. Les maintenances planifiées entre les périodes de la MFOP sur un système sont effectuées afin de lui permettre d'accomplir avec succès la prochaine MFOP. Ils sont appelés « MRP » (Maintenance Recovery Period) [129]. La MRP est de longueur flexible, la probabilité que le système survive pendant la durée de la MFOP sachant qu'il était dans un état de fonctionnement au début de la période est appelé « MFOPS » (Maintenance Free Operating Period Survivability) [129]. Ainsi le calcul de $MFOPS(t)$ à l'instant t est présenté dans la Formule 3. 10).

$$MFOPS(t) = \frac{R_{sys}(t + t_{MFOP})}{R_{sys}(t)} \quad \text{Formule 3. 10}$$

avec $R_{sys}(t)$, la fiabilité du système à l'instant t . t_{MFOP} , la longueur de MFOP, il peut être décrit en unités de temps ou être définis en unités kilométriques dans le domaine du transport.

La MFOP cherche à assurer aux exploitant la totale disponibilité opérationnelle de leurs systèmes sur des périodes d'opérations données (voir la Figure 3. 13). Pour ce faire, la MFOP diffère virtuellement toutes les maintenances correctives à venir vers des périodes planifiées de maintenances préventives. Cette caractéristique permet de maximiser la disponibilité opérationnelle des systèmes. En évitant au maximum les arrêts non planifiés, la MFOP réduit les coûts de réparation du système mais également le support logistique et la gestion des pièces de rechange en diminuant le besoin de ré-planifier les missions[21].

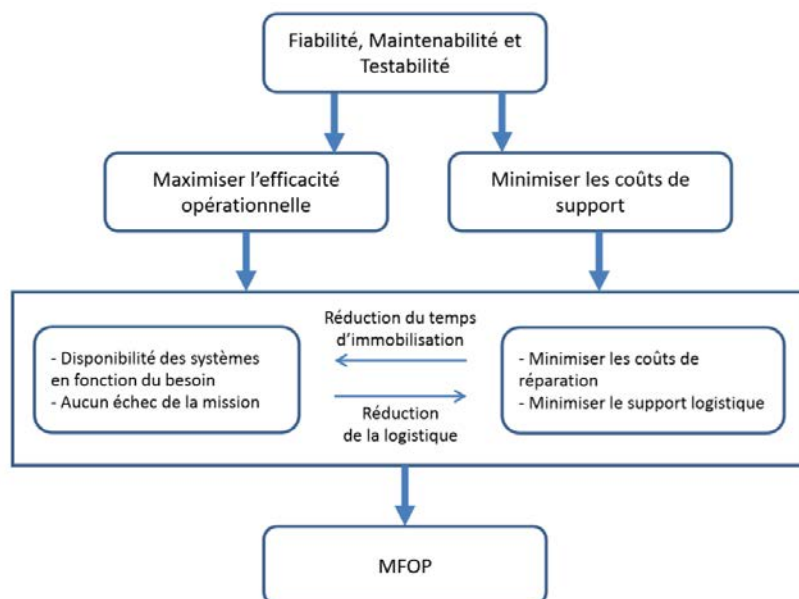


Figure 3. 13: Motivations pour la MFOP[130]

Cette approche présente toutefois trois inconvénients : 1) elle ne considère pas en détail les processus liés aux opérations de maintenance tels que les différents type d'activités ni leurs risques de blocage pouvant allonger la durée de ces opérations et des pénalités de temps d'écoulement et de retard[131] ; 2) certaines variables de la MFOP ne sont pas dynamiques, à l'image de la longueur de la MFOP et le niveau de confiance lié à la MFOP à l'instant t ; 3) elle s'adresse principalement aux exploitants afin d'assurer la disponibilité opérationnelle des composants(sous-systèmes) à l'échelle d'un système, mais pas à l'échelle d'une flotte de systèmes, qui dispose en plus de ses propres critères applicables à l'ensemble des systèmes mobile de la flotte, par exemple, d'entrée en vigueur de normes environnementales, de gestion d'aléas climatiques ou de décisions politiques régionales, etc.

Pour pallier ces inconvénients, nous nous proposons d'intégrer à la MFOP l'analyse des causes profondes RCA (Root cause analysis)[132] et le DCA((Difference of Convex functions Algorithm)[131], et dans le but d'identifier, analyser et résoudre efficacement les risques de blocage pouvant prolonger la durée de ces opérations de maintenance, de minimiser le coût et la lenteur des opérations de maintenance et de gérer dynamiquement le calcul de certaines variables de la MFOP. Nous nous proposons également d'adapter la MFOP aux contextes de la flotte afin de gérer la disponibilité opérationnelle des systèmes mobile de la flotte.

Il est important de noter que nous avons conscience que ce choix de politique de maintenance n'est pas systématique et qu'il est guidé ici par notre expérience dans le domaine de la maintenance d'une flotte de systèmes complexes mobiles et homogènes. Dans d'autres contextes, un autre choix est potentiellement faisable. Sans le prouver dans ce document, nous pensons que notre architecture est capable de supporter la diversité des choix de politique de maintenance.

Maintenant que ce choix est fait, nous pouvons donner les éléments importants concernant la particularisation de ce niveau. Ainsi, en s'appuyant sur l'état de santé actuel, prévisionnel et sur les différents contextes opérationnels des systèmes mobiles de la flotte, le Niveau Holonique « Maintenance Dynamique » a comme particularité d'évaluer les indicateurs de performance (KPI) correspondant à une stratégie et des opérations de maintenance. Cette évaluation est menée en trois phases : 1) une justification des contextes opérationnels au niveau flotte de différents conseils issus des systèmes mobiles ; 2) l'application d'une politique de maintenance dynamique basée sur le modèle MFOP étendu avec la méthode RCA; et 3) une gestion des opérations de maintenance et au retour d'expériences(REX).

Le holon « maintenance dynamique » de ce niveau est constitué de trois (sous-)holons, respectivement de « justification de contextes », de « politique de maintenance dynamique » et de « gestion des opérations de maintenance et de REX », chacun mettant en œuvre l'une de ces trois phases, et chacun d'entre eux potentiellement et également décomposé de manière récursive. La Figure 3. 14 représente cette modélisation. Le processus global est le suivant : lorsqu'un système mobile envoie son état de santé actuel (SurfEvent de type « signalement ») et prévisionnel (SurfEvent de type « haut-niveau ») vers le centre de maintenance, ils sont pris en charge par l'holon « justification de contextes » afin d'étudier au niveau de la flotte s'il ne s'agit pas d'une élaboration erronée (Par exemple, de par un manque de vue sur les contextes liés à l'infrastructure). Si le résultat est probant, il informe l'Holon « politique de maintenance dynamique » afin de prioriser et optimiser les opérations à mener selon l'objectif et la politique de maintenance adoptés. Cette étape peut être effectuée en parallèle avec celle de « justification de contextes ». Le résultat de l'Holon « politique de maintenance dynamique » permet à l'Holon de « gestion des opérations de maintenance et REX » d'identifier et optimiser les causes bloquantes durant les opérations

d'investigation, réparation, etc. et partager le REX avec les Holons de « justification de contextes », « politique de maintenance dynamique » et les autres Holons des systèmes mobiles concernés.

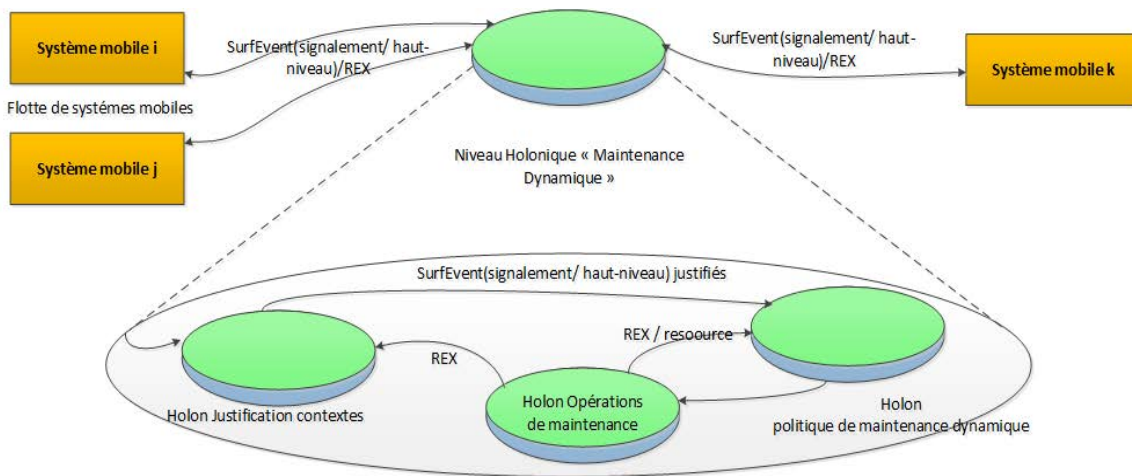


Figure 3. 14: Niveau Holonique « Maintenance Dynamique »

Les sous-section suivantes détaillent plus spécifiquement les fonctions des trois Holons « justification de contextes » « politique de maintenance dynamique » et « gestion des opérations de maintenance et REX » et les sous-holons qui les composent récursivement, le cas échéant.

4.3.1. Holon « Justification de contexte »

L'élaboration de conseils au niveau des systèmes mobiles dresse le bilan de surveillance des Holons de différents Niveaux Holoniques qui ne disposent pas souvent des vues générales au niveau flotte pour leur permettre d'associer les contextes liés à l'infrastructure, aux opérations d'exploitation et/ou aux autres systèmes hétérogènes. Un exemple illustratif de ce manque de vue, est typiquement lorsqu'une défaillance dans l'infrastructure empêche le bon fonctionnement d'un système mobile. Dans ce cas, selon notre approche, les Holons de surveillance de ces systèmes mobiles relèvent a priori des défaillances ou des dégradations qu'il est impossible d'imputer au système mobile, ces relevés seront corrigés par un Holon ayant une vue plus générale au niveau flotte.

La mission de ce Holon « justification de contexte » est ainsi d'avérer ou non les défaillances ou les dégradations des systèmes mobiles au niveau flotte et ainsi réduire le coût lié à la génération de fausses alarmes. L'algorithme de fonctionnement de cet Holon est basé sur un mécanisme d'inhibition couramment utilisé dans un tel contexte. Il est décrit ci-dessous (voir l'Algorithme 3. 3).

Entrée : SurfEvent_i

Début

le Niveau Holonique « maintenance dynamique » ajoute un coût lié au **SurfEvent_i** dans l'Holon « politique de maintenance dynamique » ;

Evaluation du **SurfEvent_i** et association à son contexte niveau flotte dans une durée « T » :

$$\text{Résultat} = \{I(\text{SurfEvent}_i, t, c_i), \text{ tant que } t \leq T\}$$

Si Résultat = « vrai » alors

Inhibition du **SurfEvent_i** et soustraction de son coût au niveau de l'Holon « politique de maintenance dynamique » ;

Mise à jour des connaissances des Holons des systèmes mobiles de la flotte.

Sinon alors

Confirmation du **SurfEvent_i** à l'Holon « politique de maintenance dynamique ».

Fin

Algorithme 3. 3: Algorithme de fonctionnement de l'Holon « justification de contextes » au niveau flotte à partir de l'état de santé d'un système mobile « i »

Dans l'Algorithme 3. 3, dans un premier temps, l'Holon « justification de contextes » reçoit par le biais du Niveau Holonique « Maintenance Dynamique », l'état de santé actuel et prévisionnel d'un système mobile au travers d'un **SurfEvent_i**. Il évalue le **SurfEvent_i** en fonction de son type (signal, indicateur, alarme, signalement, Haut-niveau) et l'associe à son contexte niveau flotte pour une durée « T ». Ce contexte contient des informations collectées via des applications hétérogènes, par exemple, les codes défaut associés, les données de l'infrastructure, la position GPS, l'historique de retour d'expériences, etc. Le résultat de l'analyse de la justification peut être soit « vrai » soit « faux ». Il est « faux » si l'Holon n'arrive pas à justifier la défaillance ou la dégradation. Dans ce cas, il confirme son résultat à l'Holon « politique de maintenance dynamique ». Sinon, cela signifie que l'Holon arrive à justifier la défaillance ou la dégradation, et dans ce cas, il doit l'inhiber par la soustraction du coût au préalablement ajouté par le Niveau Holonique « maintenance dynamique » au niveau de l'Holon « politique de maintenance dynamique ». Les connaissances des Holons des systèmes mobiles de la flotte sont alors mises à jour. Le principe d'ajouter et soustraire (mettre à jour) des coûts liés aux états de santé actuel, prévisionnel des systèmes permet à l'Holon « politique de maintenance dynamique » d'être le plus réactif possible en intégrant dans son modèle le coût de l'état de santé durant l'étape de justification.

4.3.2. Holon « Politique de maintenance dynamique »

Ce second Holon reçoit en entrée de la part de l'Holon « justification de contextes » le résultat de justification des états de santé actuel et prévisionnel (**SurfEvent_i**) qui comprend des informations sur les défaillances ou des dégradations des systèmes mobiles. Il reçoit également le REX de la part de l'Holon « gestion des opérations de maintenance et REX ». Ces informations sont alors évaluées au regard de la politique de maintenance dynamique adoptée précédemment décrite (la MFOP) pour élaborer une planification dynamique des opérations de maintenance. Le résultat de cette planification dynamique des opérations de maintenance est alors transmis vers l'Holon « gestion des opérations de maintenance et REX ».

La planification dynamique des opérations de maintenance de l'ensemble des systèmes mobiles de la flotte constitue une étape clé du processus global et un sujet de recherche à proprement parler. Nous la décrivons ici succinctement, avec nos choix de conception, bien qu'elle sorte du cadre de cette thèse et qu'elle reste perfectible sur de nombreux aspects, notamment sur ceux liés à l'optimisation mathématique globale de la planification de ces opérations (en utilisant par exemples les outils de la recherche opérationnelle).

Dans le cadre de cette thèse, la planification dynamique des opérations de maintenance est effectuée au travers d'une heuristique inspirée de la MFOP et décrite en deux étapes : **la première étape** consiste à évaluer à travers différents (sous-)Holons, la politique de maintenance de chacun des systèmes mobiles de la flotte (planification des opérations au niveau des composants des systèmes mobiles). Le résultat d'évaluation de cette première étape alimente **la deuxième étape** qui consiste, pour un Holon de planification niveau flotte, à évaluer l'impact sur l'exploitation de la flotte. Les sous-section suivantes détaillent les deux types d'Holons responsables chacun de l'une de ces deux étapes, Holons que nous appelons respectivement « Holon de politique de maintenance des composants d'un système mobile » et « Holon impact sur exploitation de la flotte » (voir Figure 3. 15).

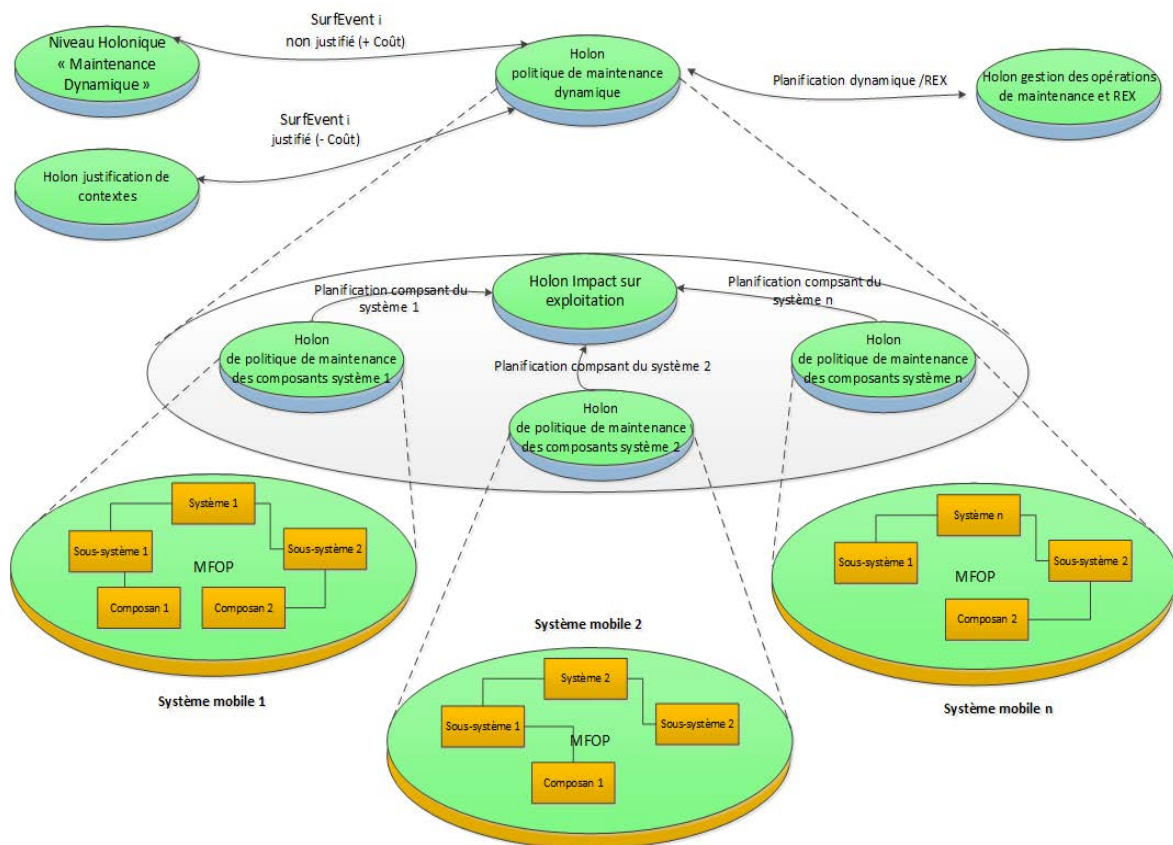


Figure 3. 15: Holon « politique de maintenance dynamique » d'une flotte des systèmes mobiles

- **Holon « politique de maintenance des composants d'un système mobile »**

Ce (sous-)Holon intègre et met en œuvre une politique de maintenance de composants des systèmes mobiles que nous avons proposé par ailleurs [21]. La politique proposée se base sur la

MFOP afin de garantir avec un certain niveau de confiance, des périodes d'opérations en continu sans arrêt lié soit à la maintenance soit à la panne d'un système. Elle s'appuie à la fois sur la structure fiabiliste du système et sur les informations de surveillance disponibles en ligne. Celles-ci se composent d'informations relatives à l'état de santé des composants mais également à leurs conditions d'utilisation. La flexibilité du processus permet d'intégrer, dans la décision, des niveaux d'informations différents suivant les composants. La Figure 3. 16 présente les différentes étapes de la politique de maintenance développée et mise en œuvre par cet Holon:

- **La première étape** du processus de décision consiste à définir à l'instant t (qui représente soit la fin d'une MFOP, soit une panne au niveau du système) la nécessité d'une intervention de maintenance sur le système. L'objectif est d'évaluer la probabilité que le système fonctionne jusqu'à la fin de la prochaine MFOP, sachant l'information disponible à t (voir la Formule 3. 11) :

$$MFOPS(t) = \frac{R_{syst}(t + t_{MFOP}/SurfEvent_{i,t_{\{i \text{ de } 1 \text{ à } n\}}})}{R_{syst}(t/SurfEvent_{i,t_{\{i \text{ de } 1 \text{ à } n\}}})} = \frac{P(T_{syst} > t + t_{MFOP}/SurfEvent_{i,t_{\{i \text{ de } 1 \text{ à } n\}}})}{P(T_{syst} > t/SurfEvent_{i,t_{\{i \text{ de } 1 \text{ à } n\}}})} \quad \text{Formule 3. 11 [21]}$$

Avec T_{syst} : la durée de bon fonctionnement du système, n : le nombre des composants, $SurfEvent_{i,t}$: l'information disponible sur l'état de santé du composant i à l'instant t . Cette information peut être quantitative (mesure de dégradation) ou qualitative (par exemple : « Marche » / « Panne » / « Mauvaise » / Critique » / « acceptable » / « Bon »).

- **La seconde étape** du processus de décision consiste à comparer la $MFOPS(t)$ du système avec un niveau de confiance NC spécifié:
 - Si la $MFOPS(t) > NC$, le processus de décision considère que l'intervention de maintenance n'est pas nécessaire. Le système mobile peut alors être déployé sur la prochaine MFOP sans passer par l'atelier.
 - Dans le cas contraire, si la $MFOPS(t) < NC$, l'intervention de maintenance est jugée indispensable.

Lorsqu'une intervention de maintenance est requise, la politique de maintenance doit identifier les composants à remplacer pour que le système puisse assurer la prochaine MFOP avec le niveau de confiance demandé. Dans ce cadre, un problème d'optimisation sous contrainte est introduit pour sélectionner les remplacements à réaliser. Le choix des maintenances à effectuer sur un système mobile peut être réalisé de différentes manières. De façon générale, le problème peut être exprimé comme suit (voir la Formule 3. 12):

$$\min_{\{X\}} J_w(X) \text{ avec } MFOPS(X, t) > NC \quad \text{Formule 3. 12 [21]}$$

Avec X un ensemble de composants à remplacer pour une solution donnée, J_w le critère de décision de maintenance avec w l'indice du critère de décision étudié, $MFOPS(X, t)$ la $MFOPS$ du système à t après remplacement de l'ensemble X et NC le niveau de confiance à atteindre sur la prochaine MFOP.

Les critères de décision de maintenance J_w proposés pour le choix des opérations de maintenance à effectuer sur un système mobile sont les suivants :

- **Le premier critère** proposé, appelé J_1 , a pour objectif d'atteindre le niveau de confiance spécifié sur la prochaine MFOP en minimisant les coûts de remplacement (voir Formule 3. 13).

$$J_1(X) = \sum_{i=1, i \in X}^n (C_i + D_i \times \tau_{MO}) - \sum_{v=1}^{m'} \delta_{\{U_v \subset X\}} \times DS_v \times \tau_{MO}$$

Formule 3. 13 [21]

Avec m' le nombre d'ensembles d'opérations de maintenance dépendantes, $\delta_{\{U_v \subset X\}}$ une variable binaire qui vaut 1 quand le v ième ensemble d'opérations de maintenance dépendantes est inclus dans l'ensemble X (0 sinon), DS_v la durée économisée si la maintenance est effectuée simultanément sur les composants de cet ensemble, C_i le coût du composant i , D_i la durée de remplacement du composant i , τ_{MO} le taux horaire de la main d'œuvre.

- **Le second critère** proposé, appelé J_2 , est basé sur la minimisation du ratio entre les coûts de remplacement et l'évolution de la MFOPS avant et après réalisation de l'ensemble de remplacement X . Ce critère vise à promouvoir le meilleur compromis entre coûts de remplacement et gain de fiabilité sur la prochaine MFOP (voir la Formule 3. 14). Il est exprimé par:

$$J_2(X) = \frac{\sum_{i=1, i \in X}^n (C_i + D_i \times \tau_{MO}) - \sum_{v=1}^{m'} \delta_{\{U_v \subset X\}} \times DS_v \times \tau_{MO}}{MFOPS(X, t) - MFOPS(t)}$$

Formule 3. 14 [21]

- **Le troisième et dernier critère**, nommé J_3 , est basé sur la minimisation du ratio entre les coûts de remplacement et une quantité $DR(X)$. Celle-ci définit, pour un ensemble de remplacements X , la distance restante en kilomètres avant que la fiabilité du système n'atteigne le niveau de confiance spécifié NC (voir la Formule 3. 15) .

$$J_3(X) = \frac{\sum_{i=1, i \in X}^n (C_i + D_i \times \tau_{MO}) - \sum_{v=1}^{m'} \delta_{\{U_v \subset X\}} \times DS_v \times \tau_{MO}}{DR(X)}$$

Formule 3. 15 [21]

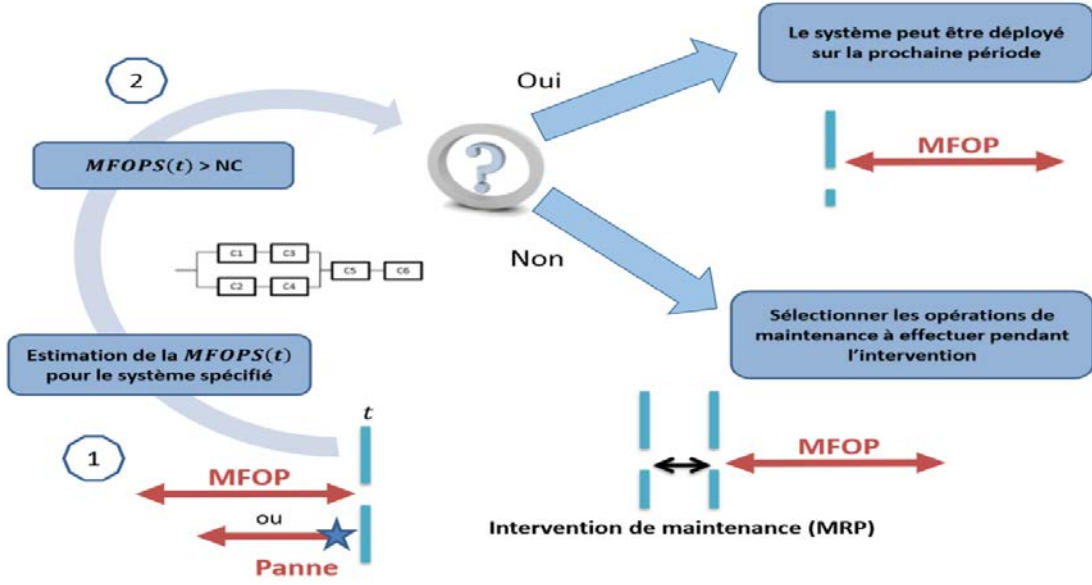


Figure 3. 16: Synthèse de la politique de maintenance proposée[21].

Comme introduit dans le chapitre I, la politique de maintenance d'un système mobile est évaluée à travers du coût de maintenance total du système $C_{TM}(t)$. Il est calculé sur un horizon de temps fini avec $C_R(t)$ le coût de remplacement des composants du système sur $[0, t]$ (voir la Formule 3. 16), $C_{SC}(t)$ le surcoût lié à la maintenance corrective lorsque le système est en panne sur $[0, t]$ (voir la Formule 3. 17), et enfin $C_{SD}(t)$ le surcoût lié au diagnostic lorsque le système est en panne sur $[0, t]$ (voir la Formule 3. 18). Ces coûts intègrent tous les remplacements effectués de manière préventive ou corrective. Ils sont décrits ci-dessous :

$$C_R(t) = \sum_{K=1}^{N_{AM}(t)} [(\sum_{i=1}^n \delta_{i,k} \times (C_i + D_i \times \tau_{MO}) - \sum_{v=1}^{m'} \delta_{\{U_v \in M_k\}} \times DS_v \times \tau_{MO}) + C_{Log}]$$

Formule 3. 16 [21]

Avec M_k un ensemble comprenant les remplacements effectués pendant l'arrêt de maintenance k , $N_{AM}(t)$ le nombre d'arrêts de maintenance sur l'intervalle de temps $[0, t]$, n le nombre de composants dans le système, $\delta_{i,k}$ une variable binaire qui indique le remplacement du composant i pendant l'arrêt de maintenance k , et C_{Log} le coût logistique lié à l'entrée du système en atelier de maintenance.

$$C_{SC}(t) = \sum_{l=1}^{N_P(t)} [(\sum_{i=1}^n \delta_{i,l} \times (D_i \times \tau_I) - \sum_{v=1}^{m'} \delta_{\{U_v \in M_l\}} \times DS_v \times \tau_I) + C_{Remor} + ((D_{Remor} + D_{Log}) \times \tau_I)]$$

Formule 3. 17 [21]

avec $N_P(t)$ le nombre de pannes du système sur l'intervalle de temps $[0, t]$, $\delta_{i,l}$ une variable binaire qui indique le remplacement du composant i pendant l'arrêt de maintenance correctif l , τ_I le coût de la perte d'exploitation par heure d'immobilisation, C_{Remor} le coût de remorquage du système, D_{Remor} la durée de remorquage du système et enfin D_{Log} la durée logistique de prise en charge du système à son entrée en atelier.

$$C_{SD}(t) = \sum_{l=1}^{N_P(t)} [(C_{U,D} + D_{U,D} \times \tau_I) \times N_{SIF}]$$

Formule 3. 18 [21]

avec $C_{U,D}$ le coût unitaire de diagnostic pour un composant (coût supposé fixe), $D_{U,D}$ la durée unitaire de diagnostic pour un composant (durée supposé fixe) et enfin N_{SIF} le nombre de composants dans le système dont l'information de surveillance n'est pas disponible.

Les paramètres de notre politique, à savoir la longueur de la période d'une MFOP et le niveau de confiance NC , sont alors optimisés en résolvant ce problème d'optimisation sous contraintes en fonction du coût total de maintenance. Ce coût, évalué sur un horizon fini, intègre les coûts directs associés aux opérations de maintenance et les coûts indirects engendrés par les immobilisations.

Le résultat d'évaluation de la fonction de ce Holon contient ainsi soit « rien » si la « $MFOPS(t) > NC$ », soit sinon une planification optimisée contenant la liste des composants à remplacer pour que le système mobile puisse assurer la prochaine MFOP avec le niveau de confiance demandé. Cette liste est destinée à être évaluée par l'Holon « impact sur exploitation de la flotte » détaillé dans la sous-section suivant.

- **Holon « impact sur exploitation de la flotte »**

Cet Holon reçoit en premier lieu en entrée de la part des différents Holons « politique de maintenance des composants d'un système mobile », la planification contenant la liste des composants à remplacer pour que le système mobile puisse assurer sa prochaine MFOP avec le niveau de confiance demandé. Il croise alors ces informations avec celles issues de la flotte afin de mesurer leurs impacts sur l'exploitation de celle-ci.

En considérant l'ensemble des opérations de maintenance à effectuer sur la flotte de systèmes mobiles comme les opérations à effectuer sur les composants d'un système, le problème revient alors à identifier et sélectionner les différents systèmes mobiles à aiguiller vers les dépôts de maintenance pour que la flotte puisse être assurée d'une disponibilité satisfaisante avec le niveau de confiance demandé. Dans le cadre de nos travaux, ce niveau de confiance est noté (m). Il est fixé par les opérateurs de maintenance, il indique le nombre maximal de visite à ne pas dépasser durant une période (cf. sous-section Holon « gestion des opérations de maintenance et REX »).

Ainsi, nous nous trouvons à nouveau face à un problème d'optimisation sous contrainte pour identifier et aiguiller les systèmes mobiles vers les dépôts de maintenance, sous la contrainte de ne pas dépasser le nombre maximal de visite m durant une période. Ce problème peut être modélisé de la manière suivante (voir la Formule 3. 19) :

$$\min_{\{Y\}} J(Y) \text{ sous – contraintes } MFOPS_F(Y, t) \leq m \quad \text{Formule 3. 19}$$

Avec Y un ensemble des systèmes mobiles à aiguiller vers les dépôts de maintenance, J le critère de l'impact choisi sur l'exploitation de la flotte, $MFOPS_F(Y, t)$ la $MFOPS$ de la flotte(F) à t après sélection de l'ensemble Y , et m le nombre maximal de visite à ne pas dépasser sur la prochaine MFOP.

Le critère de l'impact sur l'exploitation de la flotte regroupe en fait un ensemble de critères appliquées à l'ensemble des systèmes mobile de la flotte et peut être géré par ordre de priorité. Ces critères sont exprimés, par exemple par les disponibilités des dépôts selon la distance géographique, le type de mission (commerciale, hors commerciale) d'un système mobile, le taux d'occupation moyenne d'un ligne / station, les évènements sociaux, l'entrée en vigueur des normes environnementales, les aléas climatiques, les décisions politique, etc.

L'impact sur l'exploitation de la flotte est évalué au travers du coût de maintenance total des systèmes mobile de la flotte $C_{TMF}(t)$. Il est calculé sur un horizon de temps fini (la durée de contrat de maintenance, par exemple) et se décompose de la façon suivante (voir la Formule 3. 20):

$$C_{TMF}(t) = \sum_{i=1}^N C_{TMSi}(t) * \delta_i \quad \text{Formule 3. 20}$$

Avec $C_{TMSi}(t)$ le coût de maintenance total d'un système mobile sur $[0, t]$, N le nombre de systèmes mobiles de la flotte, δ_i une variable binaire qui vaut 1 si le système mobile i est sélectionné pour aller dans un dépôt de maintenance, 0 sinon.

Le résultat de planification de cet Holon « impact sur l'exploitation de la flotte » est alors transmis à l'Holon « gestion des opérations de maintenance et REX ».

4.3.3. Holon « gestion des opérations de maintenance et REX »

L'objet de l'Holon « gestion des opérations de maintenance et REX » est d'analyser, via un modèle d'analyse des causes profondes, les causes bloquantes lors de déroulement des processus de réparations maintenance et de gérer dynamiquement le calcul de certaines variables de la MFOP des Holons « politique de maintenance des composants d'un système mobile » (à l'image de la longueur de MFOP) et le niveau de confiance NC et certaines variable de le $MFOPS_F$ de l'Holon « impact sur exploitation de la flotte » (à l'image de m , le nombre maximal de visite à ne pas dépasser sur la prochaine $MFOPS_F$). Par cause bloquante, nous entendons les retards liés à livraison des pièces, à la disponibilité de infrastructures (dépôts de maintenance), au manque d'outillage et de main d'œuvre, ou encore au non-respect de plan de travaux.

L'Holon « gestion des opérations de maintenance et REX » reçoit de la part de l'Holon « politique de maintenance dynamique » les planifications dynamiques des opérations de maintenance (voir Figure 3. 17). Il analyse alors ces informations de planification à l'aide d'une méthode 5P (Pourquoi) couplée aux 5M (Main d'œuvre, Moyen, Méthode, Matière, Milieu) du modèle d'analyse des causes profondes RCA[132]. L'objectif est comme indiqué, d'identifier et analyser les causes bloquantes en continu lors du déroulement des processus de réparations maintenance et de gérer dynamiquement le calcul de certaines variables de la MFOP. Le résultat d'évaluation de ce modèle sert à alimenter en REX les Holons « politique de maintenance dynamique », « justification de contextes » et les modèles de surveillance des Holons des systèmes mobiles. Ces résultats peuvent également servir lors des futures opérations de maintenance et pour adapter les modèles de conception (retrofit, conception des générations futures des systèmes mobiles, etc.). La Figure 3. 17 formalise ce processus en utilisant la représentation « pm » qui renseigne à la « date » du jour, l'état et le type de d'opération de maintenance (en attente, planifiée, non finale, corrective, préventive, etc.) . En fonction des besoins pour les opérations (en cours, pièces en attente, infrastructure en attente, etc.), des services et des ressources sont alors alloués. Dans chaque étape du processus est nous nous associons l'algorithme DCA afin minimiser le coût et la lenteur des tâches des opérations maintenance.

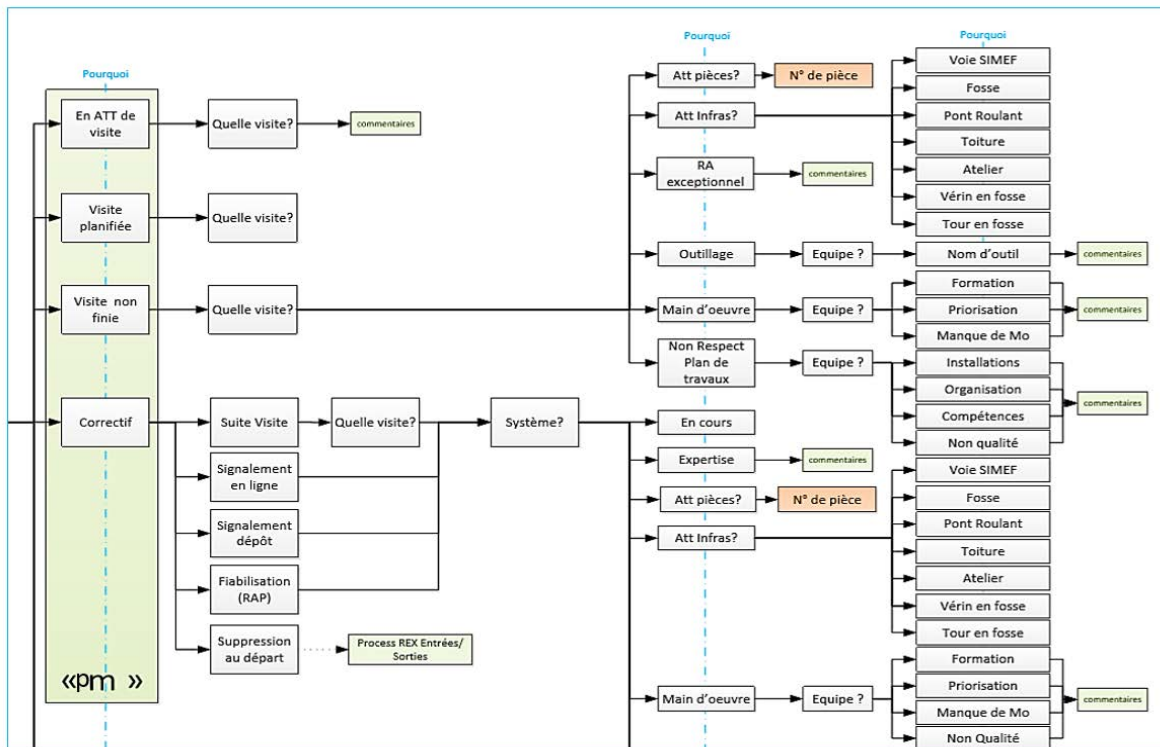


Figure 3. 17: Processus «pm » pour la gestion des opérations de maintenance et REX

Durant la réalisation des opérations de maintenance, le processus «pm » permet d'évaluer plusieurs indicateurs tels que la durée moyenne de séjour d'une opération de maintenance (selon le type de visite ou non), la durée moyenne d'une opération d'investigation, réparation, etc. et de vérifier l'atteinte des objectifs(KPI) en calculant par exemple des indicateurs pour analyser les différents types de visites de la flotte de façon à souligner leurs tendances à plus long terme. Par exemple, le nombre de visites durant une période est calculé selon une moyenne mobile comme suit (voir la Formule 3. 21):

$$\bar{x}_n = \bar{x}_{n-1} + \frac{x_n - x_{n-N}}{N} \quad \text{Formule 3. 21}$$

Avec:

\bar{x}_{n-1} : la moyenne mobile de la période précédente, N : la période (par semaine, par mois et par semestre),
 x_n : le nombre distinct de visites par jour, ces visites sont de natures prédéfinies (correctives, préventives, imprévu, etc.)

On peut alors calculer des taux de robustesse par rapport à une cible. Par exemple, toujours dans le cas du nombre de visites par jour, le taux de robustesse peut être calculé comme suit (voir la Formule 3. 22):

$$\text{taux de robustesse} = \frac{\bar{x}_n * 100}{m} \% \quad \text{Formule 3. 22}$$

Ces indicateurs et objectifs(KPI) offrent ainsi à l'exploitant de la flotte des données précieuses pour améliorer et maîtriser en continu la gestion et l'optimisation du coût de ces ressources et opérations. Ils permettent aux Holons « politique de maintenance dynamique » d'effectuer des ajustements dynamiques des variables de la MFOP et au Holon « impact sur exploitation de la flotte » d'effectuer des ajustements dynamiques de m (nombre maximal de visite à ne pas dépasser

sur la prochaine *MFOPS_F*). Ils permettent enfin d'offrir aussi des données précieuses pour les futurs modèles de conception afin d'évaluer et d'investir sur la fiabilité des composants de ces futurs systèmes.

Les trois éléments clés de notre contribution à savoir la modélisation d'une forme générique d'évènements « SurfEvent », l'architecture par Niveau Holonique « EMH² » ; et une conception de « EMH² » particularisée en fonction des niveaux holoniques ont été présentés.

5. Conclusion

Dans ce chapitre, une architecture de surveillance à base d'agents intelligents pour l'aide à la maintenance d'une flotte de systèmes mobiles, appelée EMH² a été proposée. Cette architecture permet d'élaborer une surveillance embarquée et au sol robuste de manière décentralisée et coopérative. Les travaux présentés dans ce chapitre ont fait l'objet de deux présentations dont une dans une conférence internationale [1],[133]. Une publication actuellement en seconde lecture a également été réalisée [134].

Nous avons présenté dans ce chapitre les trois éléments clés de notre contribution à savoir la modélisation d'une forme générique d'évènements « SurfEvent » ; l'architecture par Niveaux Holoniques « EMH² » ; et une conception de « EMH² » particularisée en fonction des niveaux holoniques. Nos grands choix de conception sont tous basés sur des études de comparaison explorant les différentes approches de modélisation. Pour implémenter la fonction de diagnostic, nous avons retenu le système expert en tant que méthode de surveillance basée sur la connaissance. Ce système expert sert à modéliser les holons des Niveau Holonique « Acquisition de données, Manipulation de données, Détection et Evaluation de Santé » de l'architecture EMH² pour leur donner la capacité de localiser, de raisonner et de prendre des décisions dans un

environnement réparti. Pour la fonction de pronostic, nous avons retenu la méthode de l'arbre de décision en tant que méthode basée sur des données historiques afin d'octroyer aux Holons du Niveau Holonique « Pronostic », la capacité de prédire et expliquer les influences entre les événements. Enfin, pour la fonction d'élaboration de conseils, nous avons retenu un modèle basé sur la MFOP afin de permettre aux Holons du Niveau Holonique « maintenance dynamique », de définir dynamiquement des périodes à la fin de chaque mission où le système peut être maintenu. Ses résultats sont analysés à l'aide de la méthode RCA afin d'améliorer l'efficacité des opérations de maintenance.

Ce chapitre nous a ainsi permis d'apporter des éléments de réponse par rapport aux spécifications cibles introduites dans la section 2.6 du chapitre 1. En effet, nous avons pu répondre aux spécifications cibles liés à la modularité, à travers de l'introduction du modèle générique « SurfEvent » ; du déploiement repartie et automatisé, à travers de l'introduction de l'architecture Holonique ; et de la réactivité, adaptabilité, précision et optimisation du coût de la maintenance d'une flotte de systèmes mobiles, à travers de l'introduction de l'étape de particularisation des niveaux.

Le chapitre suivant propose une simulation du niveau Holonique « Evaluation de Pronostic » pour valider certains aspects de notre contribution. Un Processus de déploiement et les principaux aspects méthodologiques sont détaillés, illustrant une mise en œuvre possible de nos propositions dans le contexte technologique actuel.

Ch. IV Simulation, mise en œuvre et déploiement de l'architecture

Introduction

Dans le chapitre précédent, nous avons présenté l'architecture de surveillance EMH², à base d'holons pour l'aide à la surveillance pour contribuer à la maintenance proactive d'une flotte de systèmes mobiles, complexes et homogènes. Cette architecture élabore une surveillance embarquée et au sol robuste de manière décentralisée et coopérative. Sa conception intègre la modélisation d'une forme générique d'évènements « SurfEvent », d'une architecture par Niveaux Holoniques et une particularisation de ces niveaux en fonction de leurs missions.

Ce chapitre traite de la **faisabilité** et de l'**applicabilité** de notre contribution. Pour ce faire, il contient une étude en simulation et présente une méthodologie de mise en œuvre ainsi qu'une implémentation possible de l'architecture de surveillance EMH² en fonction de la technologie disponible au moment de la rédaction de cette thèse. Il est composé de quatre parties:

- La première partie de ce chapitre sera consacrée à l'étude en simulation qui concerne spécifiquement le niveau Holonique « Evaluation de pronostic ». Cette simulation vise à tester la modélisation des Holons « **pronostic local** » au sein d'une flotte de système mobile et leurs

performances décrite en termes d'estimation de la durée moyenne de génération de connaissances et d'intégration des retours d'expérience sur le processus de surveillance par comparaison avec une situation de référence.

- La seconde partie de ce chapitre sera consacrée aux aspects déploiement et méthodologique pour aider tout décideur à mettre en œuvre, dans son contexte applicatif, notre proposition.
- La troisième partie de ce chapitre sera consacrée à une implémentation possible de l'architecture EMH².
- La quatrième partie sera consacrée à la proposition d'un guide méthodologique de déploiement pour accompagner les décideurs dans leur démarche applicative de l'architecture EMH².

Les points 2), 3) et 4) ci-dessus sont présentés, traités et discutés en tenant compte de l'offre technologique disponible au moment de la rédaction de ce mémoire. Il est ainsi possible de les faire évoluer dans un futur proche en tenant compte de l'évolution à court et moyen terme des technologies.

1. Simulation du niveau Holonique « Evaluation de Pronostic »

Le niveau Holonique « Evaluation de Pronostic » joue un rôle primordial pour l'efficacité de la surveillance (voir chapitre 3, section 4.2). Ce niveau a la responsabilité de classer l'influence entre les SurfEvents, de prédire de nouveaux SurfEvents ou d'expliquer des situations qualitatives ou quantitatives, au sein d'un système mobile et au niveau de la flotte entière. Ses résultats et recommandations de surveillance servent de base aux modèles de surveillances des autres Niveaux Holoniques. Par exemple les Holon du Niveau Holonique « Evaluation de Santé » exploite directement les résultats, recommandations et conseils du Niveau Holonique « Evaluation de Pronostic » afin d'enrichir leurs connaissances, raisonnement et leur capacité de décision. L'étude de notre simulation est structurée en six étapes détaillées comme suit.

1.1. Objectif de la simulation

A travers cette simulation, on vise à simuler les comportements et coopérations des Holons « **pronostic local** » du niveau Holonique « Evaluation de Pronostic ». L'indicateur de performance choisi est la mesure de la durée moyenne d'élaboration de ses conseils (essentiellement des connaissances et REX). Cette durée relève, dans le cadre de la spécification cible définie dans le chapitre 1 « **adaptabilité et apprentissage** » du temps nécessaire pour caractériser, comprendre ou organiser le processus de surveillance d'un équipement ou d'un système nouveau ou révisé.

Le modèle de simulation contient les algorithmes de construction de l'arbre de décision présenté auparavant (voir la section 4.2 du chapitre 3), dans le but de classer les dépendances et les relations entre les SurfEvents de différents niveaux Holoniques pour un intervalle de temps donné et calcule la durée moyenne de cette classification. Le modèle prend en entrée un ensemble de SurfEvents d'un système cible et génère en sortie une durée moyenne de classification. La Figure 4. 1 suivant illustre le modèle de simulation pour un ensemble de SurfEvents, où chaque SurfEvent de l'ensemble est expérimenté par le biais des autres SurfEvents afin d'expliquer ou de prédire son comportement.

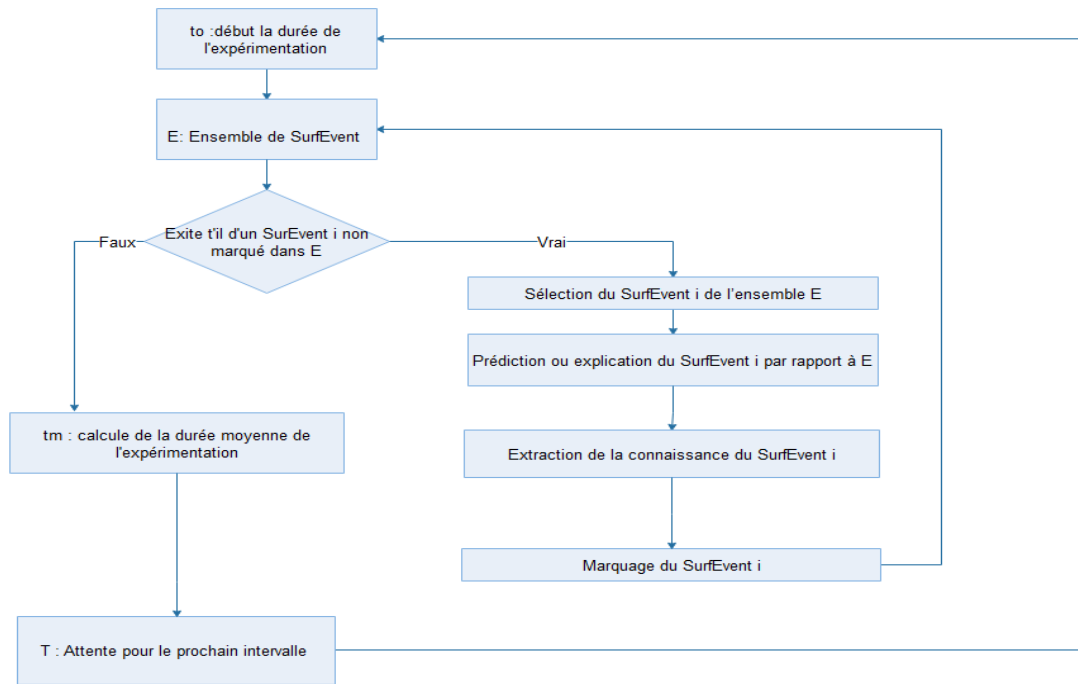


Figure 4. 1: modèle de la simulation d'un ensemble de SurfEvents

1.2. Situation actuelle de référence

La situation actuelle, considérée comme référence dans notre étude, consiste à prédire uniquement les évènements quantitatifs sur intervalle de temps donné, avec une méthode d'analyse en composantes principales (ACP). Le modèle correspondant étudie l'influence d'un évènement " x" sur une autre "y", par exemple l'étude de la température en fonction de la vitesse de refroidissement ou la distance parcourue en fonction du nombre de kilomètres. L'influence est modélisée par une relation de type régression

Formule 4. 1 suivante :

$$y = a * x + b \quad \text{Formule 4. 1}$$

Où

$$a = \frac{\sum_{i=1}^n (x_i - \text{moye}(x)) * (y_i - \text{moye}(y))}{\sum_{i=1}^n (x_i - \text{moye}(x))^2} \quad b = \text{moye}(y) - a * \text{moye}(x)$$

n : nombre de mesures des évènements ; x_i : occurrence de l'évènement x ; $\text{moye}(x)$: moyenne arithmétique de l'évènement x ; y_i : occurrence de l'évènement y ; $\text{moye}(y)$: moyenne arithmétique de l'évènement y .

Le modèle de la situation actuelle prend en entrée un ensemble d'évènements d'un système cible et en sortie génère une durée moyenne de la classification. La Figure 4. 2 détaille ce modèle, où chaque évènement « y » de l'ensemble est comparé avec un autre évènement « x » dans le but de prédire son comportement.

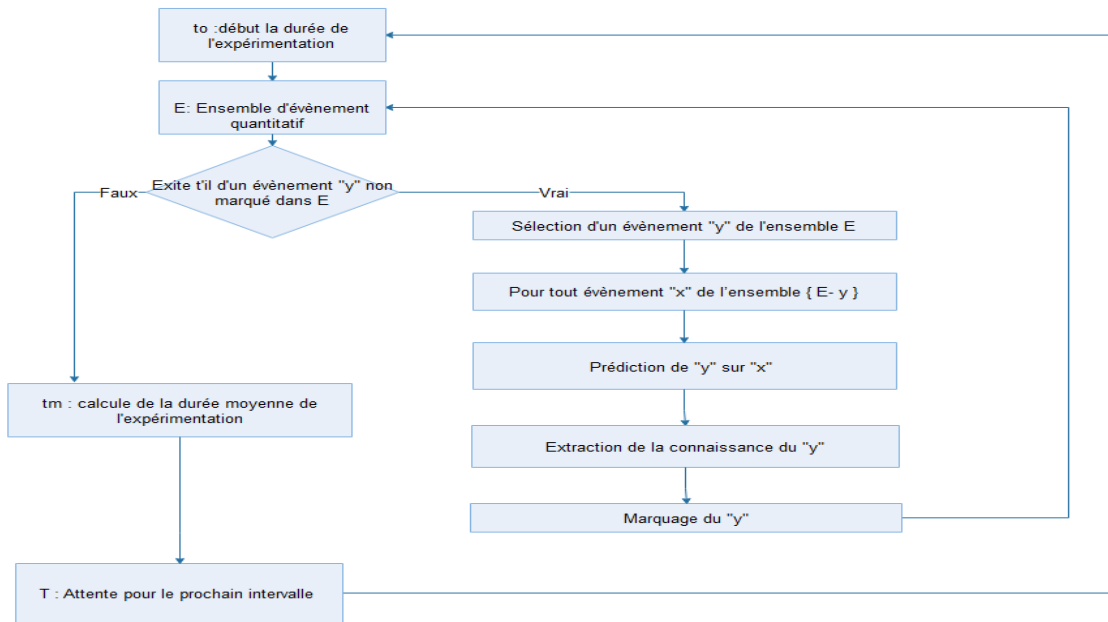


Figure 4. 2: modèle de la situation actuelle de référence

1.3. Expérimentation de la situation actuelle

Nous avons expérimenté la situation de référence dans le domaine ferroviaire sur un ensemble(E) contenant deux évènements : « vitesse de refroidissement du système traction d'un train » et « température extérieure ». L'objectif est de mesurer la performance du circuit de refroidissement des transformateurs principaux (TFP) du système de traction. Avec n : nombre de mesures des évènements = 50000 ; x : température extérieure et y : vitesse de refroidissement. L'intervalle d'attente est fixé à vingt-quatre heures (24h). La Figure 4. 3 illustre la mesure de la dispersion de ces deux évènements recensés auprès de notre partenaire sur plusieurs trains.

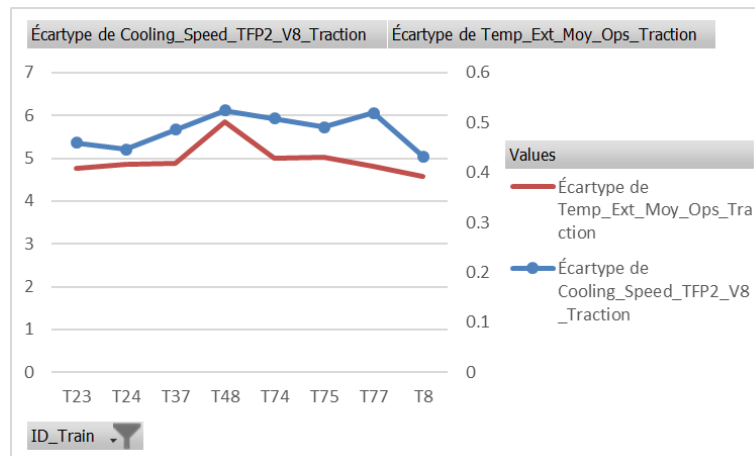


Figure 4. 3: Mesure de de la dispersion de x et y

Le déroulement de l'expérimentation est réalisé en utilisant une application appelée « fonction générique », développée en c# par notre partenaire. Actuellement, étant donné le nombre élevé d'évènements et d'information, les ingénieurs se cantonnent essentiellement à l'usage de techniques de régression linéaire pour modéliser l'influence entre deux paramètres et cette application élabore ce type de modélisation. Elle est ainsi capable d'étudier l'influence entre évènements purement quantitatifs depuis une source de données et de déterminer la prédiction

d'un évènement y en fonction d'un autre x sous la forme « $y = a * x + b$ ». La Figure 4. 4 présente l'interface de cette application. L'application est capable de réaliser la correction de données et l'étude de l'influence et les enregistrements des résultats.

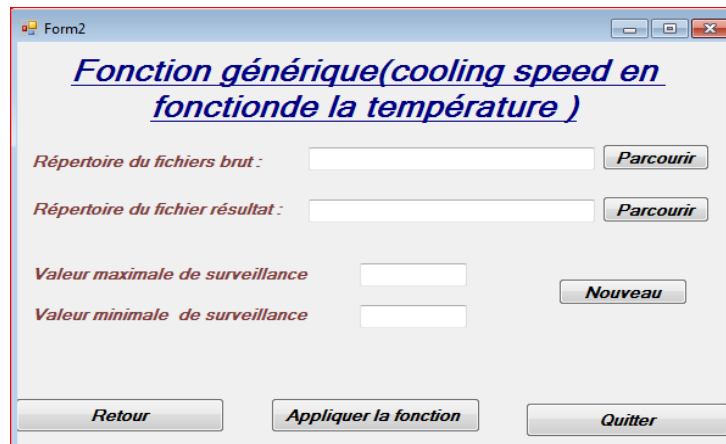


Figure 4. 4 : Application « fonction générique »

La Figure 4. 5 illustre le résultat obtenu pour la mesure de la performance du circuit de refroidissement des transformateurs du système de traction d'un train en fonction de l'évolution de la température extérieure. En utilisant la technique de régression présentée, on obtient la « meilleure » droite représentative de ces échantillons représentant les différentes occurrences des évènements, « $y = 0.074x - 3.244$ » avec $a = 0.074$ et $b = -3.244$. Cette relation mesure la dispersion entre la vitesse de refroidissement et température extérieure. L'équation de cette droite peut servir comme base de connaissance pour un système de surveillance afin de prédire le niveau de performance des transformateurs du système de traction d'un Train.

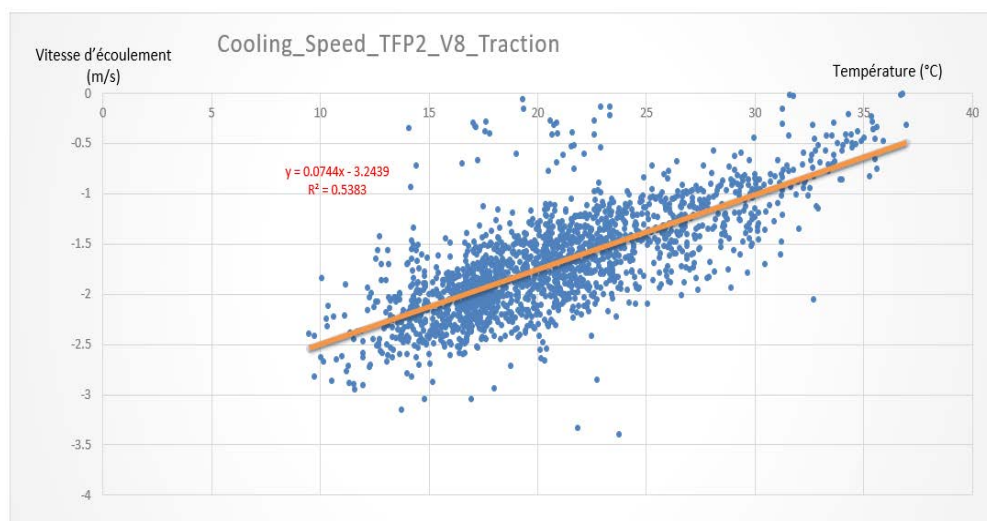


Figure 4. 5 Résultat de l'expérimentation pour la situation de référence

La durée de l'expérimentation « t » obtenue pour ces deux évènements « y et x » est d'une Minute. Nous avons effectué plusieurs itérations du modèle d'expérimentation de la situation de référence sur un ensemble (E) contenant douze (12), puis vingt-quatre (24) et enfin cinquante (50) évènements avec n « nombre de mesures des évènements » égal à 50000, ce qui correspond à une simulation d'évènements d'un système cible générés pendant vingt-quatre heures (24)). Nous avons

ainsi obtenu la durée moyenne de l'expérimentation « tm » pour un ensemble E contenant « c » évènements (Formule 4. 2), où chaque évènement doit être comparé deux à deux avec l'ensemble des autres évènements:

$$tm = t * \sum_{i=1}^{(c-2)} (c - i) \quad \text{Minutes} \quad \text{Formule 4. 2}$$

1.4. Notre Expérimentation

A partir du modèle de simulation présenté dans la section 1.1 (voir Figure 4. 1), et dans le cadre de notre expérimentation, les SurfEvents de type qualitatif ou quantitatif sont prédits et expliqués par l'algorithme C5.5 pour la construction d'arbres de décision et la génération des connaissances. L'expérimentation concerne principalement les champs nom, date, valeur appartenant à des cycles des SurfEvents.

Notre expérimentation a été implémentée avec l'outil IBM® SPSS®⁹ (Statistical Package for the Social Sciences). Il offre des modules d'analyse statistique avancée, une bibliothèque d'algorithmes d'apprentissage automatique et d'intégration avec le Big data et permet le déploiement transparent dans les applications.

L'outil IBM® SPSS® construit un arbre de décision à travers l'identification de groupes d'évènements, une identification des relations entre ces évènements et une prédiction d'évènements futurs. Il conduit à l'élaboration de modèles de classification pour la segmentation, la réduction des données et le filtrage des variables. Bien qu'il ne sera pas utilisé pour implémenter notre solution, il constitue, dans le cadre de cette expérimentation, un outil intermédiaire de mise en œuvre et d'essai des propositions présentées dans le chapitre III.

La Figure 4. 6 décrit le modèle obtenu qui prend en entrée un ensemble (E) de cycles de SurfEvents où n le nombre de mesures des évènements (cycles des SurfEvents) est fixé à 50000 et où l'intervalle de temps est fixé à vingt-quatre heures (24h). L'ensemble E des SurfEvents est constitué des SurfEvents suivants :

« MTS en UM »

« MVB-MIOSDX31V8M1-MIOSDX31V8M1-E_Q1_UM »

« MVB-MIOSDX31V1M1-MIOSDX31V1M1-E_Q_MTS »

« MVB-MIOSDX31V8M1-MIOSDX31V8M1-E_Q1_UM » et

« MVB-MIOSDX31V1M1-MIOSDX31V1M1-E_Q_MTS »

Où « MVBxxxx » concernent des relais de signalisation de défaut chauffage du système de climatisation d'une Train, et « MTS en UM » est une variable contextuelle qui contient une prédiction du comportement de ces relais signalisation. De ce fait, cette variable représente la cible à prédire dans notre l'expérimentation.

⁹ <https://www.ibm.com/fr-fr/marketplace/spss-modeler>

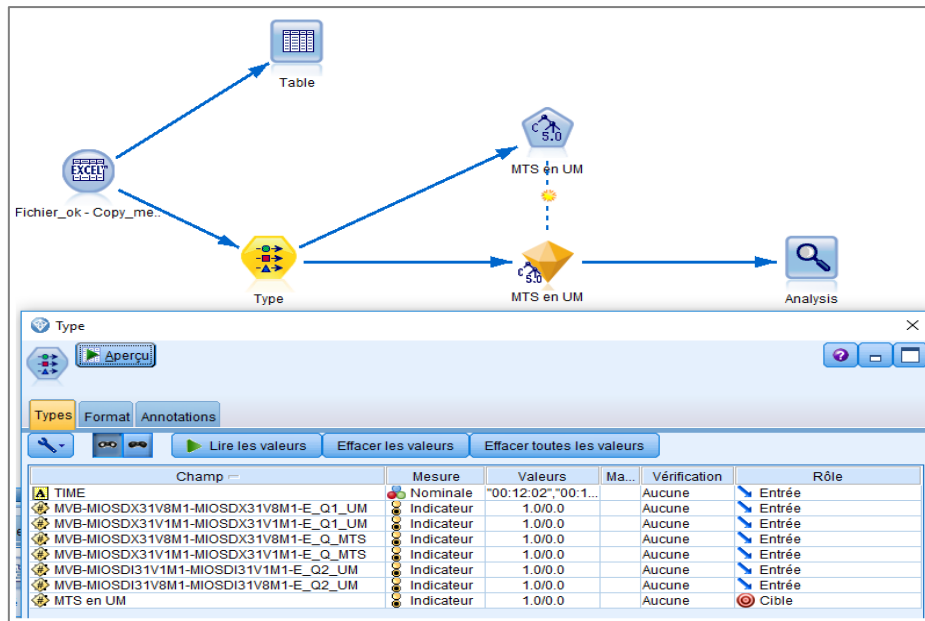


Figure 4. 6 : modèle obtenu avec l’outil IBM® SPSS®

1.5. Analyse des résultats

La durée de l’expérimentation précédente « t » est obtenue en utilisant un script écrit en python (Figure 4. 7). Elle est d’une minute et 15 secondes. Nous avons effectué plusieurs itérations de notre modèle sur un ensemble (E) de douze (12), puis vingt-quatre (24) puis cinquante (50) évènements de type SurfEvents. Le nombre n des mesures des évènements a été fixé à 5000 et à 50000. Nous avons ainsi obtenu une durée moyenne de l’expérimentation « tm » pour un ensemble E contenant « c » évènements (Formule 4. 3), qui représente le temps total pour effectuer toutes les comparaisons entre évènements:

$$tm = t * c \quad \text{Minutes} \quad \text{Formule 4. 3}$$

```
import datetime
start = datetime.datetime.now()
modeler.script.stream().runAll(None)
end = datetime.datetime.now()
t = end - start
print t
f = open('C:/Users/add/Desktop/testApp_j/export.txt', 'w')
f.write(str(t))
f.close()
```

Figure 4. 7: Script de calcul de la durée d’expérimentation

La Figure 4. 8 illustre un exemple de connaissances générées à partir du modèle réalisé, présentées sous la forme d’un arbre de décision. L’ordre hiérarchique des évènements de la racine « MTS en UM » vers les nœuds intermédiaires et les nœuds bas niveau (feuilles) est obtenu en fonction du degré d’influence des évènements sur la cible « MTS en UM ». Plus le niveau est près de la racine, plus l’évènement est dépendant de la cible. On remarque ainsi que les évènements « MVB-MIOSDX31V8M1-MIOSDX31V8M1-E_Q1_UM » et « MVB-MIOSDX31V1M1-MIOSDX31V1M1-E_Q_MTS » sont les plus influents sur la cible et donc leurs occurrences peuvent nous donner des informations précieuses sur le comportement futur des relais signalisation de défaut chauffage du système de climatisation d’un Train. Ces connaissances peuvent alors être traduites sous forme des règles de surveillance du type attribut-valeur « Si prémisse... alors Conclusion... » où le chemin

reliant une feuille à la racine de l'arbre peut être lu comme une règle de prédiction. Il est important de noter que, bien que générées « manuellement » en utilisant ici l'outil d'IBM, selon notre approche, ces règles seront générées automatiquement par notre application et seront automatiquement intégrées dans les bases de connaissances des Holons des des niveaux Holoniques « Acquisition de données », « Manipulation de données », « Détection » et « Evaluation de Santé », qui concernent ce système cible.

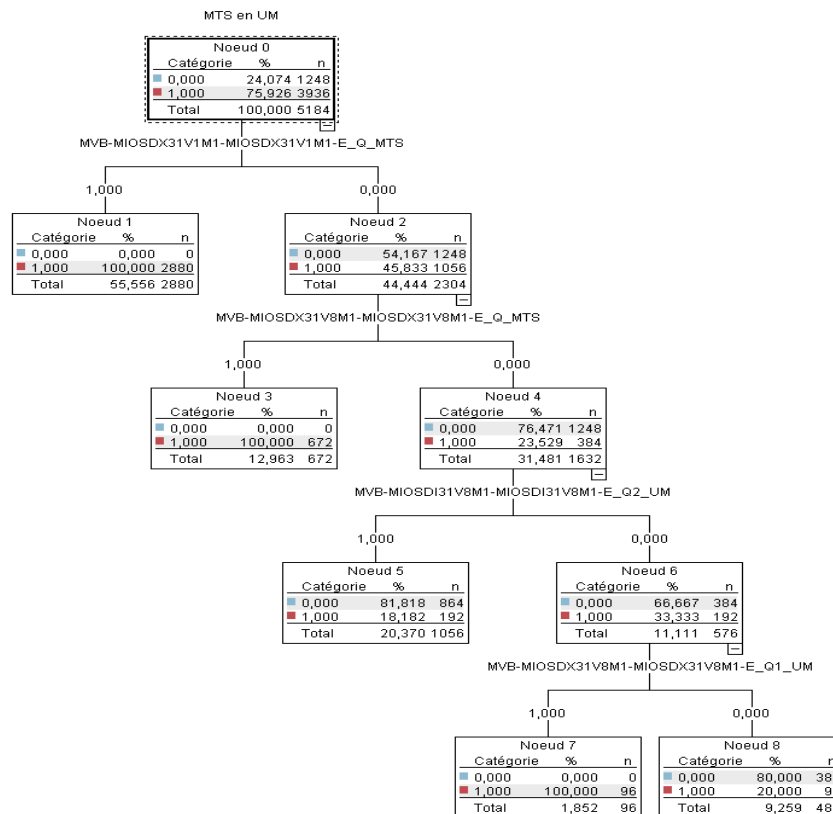


Figure 4. 8 : Arbre de décision généré à partir du modèle d'expérimentation

1.6. Impact et intérêt

Ces deux expérimentations exploitent des méthodes de modélisation basées sur les données historiques (voir la section 2.2 du chapitre I). Dans chacune de ces deux situations, une modélisation construite à partir de données historiques issues des systèmes sous surveillance permet d'associer un ensemble de mesures d'états de fonctionnement connus du système et de déterminer des changements de caractéristiques par rapport à des valeurs de référence, ou par rapport à des valeurs mesurées à des instants passés (dans une autre fenêtre temporelle).

Ces deux expérimentations illustrent la possibilité de prédire un fonctionnement. Les résultats montrent qu'il est possible de définir un mécanisme d'adaptation et de génération de connaissances.

Par comparaison avec l'expérimentation de référence, une approche de modélisation, basé sur la conception d'arbres de décision présente certains avantages.

Concernant tout d’abord la complexité des calculs menés, le modèle de référence étudie l’influence entre les événements deux à deux et à tour de rôle, et ce, quel que soit le nombre d’évènements de l’ensemble E. la durée moyenne d’expérimentation « tm » ($tm=t*i = 1(c - 2)(c-i)$ minutes) est ainsi plus élevée, comparée avec la durée moyenne d’expérimentation de notre modèle. En effet, une approche de construction par arbre de décision classifie à une itération, tous les dépendances et les relations liées à un évènement et ce, quel que soit le nombre d’évènements de l’ensemble E. La Figure 4. 9 représente le temps de calcul nécessaire pour élaborer les modèles de connaissance en utilisant les deux approches sur un mesure de 50000 évènements. La figure montre que le rythme de progression de la durée moyenne de calcul de notre modèle croit moins vite que la durée moyenne de calcul du modèle de référence. Par exemple, notre modèle est capable de déterminer les influences d’un ensemble composé de cinquante (50) évènements en moins d’un heure (57.5 minutes).

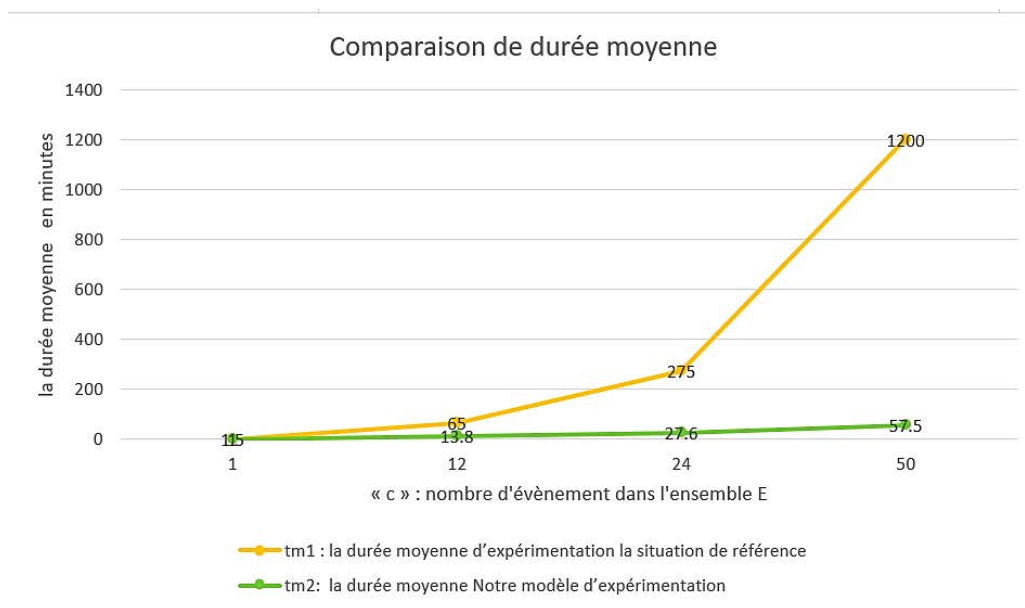


Figure 4. 9 : comparaison entre la durée moyenne de calcul pour la situation de référence et celle de notre expérimentation

Bien que cet aspect n’ait pas été évalué dans le cadre de cette expérimentation, notre approche de modélisation supporte des évènements de tout type, qualitatif et quantitatif, alors que l’approche par régression ne le permet pas. Or, dans le monde du transport, des prédictions ou des explications sur des évènements estampés de type « normal », « dégradé », « critique » par exemple sont couramment manipulées par les décideurs et ingénieurs.

Dans la section suivante, nous allons traiter des aspects déploiement et méthodologiques en lien avec notre architecture de surveillance EMH².

2. Processus de déploiement et aspects méthodologiques

Dans cette section, les processus de déploiement et aspects méthodologiques de l’architecture architecture EMH² sont présenté en trois étapes clés : le déploiement progressif, la sécurité de l’échanges de données et la formation de l’équipe en charge des opérations de maintenance.

2.1. Déploiement progressif de l’EMH²

L’architecture EMH² est très modulaire de par sa structuration Holonique. Ceci permet et facilite les déploiements progressifs sur les différents cas d’usage et applicatifs, car ce déploiement peut être

fait de manière étagée, en plusieurs itérations, sans requérir un déploiement intégral avant utilisation. Pour accompagner un tel déploiement progressif, le paradigme de l'edge-computing a été adopté pour faire de telle sorte que l'architecture EMH² implémentée soit équipée de différents nœuds de calcul « edge-computing » au fur et à mesure de son déploiement.

De manière plus détaillée, ce principe de déploiement progressif est le suivant :

- **Au niveau des systèmes cibles:** le déploiement des différents niveaux Holoniques se fait par niveau, en commençant par le niveau bas « Acquisition de données », puis « Manipulation de données », jusqu'aux plus hauts niveaux décidés. Il est mené progressivement et spécifiquement en fonction des capacités de traitement embarqué des systèmes mobiles cibles et des équipements à surveiller en fonction des contraintes des calculateurs embarqués (espace mémoire disponible, puissance de calcul, bande passante de communication ...); cette démarche, qui conduit à l'implémentation de code non intrusif pour la sécurité, offre à l'architecture de surveillance une flexibilité d'implémentation sans impacter les services fonctionnels des systèmes mobiles.
- **Au niveau des nœuds « Edge-Computing » (EC) intermédiaires:** le déploiement progressif des nœuds EC a pour but de limiter encore plus la transmission de volumes de données et d'événements entre les systèmes mobiles et avec le CM au sol. Dans notre proposition, un EC regroupe les missions des niveaux Holonique « acquisition de données », « manipulation de données », « détection » et « évaluation de santé ». Des nœuds intermédiaires entre le système mobile et le CM peuvent être ajoutés selon des critères exprimés par les exploitants de flotte, par exemple par bâtiment, par région ou par pays. L'intégration de nœuds EC supplémentaires offre également à l'architecture EMH² la capacité de redondance et de tolérance aux pannes lié au défaillance des unités de diagnostic embarqué des systèmes mobile (ce point, important pour les exploitants, n'est pas étudié plus en détail dans cette thèse).
- **Au niveau du CM:** un tel déploiement progressif facilite, en plus de garantir la continuité du service lié à la mission principale du centre de maintenance, à savoir le pronostic global et la maintenance dynamique de la flotte, la prise en charge du cycle complet d'acquisition de données, de manipulation de données, de détection et d'évaluation de santé des systèmes mobiles qui ne disposent pas, en l'état donné et au moment donné du déploiement, d'unités de diagnostic embarqués ou qui ne sont pas encore connectés aux nœuds EC.

Un exemple de déploiement progressif de l'EMH² pour quatre systèmes cibles à un instant donné est illustré Figure 4. 10. Sur cette figure, quatre systèmes mobiles présentent différents niveaux de mise en œuvre des différents niveaux Holoniques de leurs systèmes (équipement): dans le système mobile 3, trois niveaux Holoniques sont déjà implémentés alors pour son sous-système n ° 4 un seul (acquisition de données) est implémenté pour son sous-système n ° 3. Ce système mobile 3 est en outre connecté directement au CM sans utiliser pour l'instant de nœuds EC.

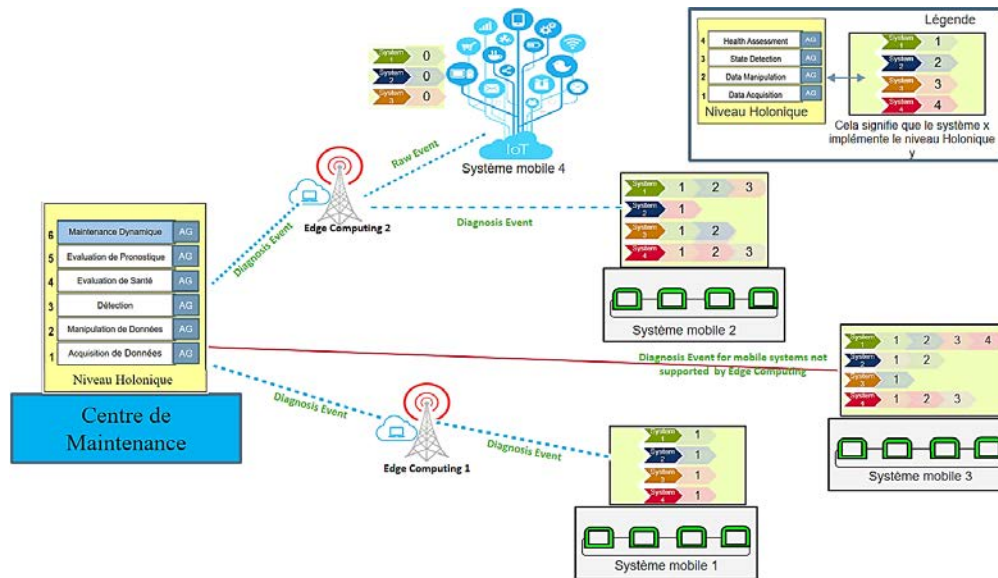


Figure 4. 10: Déploiement progressif de l'architecture EMH²

2.2. Sécurité de l'échanges de données de l'EMH²

Une des étapes clés d'un processus de déploiement réussi d'une application informatique en environnement ouvert et soumis à contraintes sécuritaires fortes comme le secteur du transport, concerne les aspects cyber-attaques et menaces externes sur les applications en question. Le déploiement de notre architecture de surveillance EMH² doit donc tenir compte de la protection de l'échange des évènements SurfEvents entre ses différents Holons et systèmes hétérogènes avec lesquels ils interopèrent. La sécurité des échanges entre les systèmes mobiles, les nœuds intermédiaires EC et le Centre de Maintenance a ainsi pour but de lutter contre la cybercriminalité [135]. L'objectif est de permettre à l'architecture EMH² d'éviter dans la mesure du possible les attaques externes, les attaques internes et les tentatives de piratage pouvant nuire les comportements des (sous-) systèmes et la sécurité des personnes. Pour ce faire, et sans travailler de manière extrêmement poussée sur ce sujet très vaste, il est supposé que pour déployer n'importe quel EMH², chaque système mobile doit disposer à minima d'un certificat signé soit par le CM ou par une tierce structure de confiance. Ce certificat lui permet de communiquer avec d'autres systèmes mobiles et le CM. Dans le cadre de notre déploiement, le standard de l'infrastructure à clé publique (PKI) [136] est proposé. Cela garantit l'authenticité, la confidentialité, l'intégrité et la disponibilité des données de surveillance et SurfEvents. Le PKI est choisi notamment en regard à une solution de type « cryptographie symétrique et asymétrique »[137] car il associe la composante physique d'un système mobile, sa localisation géographique ou son opérateur à une empreinte numérique constituée d'une paire de clés. L'approche PKI fournit également un mécanisme efficace d'authentification, de révocation, de stockage et de partage de clés, géré par une autorité de certification tierce de confiance. Dans EMH², l'implémentation de cette technologie conduit à fournir aux systèmes mobiles une paire de clés:

- **Une clé publique**, connue par tous les systèmes mobiles et systèmes hétérogènes, utilisée pour crypter et décrypter les SurfEvents lors des échanges entre les systèmes mobiles et le CM.
- **Une clé privée**, connue uniquement par le système mobile qui la détient, et qui sert à crypter, décrypter et authentifier la signature de son système mobile.

Par exemple, un système mobile crypte un SurfEvent de diagnostic qu'il envoie au CM en utilisant la clé publique de ce dernier pour assurer l'intégrité et la confidentialité de la transmission. Par conséquent, seul le CM peut décrypter le SurfEvent envoyé à l'aide de sa clé privée.

2.3. Formation de l'équipe en charge des opérations de maintenance

Le succès du déploiement de l'architecture EMH² au sein d'une organisation industrielle dépend fortement de la formation de l'équipe en charge des opérations de maintenance. Par exemple, l'architecture EMH² peut générer de manière asynchrone des conseils et recommandations par des SurfEvents de haut niveau détaillant les opérations de planification, d'inspection, de contrôle, de révision et de réparation, etc. à effectuer. Cette réactivité élevée de l'architecture constitue une innovation qui complète les méthodes de maintenance corrective déjà mises en œuvre par un opérateur de maintenance. Cependant, cette innovation peut déstabiliser l'organisation actuelle. Ainsi, pour garantir l'acceptation de cette innovation par les équipes, celles-ci doivent être correctement formées au potentiel de l'architecture EMH². Cette formation concerne l'utilisation des technologies choisies ainsi que les implémentations de tous les niveaux Holonique de l'architecture EMH².

Les objectifs sont de fournir au responsable de la maintenance la capacité de communiquer directement avec les systèmes et sous-systèmes d'un système mobile via EMH², d'étudier et d'exécuter des commandes afin d'optimiser les opérations de maintenance et de gérer le plus grand nombre possible de tâches. Il est également nécessaire de développer des outils spécifiques avec leurs interfaces homme-machine permettant aux opérateurs humains d'interagir avec EMH² : outils d'aide au dépannage, guides d'utilisation utilisant la réalité augmentée et la reconnaissance vocale pour faciliter les interactions, etc. Bien que non décrits dans nos travaux car relevant d'un cadre de réflexion traité spécifiquement au sein du laboratoire commun SurferLab, de tels outils doivent être conçus et optimisés selon une approche centrée sur l'humain afin de garantir une intégration efficace d'une « solution » EMH² au sein d'une organisation de maintenance.

3. Une implémentation possible de l'architecture EMH²

Dans cette section, et à des fins d'illustration, nous proposons une implémentation possible de l'architecture EMH². Cette implémentation se base sur le paradigme de microservice pour implémenter les Niveaux Holoniques « acquisition de données », « manipulation de données », « détection », « évaluation de la santé », « évaluation de pronostic » et « maintenance dynamique ». Le choix d'une approche par microservice a fait l'objet d'une étude comparative décrite dans l'annexe D. Cette approche vise à concevoir une solution unique comme une série de « petits » ou « micro » services où chaque service est un module autonome qui prend en charge un besoin spécifique et dispose de son propre processus d'exécution et de communication. Un microservice est facile à définir et à implémenter grâce à l'utilisation de mécanismes « légers » tels que REST (Representational State Transfer), Message Queue (MQ) et WebSocket[138]. Dans notre cadre, l'approche par microservice, comparée notamment aux architectures orientées services notamment, facilite le développement agile de solutions embarqués/au sol pour la surveillance et le diagnostic en temps réel [139], [140].

Dans la cadre de notre implémentation, le format de messages échangés entre les microservices est bien sûr construit sur le modèle SurfEvent présentés auparavant (voir la section 2 du chapitre 3). Dans la suite, seuls les microservices des différents niveaux Holoniques sont détaillés. Les autres microservices liés à l'infrastructure (microservices de configuration distribuée, d'enregistrement de service et découverte, de routage de message entre les microservices et les différentes applications hétérogènes, d'équilibrage de charge de flux des données) ne sont pas détaillés.

Les microservices de différents Niveaux Holoniques sont présentés comme suit :

- **Microservice d'acquisition de données (MAD) :** ce microservice supporte la mission du Niveau Holonique « Acquisition de données », à savoir l'acquisition et la transformation des signaux délivrés par les réseaux de capteurs et les transducteurs d'un émetteur. La transformation consiste à convertir les signaux continus en un SurfEvent de type « signal » représentant une quantité ou une qualité physique, auquel sont associés les différents champs d'un SurfEvent (datation, source émetteur du signal, par exemple). Le résultat de cette acquisition est transmis aux microservices supérieurs et/ou à des applications hétérogènes. Ce microservice peut être déployé dans les différents calculateurs de l'architecture EMH², que ce soit dans un système mobile, dans un nœud edge-computing ou dans le CM lorsqu'il n'y a pas d'unité de diagnostic embarqué dans les systèmes mobiles.
- **Microservice de manipulation (MM) :** ce microservice supporte la mission du Niveau Holonique « manipulation de données ». Il effectue une analyse sur les SurfEvents de type « signal » d'un émetteur et les convertit en SurfEvents de type « indicateur ». L'opération d'analyse comprend le calcul de descripteurs significatifs (caractéristiques) et le traitement et la transformation de ces SurfEvents. Le résultat de cette manipulation est transmis aux microservices supérieurs et/ou à des applications hétérogènes. Ce microservice peut être déployé dans les différents calculateurs de l'architecture EMH², à savoir dans un système mobile, dans un nœud edge-computing ou dans le CM lorsque l'unité de diagnostic embarqué d'un système ne supporte pas encore le microservice « manipulation de données ».
- **Microservice de détection (MD) :** ce microservice supporte la mission du Niveau Holonique « détection ». Il reçoit des SurfEvents de type « indicateur » de la part de microservice de manipulation, les différentes alarmes provenant des autres applications et des équipements du système mobile non encore pourvus de microservice de détection intégré. Ce microservice a également pour mission d'identifier des anomalies sur les SurfEvents de type « indicateur » dans le but de générer des SurfEvents de type « alarme ». Les résultats sont transmis aux microservices supérieurs et/ou à des applications hétérogènes. Ce microservice peut être déployé dans les différents calculateurs de l'architecture EMH², à savoir dans un système mobile, dans un nœud edge-computing ou dans le CM lorsque l'unité de diagnostic embarqué d'un système ne supporte pas encore le microservice « détection ».
- **Microservice d'évaluation de la santé (MES) :** ce microservice supporte la mission du Niveau Holonique « Evaluation de Santé » dans le but de déterminer l'état de santé des systèmes ou équipement. Il reçoit de SurfEvents de type « alarme » provenant du microservice de détection et d'autres applications ou équipement, et réalise le diagnostic des fautes détectées sur les SurfEvents de type « alarme ». Il détermine l'état de santé courant de l'équipement, en générant des SurfEvents de type « signalement » regroupement l'ensemble des informations d'états de santé. Les résultats d'évaluation de santé sont

transmis aux microservices supérieurs et/ou des applications hétérogènes. Ce microservice peut être déployé dans les différents calculateurs de l'architecture EMH², à savoir dans un système mobile, dans un nœud edge-computing ou dans le CM lorsque l'unité de diagnostic embarqué d'un système ne supporte pas encore le microservice « Evaluation de Santé ».

- **Microservice de pronostic local (MPL) :** ce microservice supporte la mission du Niveau Holonique « Evaluation de Pronostic » lié aux études de dépendances entre les événements. Ce microservice a pour but de classifier l'influence entre les SurfEvents de microservices inférieurs (de type signal, indicateur, alarme et signalement), et de prédire de nouveaux SurfEvents ou d'expliquer des situations qualitatives au sein d'un système mobile. Ses résultats sont transmis aux microservices « Evaluation de Santé » et au sol vers les microservices supérieurs et/ou des applications hétérogènes. Ce microservice peut être déployé dans les différents calculateurs de l'architecture EMH², à savoir dans un système mobile, dans un nœud edge-computing ou dans le CM lorsque l'unité de diagnostic embarqué d'un système mobile ne supporte pas encore le microservice « pronostic local ».
- **Microservice de pronostic global (MPG) :** ce microservice supporte la mission du Niveau Holonique « Evaluation de Pronostic » lié aux générations de SurfEvents de Haut-niveau déterminant les prédictions de durées de vie résiduelle(RUL) des systèmes sous surveillance. Ce microservice reçoit en entrée les résultats d'évaluation des différents microservices « pronostic local » et de REX de la part de microservice « maintenance dynamique », il établit une classification plus général des SurfEvents des systèmes au niveau flotte. Ses résultats d'évaluations servent à alimenter les modèles des Holons de microservice « pronostic local », les Holons du microservice « maintenance dynamique » et/ou des applications hétérogènes. Ce microservice peut être déployé uniquement dans un CM.
- **Microservice de maintenance dynamique (MMD) :** Ce microservice supporte la mission du Niveau Holonique « maintenance dynamique », dans le but de planifier, gérer les opérations de maintenance. Ce microservice s'appuie sur des SurfEvents de type signalement (de l'état de santé actuel) et Haut-niveau (état de santé prévisionnel) ainsi que sur les différents contextes opérationnels des systèmes mobiles de la flotte. Il fournit une surveillance dynamique fondée sur l'optimisation et la gestion du coût de la maintenance de la flotte. Les résultats de ce microservice sont transmis aux microservices « Evaluation Pronostic », « Evaluation de santé » et/ou à des applications hétérogènes. Ce microservice est déployé uniquement dans le CM.
- **Microservice de données (DataM) :** ce microservice offre un service de stockage, d'archivage et de gestion de volume de SurfEvents pour les différents microservices de l'architecture EMH². Il peut être déployé dans les différents calculateurs de l'architecture EMH², à savoir dans un système mobile, dans un nœud edge-computing ou dans le CM.

La Figure 4. 11 ci-dessous résume cette mise en œuvre possible. Elle présente les échanges et coopérations entre une flotte de systèmes mobiles, un système edge-computing et le CM . Elle illustre aussi un scénario d'implémentation des unités de diagnostic embarqués des systèmes mobiles de la flotte.

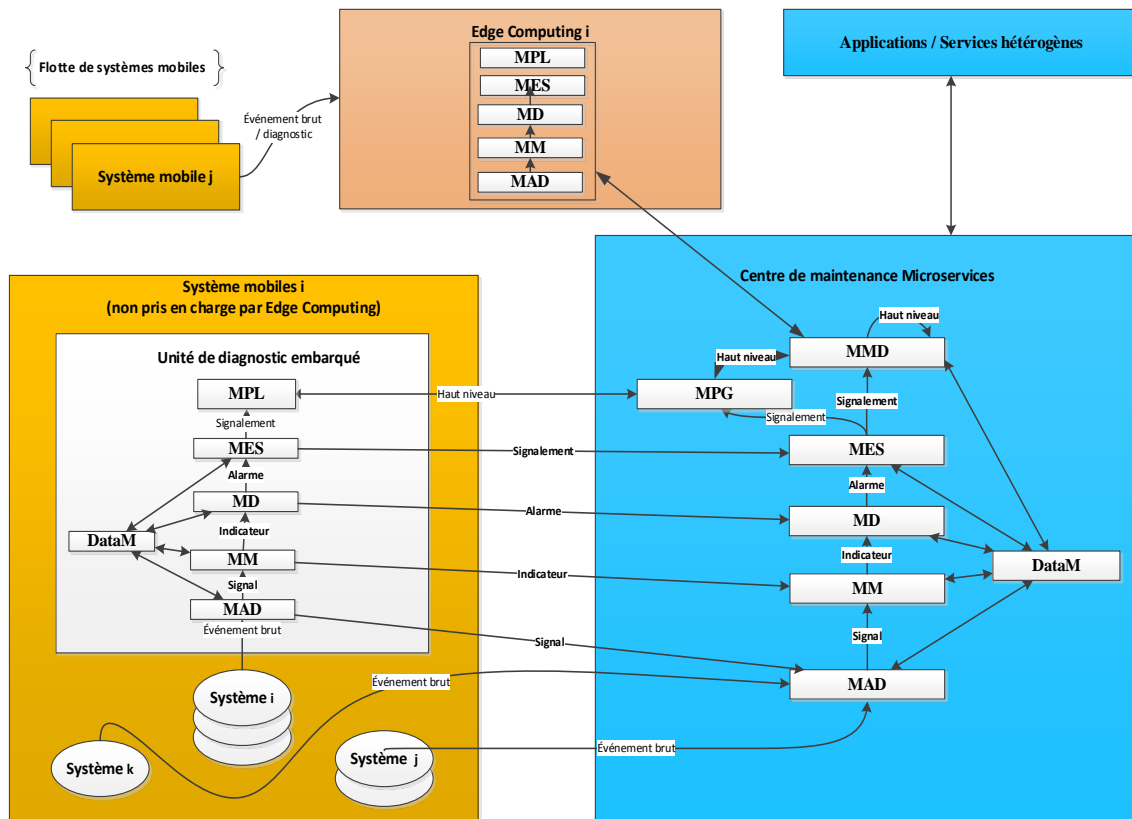


Figure 4. 11: Une Implémentation possible de l'architecture EMH² proposée en utilisant des microservices.

4. Proposition d'un guide méthodologique pour accompagner le déploiement de EMH²

Comme précisé dans cette thèse, l'architecture EMH² a été conçue de manière générique pour s'appliquer à différents types de flotte. Ainsi, même si des éléments concernant son implémentation et son intégration efficace au sein d'une organisation industrielle de maintenance ont été proposés auparavant dans ce chapitre, cela reste insuffisant pour aboutir à une implémentation complète dans la mesure où il reste nécessaire d'accompagner les décideurs dans la démarche de déploiement elle-même. A ce titre, nous avons conçu une première méthode un guide « how to ». Ce guide doit être considéré comme non abouti et largement perfectible, son élaboration complète, notamment adossée à un système d'aide informatisé constitue un sujet à part entière. Ce guide comprend 8 étapes qui sont présentés dans la suite de ce chapitre.

4.1. Évaluation de la situation actuelle pour la flotte ciblée de systèmes mobiles

L'évaluation de la situation actuelle en termes de maintenance et surveillance d'une flotte de systèmes mobile est une étape incontournable avant tout processus de déploiement d'une amélioration. Cette étape permet d'identifier les limites de la politique de maintenance actuelle de la flotte et de comprendre ses forces et ses faiblesses, en terme de technologies inappropriées notamment (communication et unités de diagnostic embarqué, par exemple). Cette situation doit être évaluée sur des indicateurs prédéfinis (KPI), par exemple, le nombre maximal de visites à ne pas dépasser durant une période donnée (voir la section 4.3.2 du chapitre 3). Lors de cette étape, les contraintes technologiques existantes et potentiellement bloquantes (bande passante, etc.) doivent être identifiées.

4.2. Définition des objectifs visés

Une fois l'étape d'évaluation de la situation actuelle réalisée, l'étape suivante consiste à définir les objectifs visés pour les KPI identifiés comme par exemple le nombre maximal de visite à ne pas dépasser durant une période donnée. Des objectifs spécifiques peuvent être définis. Par exemple, le choix des critères de décision de maintenance J_w concernant les maintenances à effectuer sur un système mobile (voir la section 4.3.2 du chapitre 3).

4.3. Définition de la structure des SurfEvents (voir la section 2 du chapitre 3)

La définition de la structure des SurfEvents constitue la première étape de déploiement à proprement parler. Cette étape sert à déterminer sa forme générique en fonction du contexte applicatif de la flotte, des objectifs (KPI) identifiés dans les étapes 1 et 2, des exigences d'interopérabilité et des technologies utilisées pour les communications. La forme générique d'un SurfEvent permet à la fois une représentation hiérarchique des événements des systèmes mobiles et un interfaçage commun pour l'échanges et la coopération entre les Holons des différents niveaux Holoniques. L'interopérabilité d'un SurfEvent permet d'établir ses protocoles de liaison avec les systèmes hétérogènes, et notamment avec l'infrastructure, ainsi qu'avec les personnes en charge des activités de design, développement, production et retrait du cycle de vie des systèmes cibles. Le mode de compression d'un SurfEvent doit également être fixé, il permet d'optimiser sa taille d'échange et de stockage dans un contexte embarqué souvent contraint.

4.4. Sélection du mode de raisonnement des Holons (voir la section 4.1 du chapitre 3)

Dans cette étape, une étude technique du fonctionnement des systèmes doit être menée afin de dresser un bilan sur le mode fonctionnement de chaque système. Ce bilan doit permettre d'attribuer à chaque système (par exemple une porte d'un train) un mode de raisonnement spécifique (par exemple par cycle ou état). Ce mode sera imposé aux différents émetteurs de ce système. Lors de la phase de diagnostic, le moteur d'inférence des Holons applique alors le mode de raisonnement choisi, afin de traiter les SurfEvents émissent.

4.5. Implémentation des microservices dans les Systèmes mobiles et nœuds Edge-computing (voir la section 2.1 du chapitre 4)

Dans cette étape, les différents Framework et plateformes logicielles d'implémentation des microservices, tels que ASP.NET Core¹⁰, Spring Cloud¹¹, flex¹², cppmicroservices¹³ et Seneca¹⁴, etc. sont discutés. Ceux qui apportent les meilleures réponses aux faiblesses technologiques identifiés lors de l'étape une, sont retenus. Leur rythme d'implémentation est validé itérativement selon la capacité résiduelle des unités de diagnostic des systèmes mobiles de la flotte qu'il est possible d'allouer pour les supporter.

La stratégie d'implémentation des noeuds edge-computing (par bâtiment, par région ou par pays) doit être validée à l'occasion de cette étape selon les études technologiques menées durant l'étape 1. Ces nœuds permettent d'augmenter les capacités de transmission de grands volumes de données au sein de EMH² et d'optimiser les redondances et mécanismes de tolérance aux pannes liés aux défaillances des systèmes de communication et des unités de diagnostic embarquées. Des

¹⁰ <https://docs.microsoft.com/fr-fr/aspnet/core/?view=aspnetcore-2.1>

¹¹ <http://projects.spring.io/spring-cloud/>

¹² <https://pypi.org/project/flex/>

¹³ <http://cppmicroservices.org/>

¹⁴ <http://senecajs.org/>

mécanismes de transport et de communication tels que REST (Representational State Transfer), Message Queue (MQ) et WebSocket doivent également être discutés et un choix effectué.

4.6. Implémentation des microservices dans le CM (voir la section 3 du chapitre 4)

Pour implémenter ces microservices le décideur dispose actuellement de deux types d'offre ou de solution ; une stratégies d'implémentation de type « On-premise », lorsque la solution est implémentée au sein de des systèmes d'information de l'exploitant de la flotte. Et une stratégie de type « Cloud », lorsque la solution est délocalisée chez un tiers et accessible à distance par réseau, moyennant un abonnement auprès de l'éditeur/hébergeur.

Ce choix est guidé par le bilan technologique de l'étape 1, mais aussi par le budget consacré par l'exploitant de la flotte des systèmes mobiles ainsi que par la prévision de l'évolution de l'architecture sur le court, moyen et long terme. Lorsque le choix se porte sur une stratégie de type « On-premise », l'exploitant de la flotte doit assurer la disponibilité d'infrastructures d'hébergement, stockage et communication fiables et disponibles. Lorsque le choix se porte plutôt sur une stratégie d'implémentation « Cloud », », l'exploitant de la flotte doit prendre en compte le niveau de service garanti par le fournisseur Cloud ainsi que son coût, non négligeable.

4.7. Formation de l'équipe en charge des opérations de maintenance (voir la section 2.3 du chapitre 4)

Dans un contexte où la maîtrise des coûts est associée à une exigence de disponibilité croissante des équipements, les mainteneurs de flotte doivent trouver l'équilibre entre les contraintes budgétaires et la qualité de service malgré les difficultés techniques. Cette étape du guide est ainsi essentielle, elle a été intégralement décrite auparavant.

4.8. Évaluation des résultats obtenus et comparaison avec les objectives visés

Dans cette dernière étape, les résultats obtenus par l'architecture EMH² désormais implémentée et utilisée sur un durée « t » sont comparés avec les objectifs généraux et KPI définis durant la deuxième étape. Sur la base de ces résultats, des réajustements peuvent être menés afin de mieux répondre aux exigences opérationnelles de la flotte des systèmes mobiles.

Ces étapes peuvent être menées séquentiellement de 1 à 8 mais certaines peuvent être menées en parallèle, comme illustré à Figure 4. 12.

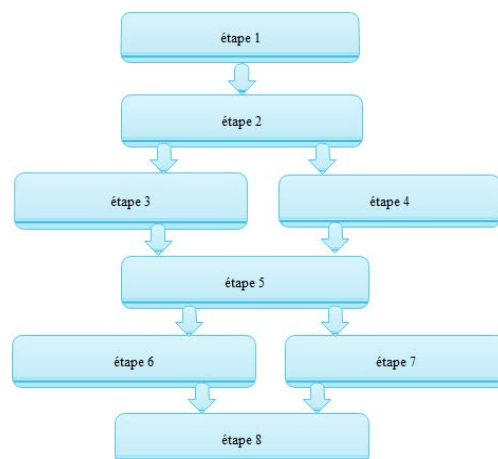


Figure 4. 12: Articulation des étapes composant le guide de déploiement proposé.

5. Conclusion

Ce chapitre, constitué de quatre grandes parties, avait pour ambition d'illustrer la faisabilité et l'applicabilité de l'architecture de surveillance EMH² proposée dans le chapitre III. Dans une première partie, une simulation simulant le mode de raisonnement des Holons « **pronostic local** » du niveau Holonique « Prognostic » a été menée pour évaluer la durée moyenne de génération de connaissances. Les résultats obtenus ont été comparés avec un modèle de référence et nous ont permis de valider, sur la base de ces simulations, notre approche basée sur un formalisme de type « arbre de décision ».

Dans la deuxième partie de ce chapitre, nous avons proposé un processus de déploiement agile de l'architecture et traité des aspects méthodologiques de l'architecture EMH². Ce processus permet un déploiement progressif au sein des différents calculateurs présents dans l'architecture, à savoir les systèmes mobiles, les nœuds edge-computing et le Centre de maintenance. La sécurité de l'échanges de données entre ces calculateurs et le besoin en formation de l'équipe en charge des opérations de maintenance ont été soulignés.

La troisième partie de ce chapitre est consacrée à l'implémentation de l'architecture à partir de la solution « microservice » afin de mettre en œuvre les Niveaux Holoniques.

La quatrième et dernière partie est consacrée à la présentation d'un premier jet d'une méthode de déploiement décrivant, sous la forme d'un guide, comment déployer EMH² en huit (8) étapes.

Ce chapitre a ainsi permis de mettre en évidence que nous traitons des spécifications cibles (voir la section 2.6 du chapitre 1) suivantes :

- **Déploiement:** le déploiement progressif de l'EMH² présenté dans (voir la section 2.1 du chapitre 4), nous a permis de concevoir une architecture opérationnelle rapidement, dès les premiers stades de sa mise en œuvre, tenant compte des limitations de transmission de grands volumes de données, du besoin en redondance et de tolérance aux pannes lié aux défaillances des systèmes de communication et des unités de diagnostic embarqué.
- **Sécurité:** le niveau de sécurité de l'EMH² présenté dans la section 2.2 du chapitre 4, intègre le standard de l'infrastructure à clé publique (PKI) dans le but de garantir l'authenticité, la confidentialité, l'intégrité et la disponibilité des données de surveillance. A cet effet, il est supposé que, pour déployer n'importe quel EMH², chaque système mobile doit avoir un certificat signé soit par le Centre de Maintenance ou par une tierce de confiance.
- **Impact organisationnel:** l'architecture EMH² répond partiellement à cette spécification (voir la section 2.3 du chapitre 4), avec l'introduction d'un outil d'aide de dépannage permettant aux responsables de la maintenance de communiquer directement avec les systèmes et sous-systèmes d'un système mobile, d'étudier et d'exécuter des commandes afin d'optimiser les opérations de maintenance et de gérer le plus grand nombre possible de tâches.

Le chapitre suivant présente une application réelle de l'architecture EMH², dans dix trains actuellement en service, en collaboration avec notre partenaire industriel, Bombardier Transport. La méthode de déploiement introduite dans la 4^{ème} partie de ce chapitre sera utilisée comme guide lors des phases de conception, de développement et d'exploitation de l'application.

Ch. V Application au domaine ferroviaire

Introduction

L'étude en simulation présentée le chapitre précédent nous a permis d'estimer la durée moyenne de génération de connaissances et de valider un aspect important de nos choix de conception relatif à l'utilisation d'arbres de décision dans le cadre du pronostic. Il a également présenté un processus de déploiement et traité d'aspects méthodologiques en considérant une implémentation possible de notre architecture à partir du paradigme de microservice. Une méthode de déploiement en huit (8) étapes a enfin été présentée dans le but de guider le processus de déploiement.

Dans la continuité de ce précédent chapitre, ce chapitre présente une application réelle de l'architecture EMH². Dans ce chapitre, cette architecture est appliquée au domaine ferroviaire et concerne dix (10) trains actuellement en service, en collaboration avec notre partenaire industriel, Bombardier Transport.

Dans la première partie de ce chapitre, la méthode de déploiement précédemment présentée dans la chapitre quatre est appliquée pour aboutir à l'implémentation complète de l'architecture de surveillance EMH². La dernière partie de ce chapitre contient une discussion globale à propos de notre étude de cas.

1. Implémentation de la solution EMH²

Comme annoncé dans l'introduction, nous appliquons le guide méthodologique précédemment présenté. Nous détaillons les huit étapes de développement qui ont été menées.

2. Évaluation de la situation actuelle pour la flotte ciblée de systèmes mobiles

Pour évaluer l'efficacité de la situation existante (et future), notre partenaire industriel a développé quatre indicateurs de performance traduisant le coût, la qualité, la prise en compte de la dimension temporelle et l'adaptabilité de l'architecture (voir la section 2.6 du chapitre 1). Ces indicateurs sont :

- **KPI # 1:** nombre de visites d'entretien de la flotte (correctives, préventives, imprévues) par semaine. Cet indicateur est une mesure des coûts de maintenance.
- **KPI # 2:** temps nécessaire à un opérateur de maintenance pour enquêter, diagnostiquer puis générer des rapports et des suivis relatifs à une opération de maintenance. Ce KPI traduit la qualité des processus de diagnostic et la rapidité de l'intervention.
- **KPI # 3:** temps requis pour mettre à jour les données concernant l'état de santé et la surveillance d'un train. Cet indicateur traduit la réactivité et la qualité de la gestion des événements lorsqu'ils se produisent.
- **KPI # 4 :** temps moyen nécessaire pour caractériser, générer les connaissances, comprendre ou organiser le processus de surveillance d'un équipement ou d'un système nouveau ou révisé. Cet indicateur traduit l'adaptabilité aux événements lorsque des systèmes nouveaux ou révisés sont introduites.

Les valeurs exactes actuelles de ces indicateurs de performance clés sont confidentielles. Cependant, des ordres de grandeur peuvent être fournis et sont indiqués dans le Tableau 5. 1. Ces indicateurs concernent l'organisation actuelle de la gestion d'événements de notre partenaire industriel.

KPI	#1	#2	#3	#4
	Nombre de visites de maintenance par semaine	Temps nécessaire pour enquêter, diagnostiquer et générer des rapports relatifs à une intervention	Temps requis par l'architecture de gestion des événements pour mettre à jour les données concernant l'état de santé	Temps moyen nécessaire à l'architecture pour organiser le processus de surveillance d'un équipement ou d'un système nouveau ou révisé.
Situation actuelle	$m > 9$ visites par semaine	≥ 45 minutes	24 Heures	03 Heures

Tableau 5. 1 : Situation actuelle pour chaque KPI

3. Définition des objectifs visés

Le partenaire industriel a défini les valeurs cibles pour les KPI en question. Ces valeurs sont données dans le Tableau 5. 2.

KPI	#1:	#2	#3	#4
Target	< 9 visites par semaine	< 30 min	< 5 Heures	< 02 Heures

Tableau 5. 2: Objectifs visés pour chaque KPI.

Ces KPIs représentent les objectifs généraux à atteindre, ils sont destinés à évaluer la performance d'une liste de systèmes considérés comme critiques pour la disponibilité des trains. Ce sont les systèmes : CVC (Chauffage, ventilation et climatisation), Accès voyageurs (Porte), Bogie, Traction et Pantographe et ce, au sein d'une flotte composée de dix (10) trains. Ces systèmes peuvent émettre différents types de défaillance(panne), à savoir des défaillances d'origine électrique, électronique, mécanique ainsi que des défaillances liées au confort des usagers. D'après une étude interne de notre partenaire, plus de 60% des pannes en exploitation proviennent de ces systèmes : 30% à 40% des pannes en exploitation proviennent des systèmes d'accès voyageurs et 5% des pannes en exploitation proviennent des systèmes de climatisation.

Les objectifs spécifiques du KPI #4 sont définis par notre partenaire sur une base de 50 évènements par système, avec un nombre de mesures d'évènements approchant 50.000 (ce qui représente le nombre de cycles générés durant vingt-quatre heures).

4. Définition de la structure de la SurfEvent

Nous avons structuré chaque SurfEvent comme suit:

- Nous avons défini une structure en concordance avec la composition systémique et hiérarchique d'une flotte de trains, où le champs « émetteur » d'un SurfEvent renseigne sur l'arborescence et les contextes de son système émetteur ainsi que la localisation de ce système dans un véhicule, train et flotte (Voir Figure 5. 1).
- Pour le format d'échange, trois formats d'échange ont été retenus : JSON ; XML et IP/MVB. Nous avons choisi Le format JSON pour l'échange d'un SurfEvent entre les différents Niveaux Holoniques et les environnements externes. Ce format offre une représentation aisée, intuitive et facilement lisible des différents champs d'un SurfEvent, accompagné d'étiquettes permettant d'en interpréter les divers sous éléments hiérarchiques, sans aucune restriction sur leurs nombres (Voir Figure 5. 1). Le format IP/MVB est un format métier spécifique d'un train (il sera détaillé dans l'étape suivante) qui nous offre la possibilité de faire l'acquisition des évènements bruts des différents systèmes du train. Le format XML est retenu afin de nous d'offrir une autre alternative d'échange notamment sous le forme de webservices avec certains environnements hétérogènes externes.
- Pour l'interopérabilité des SurfEvent, le SurfConvert a été implémenté en utilisant le protocole MQTT (Message Queuing Telemetry Transport) afin de faciliter la communication entre les différents Niveaux Holoniques et avec les systèmes externes, et notamment avec l'infrastructure, et les autres activités de design, développement, production et retrait en lien avec le cycle de vie de la flotte (Voir Figure 5. 1).

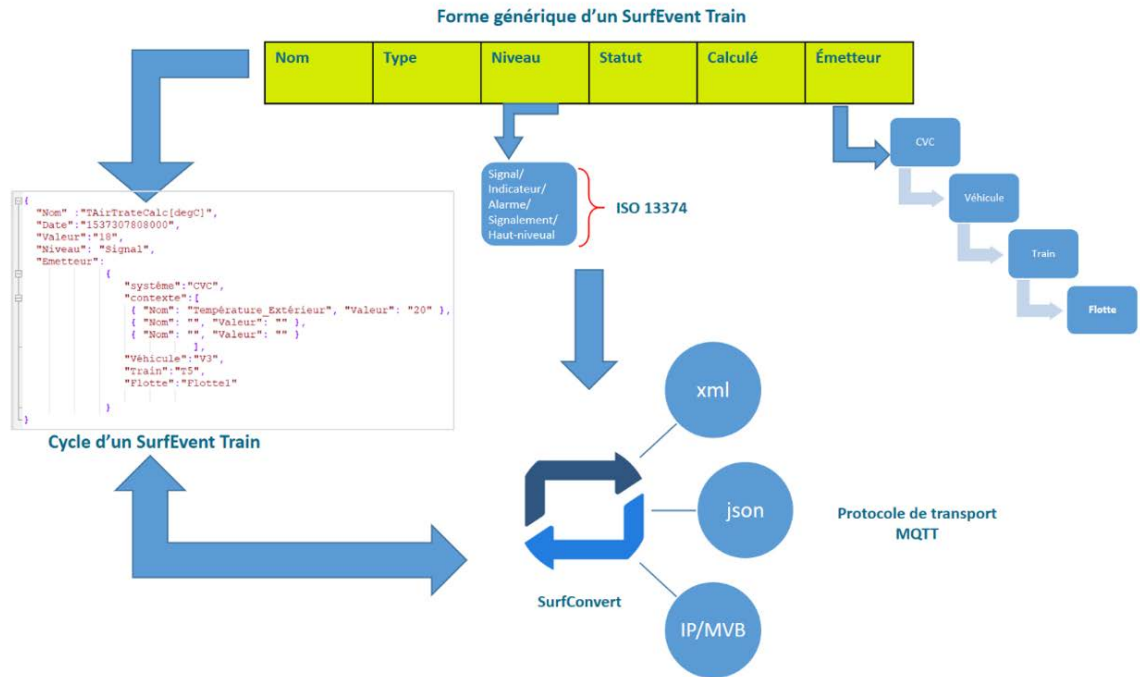


Figure 5. 1: Instanciation d'un SurfEvent pour une flotte de trains

5. Sélection du mode de raisonnement des Holons

Dans le cadre de notre expérimentation, en accord avec notre partenaire industriel, nous avons opté pour un mode de raisonnement du moteur d'inférence basé sur état, car il nécessite moins de connaissances sur le système à surveiller, contrairement à un mode basé sur cycle. Toutes les connaissances, comme expliqué et validé dans le chapitre précédent, sont modélisées en utilisant les arbres de décision. Le diagramme de séquence[141] de la Figure 5. 2 ci-dessous présente la stratégie de raisonnement correspondante mise en œuvre. Cette figure illustre les échanges et la coopération entre les Holons des Niveaux Holoniques « acquisition de données », « manipulation de données », « détection », « évaluation de Santé » et un Holon « pronostic local ».

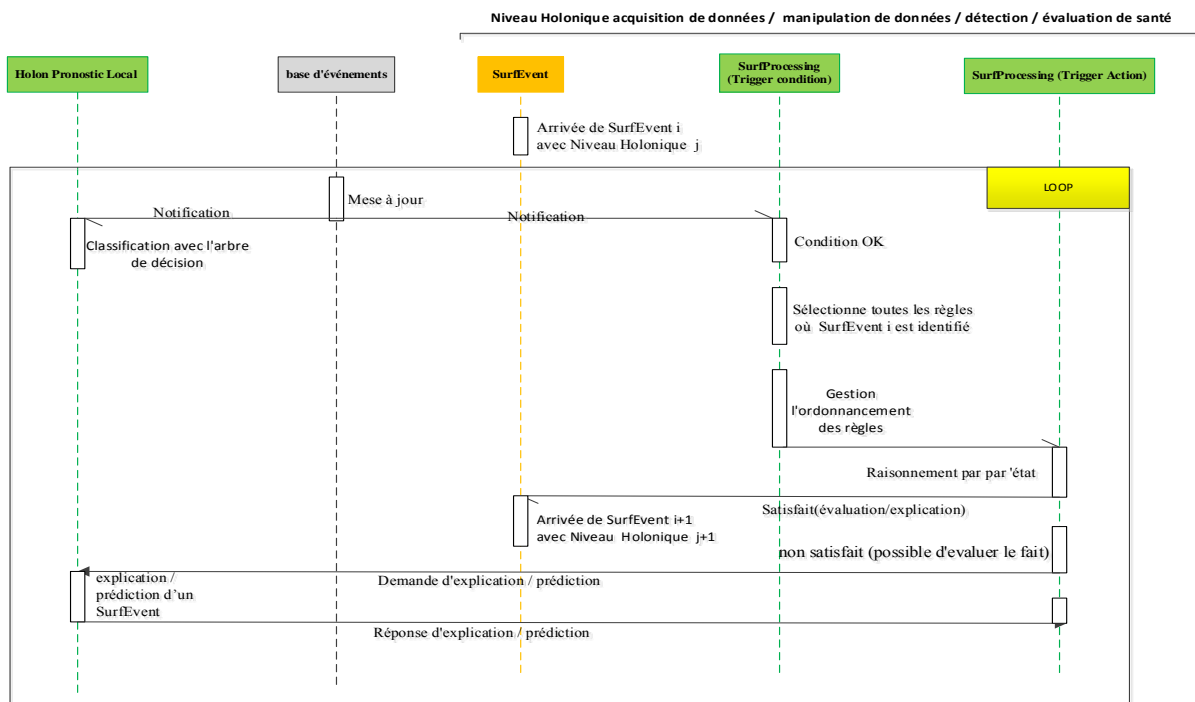


Figure 5. 2: Stratégie de raisonnement du moteur d'inférence mise en œuvre.

Le raisonnement du moteur d'inférence des Holons est accompli à partir d'une base de connaissances. Le Tableau 5. 3 fournit un échantillon de cette base de connaissances pour le système CVC, utilisant la grammaire de type 2.

SurfEvent	Type	Niveau	Statut	Expression de champ "Calculé"	Emetteur	Commentaire
Puissance	quantitatif	Indicateur	Production	(TAirTraiteCalc[degC]?)-(TAirMelange[degC]?)	CVC. Vehicule .Train. Flotte	Calculer la différence entre « TAirTraiteCalc[degC] » et « TAirMelange[degC] »
Detect_seuil	qualitatif	Alarme	Production	lif ((Puissance >10), True, null)	CVC. Vehicule .Train. Flotte	Si la «puissance» dépasse le seuil «10», la valeur de détection est «True».
Puissance_Alarme	qualitatif	Signalement	teste	lif(surfeventcount(' Detect_seuil ',BackToDate('One_Day', LocalDateTimeNow()) , LocalDateTimeNow()) > 3, True,null)	CVC. Vehicule .Train. Flotte	S'il y a un nombre de dépassements de seuil supérieur à 3 fois pendant une durée de 24h, une alarme est déclenchée.

Tableau 5. 3: Un échantillon de la base de connaissances des holons¹⁵

Comme le montre le Tableau 5. 3, un SurfEvent « Puissance » de niveau "indicateur" est calculé à partir de la différence entre les SurfEvents « TAirTraiteCalc [degC] » et « TAirMelange [degC] ». Tous les deux sont des "signaux" générés directement par des capteurs ou des transducteurs. Ce SurfEvent « puissance » déclenche une alarme active « Detect_seuil » de niveau "alarme" lorsque sa valeur dépasse le seuil "10" [degC]. Un SurfEvent « Puissance_Alarme » de niveau "Signalement" est déclenché si « Detect_seuil » a été déclenché plus de 3 fois au cours des vingt-quatre (24) dernières heures ("One_Day"). Le SurfEvent « Puissance_Alarme » peut être associé ou non à d'autres SurfEvents du même niveau ou de niveaux inférieurs afin de déterminer les conditions de santé ou les futurs modes de défaillance qui seront utilisés pour planifier les opérations de maintenance.

¹⁵ surfeventcount: compte le nombre d'occurrences d'un SurfEvent entre deux dates ;
BackToDate: renvoie la période de temps entre deux dates

6. Implémentation des microservices dans les Systèmes mobiles et nœuds Edge-computing

Dans le cadre de cette expérimentation, seuls les microservices des systèmes mobiles ont été déployés. En effet, le degré de maturité de la technologie edge-computing n'a pas été jugée suffisante pour l'instant par notre partenaire industriel, nous n'avons donc pas déployé de nœuds de ce type (cependant, cela constitue une évolution à court terme de notre application). Ce déploiement a donc été fait exclusivement dans des unités de diagnostic embarqués (calculateurs), et cela en fonction des capacités de calcul et de mémorisation de ces dernières.

L'implémentation des microservices des niveaux Holoniques « acquisition de données », « manipulation », « détection », « évaluation de la santé » et des holons de « pronostic local », a été réalisé avec la plateforme « cppmicroservices ». Cette plateforme a été développée en C++, elle offre des services dynamiques en matière de conception, construction, gestion de microservice complexes, de réutilisation des composants et de configuration des systèmes déployés.

Chaque calculateur embarqué dispose de deux interfaces de communication. L'une concerne l'interface d'échange et de coopération avec les différents microservices de l'architecture EMH² via le protocole MQTT (voir l'étape 04) ; La seconde interface est connectée aux réseaux de diagnostic des systèmes de Trains à travers des protocoles TCN (Train Communication Network) et ETB (Ethernet Train Backbone). Par exemple, le CVC est accessible via le protocole ETB et les Bogies, via le protocole MVB.

Le protocole TCN est une combinaison hiérarchique de deux bus filaires pour la transmission de données dans les trains (voir Figure 5. 3). Il se compose d'un bus de véhicule MVB à l'intérieur de chaque véhicule et d'un bus WTB (Wire Train Bus) pour connecter les différents véhicules[142]. Le bus MVB utilise les interfaces RS485 afin d'être interconnectés avec plusieurs circuits émetteurs et récepteurs, utilisés dans des systèmes multipoints en mode symétrique (Bus 2 fils).

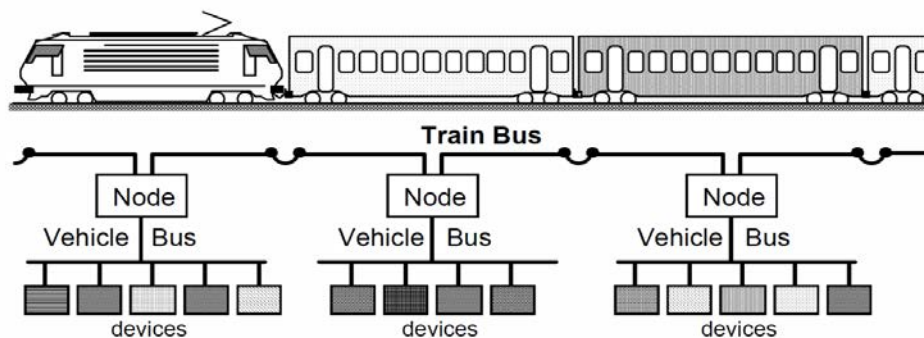


Figure 5. 3: protocole TCN d'un Train

Le protocole ETB est quant à lui, un réseau de communication basé sur la technologie Ethernet[143]. Il s'agit d'un réseau dorsal de communication à l'échelle du train, comme le bus filaire dans le réseau de communication de train (voir Figure 5. 4). Le protocole ETB utilise l'adressage IP (Internet Protocol) des réseaux afin de communiquer avec les différents systèmes.

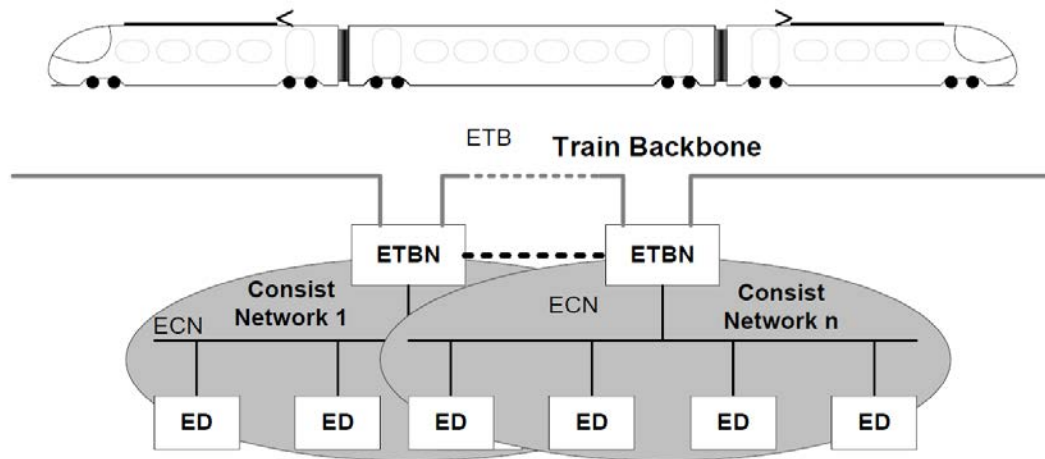


Figure 5. 4: protocole ETB d'un Train

La figure Figure 5. 5 reprend et spécifie dans notre contexte applicatif la *Figure 3. 8* du chapitre III. Elle procure l'architecture de surveillance embarquée par groupe d'émetteurs des systèmes CVC, Porte, Bogie, Traction et Pantographe des 10 trains. Cette surveillance est réalisée avec les microservices « acquisition de données », « manipulation », « détection », « évaluation de la santé » et « pronostic local », disposant des interfaçages bas niveau avec les équipements, via les réseaux de diagnostic des systèmes IP/MVB et des interfaçages haut niveau avec l'infrastructure de l'architecture EMH², via avec le MQTT. Le résultat de la surveillance (événements de diagnostic) transite vers le sol via des MCG (Mobile Communication Gateways).

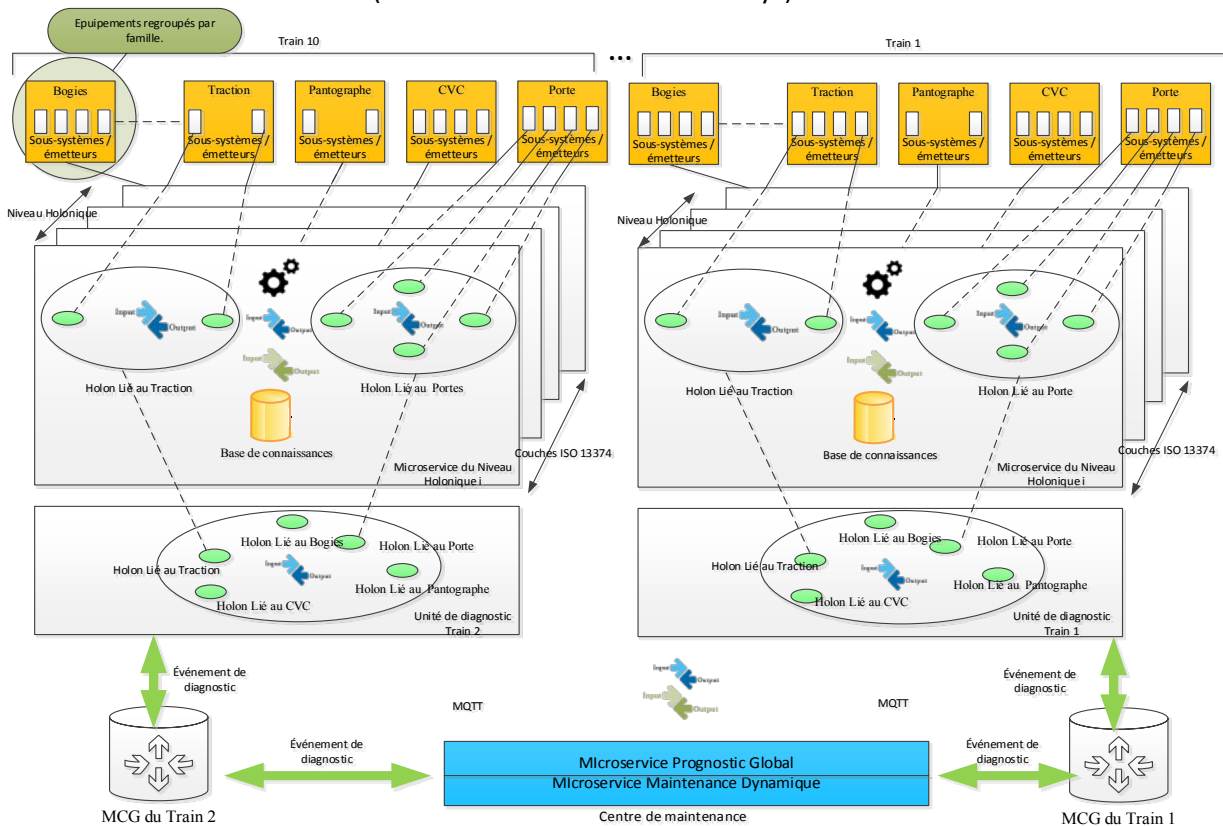


Figure 5. 5: microservices d'implémentation pour les 10 trains

7. Implémentation des microservices dans le Centre de maintenance

Dans le cadre de notre expérimentation, nous avons opté pour la stratégie d'implémentation et de déploiement « On-premise » au détriment d'une stratégie « cloud » car cette première stratégie propose une capacité d'hébergement, de stockage et de communication adaptée pour les différents microservices du Centre de Maintenance. Ces derniers ont été implémentés comme suit :

- Les microservices « acquisition de données », « manipulation », « détection », « évaluation de la santé », « pronostic local », « pronostic global » et « maintenance dynamique » ont été développés en C# à l'aide de la plateforme Microsoft ASP.NET Core. Cette plateforme est modulaire, agile et open source¹⁶, elle offre des avantages en terme d'unification de création des interface utilisateur web et des API web (application programming interface), d'intégration des workflows de développement, de gestions de versions des applications et de capacité à générer et à exécuter sur différentes systèmes (Windows, macOS et Linux).
- Le microservice « données », décrit dans la section 3 du chapitre précédant, intègre les systèmes de gestion de données SQL Relationnel, No SQL et Time Series Database (TSDB).

Conformément au guide méthodologique, plusieurs interfaces utilisateurs ont été réalisées au niveau notamment du centre de maintenance. Ces interfaces offrent aux différents acteurs de l'architecture EMH², notamment les exploitants, opérateurs de maintenance, constructeurs, équipementiers, ingénieurs système et systèmes externes, des vues intuitives et standards qui leur permettent d'interagir avec n'importe quel holons de l'architecture. Par exemple, la

Figure 5. 6 ci-après, illustre une vue d'édition de la base de connaissance des holon du système CVC, pour les ingénieurs système. L'interface offre également aux ingénieurs des outils d'analyse syntaxique et sémantique lors de l'édition de nouvelles connaissances conformément à la grammaire et à l'automate proposés 4.1 du chapitre III. L'ingénieur utilise ensuite un simulateur pour tester le comportement de la connaissance et visualiser les résultats sous forme graphique. Le statut de cette connaissance peut alors être basculé de « test » à « production » dans le ou les Holons correspondants.

¹⁶ <https://github.com/aspnet/AspNetCore>

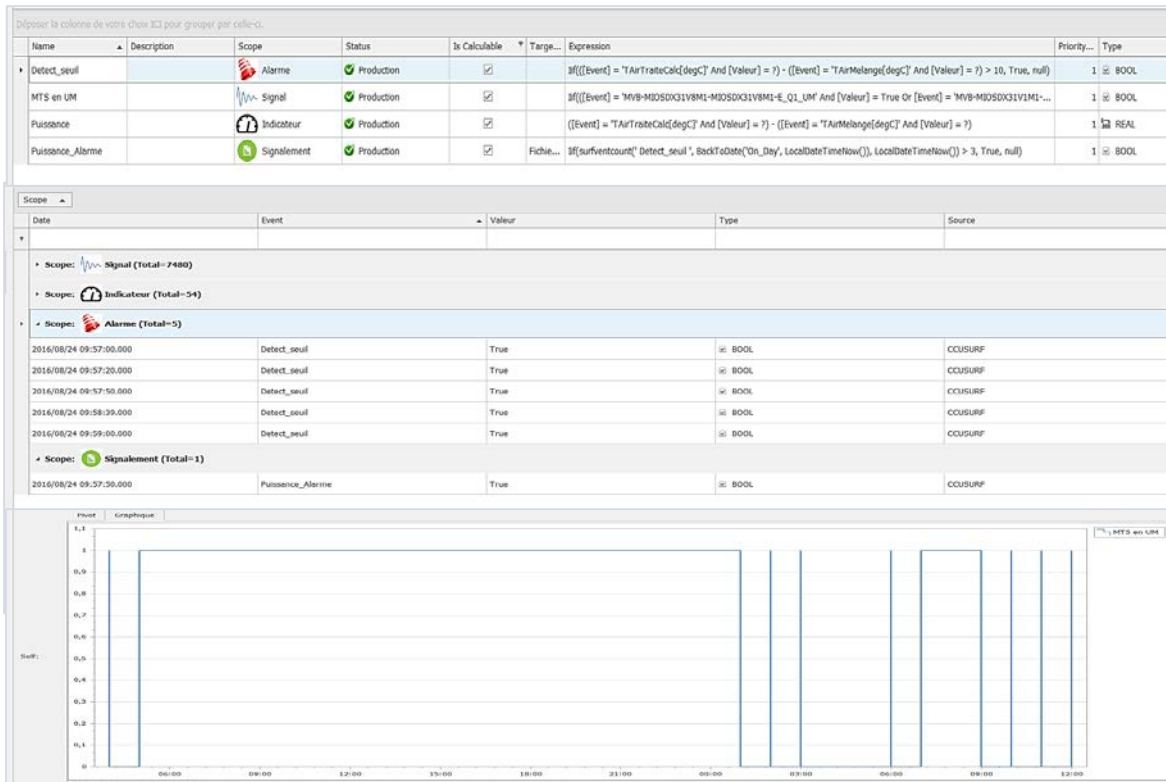


Figure 5. 6: Editeur de connaissances de l'architecture EMH²

8. Formation des équipes en charge des opérations de maintenance

Des sessions de formation ont été organisées pour faciliter l'appropriation de la solution EMH² mise en œuvre par l'équipe en charge des opérations de maintenance. Afin de les aider et d'éviter les erreurs de manipulation, un outil d'aide au dépannage a été conçu en collaboration avec l'équipe. Cet outil est connecté directement au Centre de Maintenance. Lors d'une inspection ou d'une opération de réparation, l'agent de maintenance connecte physiquement cet outil au réseau ferroviaire. Avec cet outil, il peut communiquer directement avec un équipement d'un train ou un sous-système via l'holon correspondant, ainsi que pour investiguer et exécuter des commandes. Par exemple, lors d'une opération d'inspection et de réparation sur le système CVC (HVAC1, adresse IP: 10.0.0.72) du train z5500659 (voir Figure 5. 7), l'agent de maintenance choisit d'abord le type de réseau (IP ou MVB) et le numéro de véhicule (v11), afin qu'il puisse lancer une session d'investigation sur ce matériel et recueillir les SurfEvents pertinents. La Figure 5. 7 présente l'interface utilisateur de l'outil.

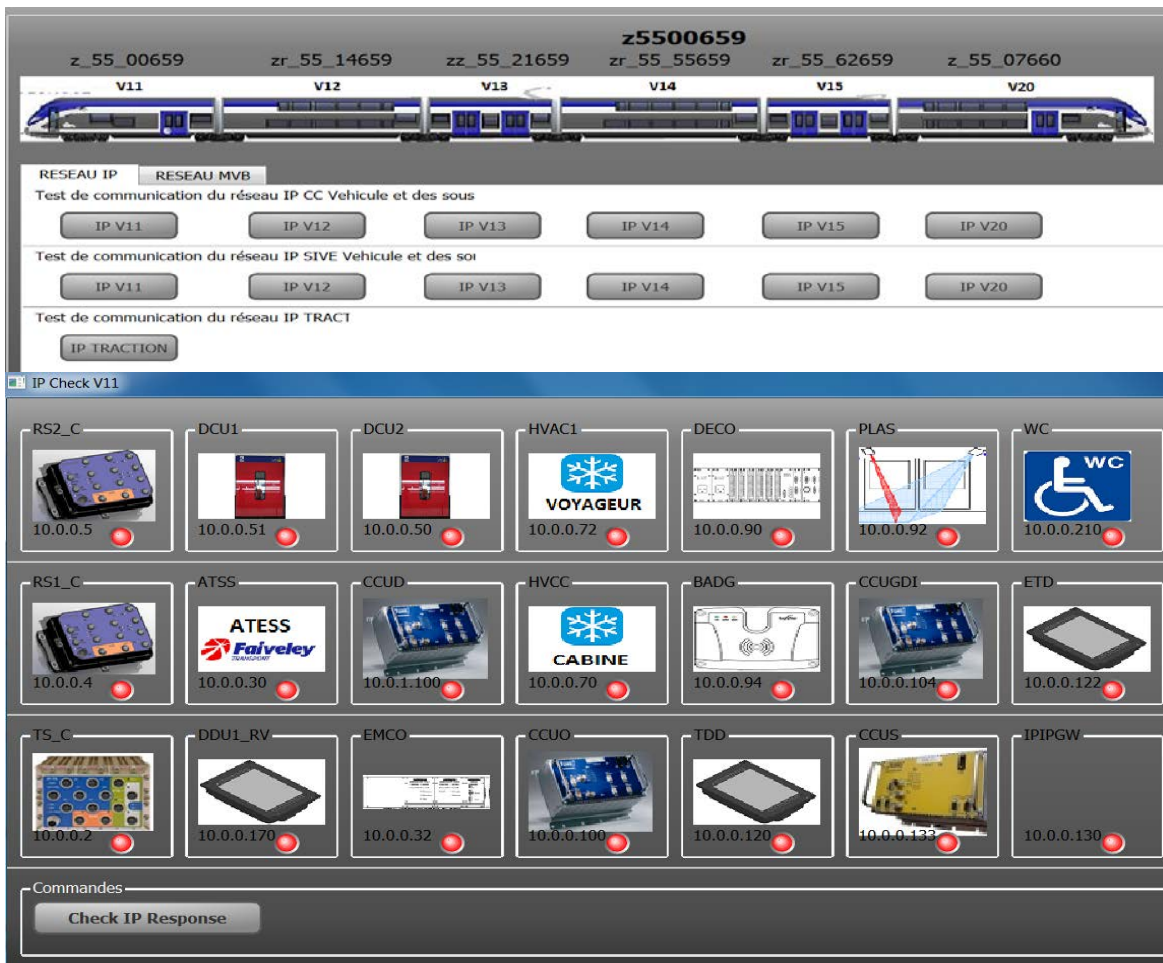


Figure 5. 7: Outil d'aide au dépannage: interface utilisateur

9. Évaluation des résultats obtenus et comparaison avec les objectifs visés.

L'expérience a été réalisée en 2016. Après plusieurs semaines d'utilisation, le partenaire industriel a pu évaluer l'impact réel de l'architecture EMH² sur le KPI introduit précédemment.

Tableau 5. 4 présente une borne sur les résultats obtenus pour chaque indicateur de performance clé (pour des raisons de confidentialité, les données exactes ne peuvent être fournies). Pour certains indicateurs, l'objectif a été clairement dépassé.

KPI	#1:	#2	#3	#4
Résultats obtenus	< 9 visites par semaine	< 20 min	< 2 Heures	< 1,5 heures

Tableau 5. 4: Résultats atteints

Quelques commentaires et retours d'expériences concernant les performances de l'architecture EMH² sont présentés ci-après:

- KPI#1: Le cycle de vie complet d'un SurfEvent est efficacement sous contrôle, de sa génération aux tâches de maintenance finale. Un calcul détaillé de cet indicateur de performance clé sur plusieurs semaines (mode glissant) est présenté à la Figure 5. 8 . Comme on peut le constater, le nombre maximal de visites, qui est de huit, n'a été rencontré qu'une seule fois. En règle générale, une seule visite était effectuée par semaine.
- KPI#2: Aujourd'hui, grâce à l'outil d'aide au dépannage, toutes les tâches sont automatisées. Cela explique la réduction de la durée moyenne des interventions à moins de 20 minutes, principalement en raison de mécanismes de contrôle direct et manuel des systèmes concernés et du partage des retours d'expérience de l'ensemble de la flotte.
- KPI#3: Jusqu'à présent, le cycle de réactivité a été réduit à moins de 2 heures. Cela est dû à la capacité de l'architecture EMH² à assurer une réactivité globale, de toute la chaîne de maintenance, incluant les phases de supervision, de diagnostic, de planification et d'optimisation des opérations de maintenance.
- KPI# : la durée moyenne d'adaptation et de génération de connaissances des systèmes a été réduit à moins de 1,5 heures. Bien que l'objectif de moins de 2 heures soit atteint, cette durée reste significativement plus grande par rapport à la durée observée lors de la simulation (voir section 1 du chapitre 4). Cela est dû au temps nécessaire pour effectuer toutes les actions de lecture/écriture et de préparation des SurfEvents, depuis le support physique de mémorisation des unités de diagnostic embarqué des systèmes mobiles jusqu'au CM.

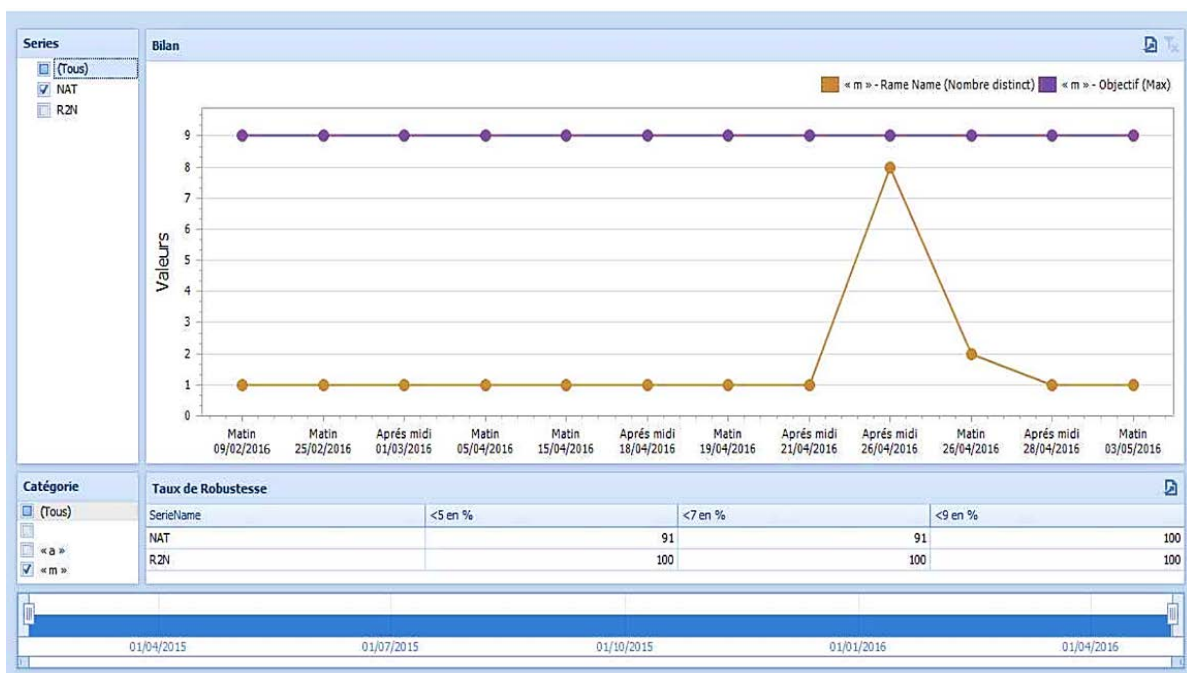


Figure 5. 8: Evolution du KPI#1 sur plusieurs semaines

10. Discussion globale sur notre étude de cas

Les ingénieurs système du partenaire industriel se sont rapidement adaptés à l'architecture EMH². La modularité et la réactivité de l'architecture ont facilité la surveillance continue de la flotte de trains. Des connaissances nouvelles ont facilement pu être identifiées et ajoutées à différents Holons.

D'après les feedbacks qualitatifs relatifs à cette étude de cas, l'architecture EMH² a été en mesure d'accélérer, de faciliter, de numériser et d'automatiser les processus de surveillance des équipements cibles dans un train et de traiter efficacement le niveau de la flotte. Le partenaire industriel est convaincu que la solution EMH² peut être appliquée à d'autres modes de transport et à d'autres domaines industriels : centrales nucléaires, systèmes de fabrication ou bâtiments intelligents.

Cette expérimentation a toutefois permis d'identifier deux limites à notre contribution:

- Tout d'abord, au cours de l'expérience, nous avons constaté un manque d'efficacité du processus d'adaptation des Holons « pronostic global », qui devrait conduire à de nouvelles connaissances précises concernant un équipement nouvellement intégré et faciliter l'intégration au sein de l'architecture EMH² ces nouvelles connaissances. Par exemple, changer une porte nécessite la mise à jour de la connaissance de son holon de surveillance. Le module d'adaptation est en cours de développement pour résoudre ce type de problème et générer automatiquement les connaissances pertinentes.
- Deuxièmement, bien que dans cette étude de cas, les indicateurs de performance clés ont été améliorés, la validation complète et l'optimisation de l'EMH² doivent encore être effectuées par le biais de simulations complémentaires et d'implémentation à l'échelle réelle d'une flotte (une centaine de trains), ce passage à l'échelle n'ayant pas été testé dans le cadre de nos travaux ni sur d'autres moyens de transports. Une seule étude de cas ne permet pas de tirer de conclusions globales quant à l'efficacité de la solution EMH² pour tout type de flotte de système mobile.

Conclusion générale

Les travaux réalisés dans le cadre de cette thèse ont porté sur la proposition d'une architecture de surveillance pour l'aide à la maintenance d'une flotte de systèmes mobiles. Cette architecture est appelée EMH², destinée à faciliter le diagnostic et le suivi d'une flotte de systèmes mobiles, exploite les principes holoniques, des plus bas (capteurs) aux plus hauts niveaux (ensemble d'une flotte de systèmes mobiles).

Le premier chapitre avait pour ambition de fixer le cadre de nos travaux ainsi que la terminologie générale relatives aux systèmes complexes, au cycle de vie de ces systèmes, à leur maintenance et leur surveillance. Nous avons également borné le cadre de notre étude. Nous avons notamment limité notre champs d'intervention sur la stratégie de maintenance proactive et notre propos est relatif aux architectures de surveillance permettant de mettre en œuvre une telle stratégie de maintenance. Le premier chapitre a notamment permis d'introduire les spécificités des systèmes cibles considérés dans nos travaux et de présenter les problématiques et enjeux principaux rencontrés par les industriels et les opérateurs de flottes de systèmes mobiles puis d'introduire les indicateurs recherchés. Ce chapitre se conclue par la mise en évidence d'une problématique globale que nous avons traduite en plusieurs spécifications à respecter.

Le second chapitre s'est focalisé sur les architectures de surveillance de systèmes mobiles en dressant tout d'abord un état de l'art des approches de la littérature. Différentes catégories d'architectures de surveillance ont été définies et deux catégories d'architecture complémentaires, à savoir « Décentralisée et coopératives », et « Edge-centralisée » ont été choisie, sur la base des spécifications cibles établies dans le premier chapitre.

Le troisième chapitre fut consacré à la description de notre proposition : une architecture de surveillance Holonique pour l'aide à la maintenance d'une flotte de systèmes mobiles, basée sur une version améliorée d'EMH, appelée EMH². EMH² généralise l'utilisation de principes HOLONIQUES précédemment proposés dans l'architecture EMH, typiquement la décomposition récursive d'Holons coopératifs. EMH² est différente dans la mesure où elle est notamment conçue pour être applicable à tout type de système mobile ou d'équipement jusqu'au niveau de la flotte, alors que l'architecture EMH était applicative, limitée au niveau d'un élément de la flotte et était dédiée à la surveillance d'un système porte d'un train. EMH² repose en outre sur le concept innovant d'un modèle d'événement unifié, appelé « SurfEvent ». Son architecture par « Niveaux HOLONIQUES », est inspirée de la norme ISO 13374 et adaptée au contexte d'une flotte de systèmes mobiles. L'objectif pour EMH² est de proposer un traitement automatisé de la surveillance plus modulaire, réactif et adaptatif tout en gérant et en optimisant les coûts de maintenance.

Le quatrième chapitre avait pour objet de montrer la faisabilité technique et l'applicabilité de notre proposition en fonction des solutions technologiques actuellement disponibles. La première partie de ce chapitre fut consacrée à une simulation du niveau HOLONIQUE « Evaluation de Pronostic » pour valider le choix fondamental dans notre approche du modèle de connaissances des Holons « **pronostic local** ». La simulation a permis d'estimer cette faisabilité en calculant la durée moyenne de génération de connaissances et d'intégration les retours d'expérience dans le processus de surveillance. Le quatrième chapitre contient également un processus de déploiement et traite d'aspects méthodologiques de sécurité et d'adaptation, destiné aux différents entités de

l'architecture EMH², à savoir, les systèmes mobiles, les nœuds intermédiaires de calcul « edge-computing » et le Centre de Maintenance. Le paradigme de microservice a été choisi pour mettre en œuvre les différents Niveaux Holoniques de l'architecture. Ce chapitre se termine par la présentation d'un guide méthodologique pour aider au déploiement de l'architecture EMH².

Dans le dernier chapitre, l'architecture de surveillance EMH² proposée a été appliquée au domaine ferroviaire sur une flotte de dix trains en service, en collaboration avec notre partenaire industriel, Bombardier Transport. La méthode de déploiement précédemment présentée dans le chapitre quatre a été utilisée ici comme guide pour l'implémentation de cette architecture de surveillance EMH². Les bénéfices de notre approche ont été mis en évidence sur quatre indicateurs clés. Les résultats de cette étude, bien qu'obtenus spécifiquement et uniquement dans ce cas, nous incitent à poursuivre nos travaux et à appliquer EMH² dans des domaines industriels relevant du transport ou non.

De notre point de vue, la contribution scientifique est triple : la proposition d'un modèle générique d'événement SurfEvent, une architecture de surveillance générique construite sur les paradigmes holoniques et une modélisation spécifique des différents niveaux holoniques et holons la composant. La contribution technique se situe sur la conception d'une architecture logicielle complète mettant en œuvre cette architecture et sur son application réussie sur une flotte de 10 trains. Notre ambition est que l'architecture EMH² accélère, facilite, numérise et automatise les processus de diagnostic et de surveillance de différents types d'équipements à l'intérieur d'un système mobile et permette de traiter de la maintenance au niveau de la flotte.

Perspectives de Recherche

Les résultats très prometteurs obtenus et les feedbacks de notre partenaire dans le cadre de notre étude de cas, nous incitent à poursuivre le développement 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 l'amélioration de l'efficacité du processus d'adaptation des Holons « pronostic global » ayant pour mission de prendre en charge au sein de l'architecture EMH² l'intégration de nouvelles connaissances concernant un équipement nouvellement intégré. Un autre point concerne la mise en œuvre de nouvelles simulations pour valider spécifiquement d'autres aspects de notre proposition. Dans cette thèse, une seule simulation a été proposée, l'objectif est alors de valider complètement l'architecture par ces simulations. Les aspects « optimisation », mentionnés à deux reprises dans notre document n'ont pas également été traités comme ils le devraient. Ce point ouvre des perspectives intéressantes concernant de nouveaux problèmes d'optimisation en maintenance.

Les perspectives à moyen terme concernent l'application la solution EMH² à d'autres modes de transport et aux domaines industriels, typiquement des centrales nucléaires, des systèmes de fabrication ou des bâtiments intelligents. Ces applications serviront non seulement à tester la généralité et l'applicabilité de notre architecture, mais aussi tester la pertinence de notre guide de déploiement qu'il reste à informatiser par ailleurs. Un autre point que cela nous obligera à considérer est relatif au fait que la définition des objectifs généraux et spécifiques à atteindre pour une politique de maintenance d'une flotte change d'un domaine à un autre, en fonction de degré de

sûreté de fonctionnement et des exigences en terme de maintenabilité, fiabilité sécurité et disponibilité¹⁷.

Les perspectives à long terme s'articulent autour de plusieurs axes de Recherche :

- Dans le modèle des Holons de type « pronostic global », l'algorithme de fusion, d'unification, et de validation au niveau flotte des résultats des études de classification, d'influence, de prédiction ou d'explication des différents Holons de type « pronostic local » et de REX, ne dispose ni de capacité d'apprentissage intelligent ni d'outils de comparaison détaillée des situations observées au niveau d'un type d'équipement niveau flotte. Une perspective consiste dès lors à envisager l'intégration de réseaux de neurones artificiels afin d'enrichir l'architecture de surveillance EMH² proposée au travers d'un nouveau modèle disposant de capacité de classification et de généralisation de règles de surveillance d'une base de connaissance [67].
- Comme annoncé dans la section 2.5 du chapitre I, nos travaux ne couvrent que la dimension technologique de la surveillance. D'autres dimensions, succinctement abordées, notamment psychologique, ergonomique (coopération homme-machine), organisationnelle (formation des ingénieurs, nouveaux métiers) et financière (analyse des coûts), pourraient enrichir avantagement l'architecture EMH² proposée par de nouvelles approches et des outils centrée sur l'humain.
- Nous avons également précisé plusieurs hypothèses de travail dans le chapitre I (absence d'événements rares) et III (observabilité totale des événements et granularité jusqu'aux émetteurs de ces événements) qu'il serait intéressant d'étudier plus en profondeur afin de cerner de manière plus précise les limites de nos travaux et leur validité dans le monde réel, ce qui pourrait nous pousser à rendre notre contribution encore plus générique en relâchant l'une ou l'autre de ces hypothèses.
- L'émergence de la vision « service » dans le secteur du transport ou les systèmes de transport autonomes [144], [145] vont devenir des facteurs important pour la compétitivité des industriels. Ceci les incitera sous peu à développer de nouveaux services dans le domaine de la maintenance et de la surveillance de flotte. De notre point de vue, nos travaux pourraient aider les constructeurs et les opérateurs de flotte à développer ces nouveaux services. Dans ce contexte, une perspective concerne l'étude et l'adaptation de nos propositions selon cette vision « service » ou « véhicule autonome ». Par exemple, un train autonome devrait logiquement intégrer des mécanismes d'auto-surveillance, ce que permet l'architecture EMH².
- Avec les cycles de vie relativement longs des systèmes mobiles considérés, les connaissances générées par EMH² vieillissent logiquement. Par conséquent, la notion d'oubli, et de facteur d'oubli, mériterait d'être intégré dans EMH² afin de favoriser les connaissances les plus récentes.
- Finalement, la dimension « développement durable » n'est pas considérée explicitement dans nos propos. Placer EMH² selon une perspective en lien notamment avec la maintenance durable constitue ainsi de facto une perspective très intéressante : elle permettrait de considérer par exemple la consommation énergétique de la flotte des

17 <https://norminfo.afnor.org/norme/X60-500/terminologie-relative-a-la-fiabilite-maintenabilite-disponibilite/92160>

systèmes complexes surveillés, et plus généralement, leur empreinte environnementale (risque de pollution, dégagement de gaz à effet de serre, etc.).

Références

- [1] A. Fadil, J. Clarhaut, G. Branger, and D. Trentesaux, "Smart Condition Based Maintenance (S-CBM) for a Fleet of Mobile Entities," in *Service Orientation in Holonic and Multi-Agent Manufacturing*, vol. Studies in Computational Intelligence book series (SCI, volume 694), Springer, Cham, 2017, pp. 115–123.
- [2] J. Alanen, K. Haataja, O. Laurila, J. Peltola, and I. Aho, "Diagnostics of mobile works machines," <http://www.vtt.fi/inf/pdf/tiedotteet/2006/T2343.pdf>, VTT Technical Research Centre of Finland-2006. .
- [3] A. Muller, A. Crespo Marquez, and B. lung, "On the concept of e-maintenance: Review and current research," *Reliab. Eng. Syst. Saf.*, vol. 93, no. 8, pp. 1165–1187, Aug. 2008.
- [4] A. Le Mortellec, J. Clarhaut, Y. Sallez, T. Berger, and D. Trentesaux, "Embedded holonic fault diagnosis of complex transportation systems," *Eng. Appl. Artif. Intell.*, vol. 26, no. 1, pp. 227–240, Jan. 2013.
- [5] "ISO/IEC/IEEE 15288:2015 Systems and software engineering -- System life cycle processes." [Online]. Available: <https://www.iso.org/standard/63711.html>.
- [6] W. E. Deming, *The new economics for industry, government, education*. Cambridge, Mass.: Massachusetts Institute of Technology. Center for Advanced Engineering Study, 1993.
- [7] A. 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," Université de Valenciennes et du Hainaut Cambrésis, 2014.
- [8] D. Trentesaux, T. Knothe, G. Branger, and K. Fischer, "Planning and Control of Maintenance, Repair and Overhaul Operations of a Fleet of Complex Transportation Systems: A Cyber-Physical System Approach," in *Service Orientation in Holonic and Multi-agent Manufacturing*, T. Borangiu, A. Thomas, and D. Trentesaux, Eds. Cham: Springer International Publishing, 2015, pp. 175–186.
- [9] D. F. Haasl, N. H. Roberts, W. E. Vesely, and F. F. Goldberg, "Fault tree handbook," Nuclear Regulatory Commission, Washington, DC (USA). Office of Nuclear Regulatory Research, NUREG-0492, Jan. 1981.
- [10] M. Andreacchio, A. Bekrar, R. Benmansour, and D. Trentesaux, "Balancing preventive and corrective maintenance of aircraft assets: A cyber-physical systems approach," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 500–503.
- [11] R. L. Flood and E. Carson, *Dealing with Complexity: An Introduction to the Theory and Application of Systems Science*, 2nd ed. Springer US, 1993.
- [12] R. Shabali, "Definition: What is a complex system?," *webmindset*, 16-Aug-2015. .
- [13] L. Escobar *et al.*, "Design and implementation of complex systems using Mechatronics and Cyber-Physical Systems approaches," in *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2017, pp. 147–154.
- [14] J. Mbuli, D. Trentesaux, J. Clarhaut, and G. Branger, "Decision support in condition-based maintenance of a fleet of cyber-physical systems: A fuzzy logic approach," in *2017 Intelligent Systems Conference (IntelliSys)*, 2017, pp. 82–89.
- [15] E. Thomas, "Contribution à la prise de décision dynamique en maintenance prévisionnelle par formalisation d'un principe d'opportunité," thesis, Nancy 1, 2009.
- [16] "AFNOR: Terminologie de la maintenance. NF EN 13306 - Janvier 2018," 2018. [Online]. Available: <https://www.boutique.afnor.org/norme/nf-en-13306/maintenance-terminologie-de-la-maintenance/article/828621/fa185755>.
- [17] "AFNOR: Norme X60-500 | Terminologie relative à la fiabilité - Maintenabilité - Disponibilité," 1988. [Online]. Available: <https://norminfo.afnor.org/norme/X60-500/terminologie-relative-a-la-fiabilite-maintenabilite-disponibilite/92160>.

- [18] D. Trentesaux and G. Branger, "Data management architectures for the improvement of the availability and maintainability of a fleet of complex transportation systems: a state-of-the-art review," *Sohoma17*, Studies in Computational Intelligence book series-2018.
- [19] L. Belkacem, Z. Simeu-Abazi, H. Dhouibi, E. Gascard, and H. Messaoud, "Diagnostic and prognostic of hybrid dynamic systems: Modeling and RUL evaluation for two maintenance policies," *Reliab. Eng. Syst. Saf.*, vol. 164, pp. 98–109, Aug. 2017.
- [20] M. Monnin, A. Voisin, J.-B. Léger, and B. lung, "Fleet-wide health management architecture," presented at the Annual Conference of the Prognostics and Health Management Society, PHM 2011, 2011, pp. 547–554.
- [21] R. Lesobre, *Modélisation et optimisation de la maintenance et de la surveillance des systèmes multi-composants - Applications à la maintenance et à la conception de véhicules industriels*. Grenoble Alpes, 2015.
- [22] A. Yokoyama, "Innovative Changes for Maintenance of Railway by Using ICT—To Achieve 'Smart Maintenance,'" *Procedia CIRP*, vol. 38, pp. 24–29, 2015.
- [23] Roman Muzalevsky, "The Implications of China's High-Speed Eurasian Railway Strategy for Central Asia," *The Jamestown Foundation*, 2010. [Online]. Available: http://www.jamestown.org/single/?no_cache=1&tx_ttnews%5Btt_news%5D=36234. [Accessed: 12-Jun-2016].
- [24] "CEN - EN 13306 - Maintenance - Maintenance terminology | Engineering360." [Online]. Available: <https://standards.globalspec.com/std/1297524/cen-en-13306>.
- [25] Mobley, R.K., Higgins, L.R, and Wikoff, D.J., *Maintenance Engineering Handbook.*, 7th Edition. McGraw-Hill Professional Publishing. 2008.
- [26] H. ARNOUX, "Environnement, sécurité et maintenance," *Ref : TIP095WEB - "Maintenance,"* 10-Oct-2005. [Online]. Available: <https://www.techniques-ingenieur.fr/base-documentaire/genie-industriel-th6/fonction-strategique-de-la-maintenance-42137210/environnement-securite-et-maintenance-mt9555/>. [Accessed: 09-Apr-2018].
- [27] O. Boiral, "Environnement et économie : une relation équivoque," *Vertigo - Rev. Électronique En Sci. Environ.*, no. Volume 5 Numéro 2, Nov. 2004.
- [28] E. Levrat, B. lung, and A. C. Marquez, "E-maintenance: review and conceptual framework," *Prod. Plan. Control*, vol. 19, no. 4, pp. 408–429, Jun. 2008.
- [29] Zaim, S., Turkyilmaz, A., Mehmet, F. A., Umar, A. and Omer, F. D., *Maintenance Strategy Selection Using AHP and ANP Algorithms: a Case Study*. Journal of Quality in Maintenance Engineering, 2012.
- [30] Ribot, P, "Vers l'intégration diagnostic / pronostic pour la maintenance des systèmes complexes.," Université de Toulouse 3 Paul Sabatier, 2009.
- [31] R. K. Mobley, *An introduction to predictive maintenance* /. New York, NY : Van Nostrand Reinhold, 1990.
- [32] 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., 2009.
- [33] M. Bengtsson, *Condition Based Maintenance Systems—An investigation of technical constituents and organizational aspects*. 2004.
- [34] Deloux, E., "Politiques de maintenance conditionnelle pour un système à dégradation continue soumis à un environnement stressant.," Thèse de doctorat. Université de Nantes., 2008.
- [35] M. Kajko-Mattsson, "Corrective Maintenance Maturity Model: Problem Management," in *International Conference on Software Maintenance, 2002. Proceedings.(ICSM), 2002*, p. 0486.
- [36] Alali Alhouaij, A., "Contribution à l'optimisation de la maintenance dans un contexte distribué.," Université de Grenoble, 2010.
- [37] P. Cochetoux, "Contribution à la maintenance proactive par la formalisation du processus de pronostic des performances de systèmes industriels," thesis, Nancy 1, 2010.

- [38] R. Dekker, R. E. Wildeman, and F. A. van der Duyn Schouten, "A review of multi-component maintenance models with economic dependence," *Math. Methods Oper. Res.*, vol. 45, no. 3, pp. 411–435, Oct. 1997.
- [39] E. BERK and K. MOINZADEH, "Analysis of maintenance policies for M machines with deteriorating performance," *IIE Trans.*, vol. 32, no. 5, pp. 433–444, May 2000.
- [40] L. Papić, B. Vasić, and J. Knežević, "Sources of fuzziness in vehicle maintenance management," *J. Qual. Maint. Eng.*, vol. 3, no. 4, pp. 281–288, Dec. 1997.
- [41] M. Lebold, K. Reichard, and D. Boylan, "Utilizing dcom in an open system architecture framework for machinery monitoring and diagnostics," in *2003 IEEE Aerospace Conference, 2003. Proceedings*, 2003, vol. 3, pp. 3_1227-3_1236.
- [42] M. Basseville and M.-O. Cordier, "Surveillance et diagnostic de systèmes dynamiques: approches complémentaires du traitement de signal et de l'intelligence artificielle," INRIA, report, 1996.
- [43] A. Azarian, A. Siadat, and P. Martin, "A new strategy for automotive off-board diagnosis based on a meta-heuristic engine," *Eng. Appl. Artif. Intell.*, vol. 24, no. 5, pp. 733–747, Aug. 2011.
- [44] 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." .
- [45] E. Reed, J. Schumann, and O. J. Mengshoel, "Verification and Validation of System Health Management Models using Parametric Testing," *Proc InfotechAerospace 2011*, 2011.
- [46] Guillén, A. J., Gómez, J. F., Crespo, A., Guerrero, A., Sola, A., & Barbera, L., "Advances in PHM Application Frameworks: Processing Methods, Prognosis Models, Decision Making. Chemical Engineering Transaction, Vol.33, 391-396."
- [47] Vachtsevanos, G. J., Lewis, F.L., "Intelligent fault diagnosis and prognosis for engineering systems. 1st ed., John Wiley & Sons, Hoboken, New Jersey."
- [48] D. A. Tobon-Mejia, K. Medjaher, and N. Zerhouni, "CNC machine tool's wear diagnostic and prognostic by using dynamic Bayesian networks," *Mech. Syst. Signal Process.*, vol. 28, pp. 167–182, Apr. 2012.
- [49] Khol, J., Bauer, "Role-Based Diagnosis for Distributed Vehicle Functions. In 21st International Workshop on the Principles of Diagnosis (DX), Portland USA.," 2010. [Online]. Available: <https://www.phmsociety.org/node/494>.
- [50] 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.
- [51] International Atomic Energy Agency, *Implementation strategies and tools for condition based maintenance at nuclear power plants*. Vienna: International Atomic Energy Agency, 2007.
- [52] N. Stanfelj, T. E. Marlin, and J. F. MacGregor, "Monitoring and Diagnosing Process Control Performance: The Single-Loop Case," in *IEEE Xplore, 1991 American Control Conference*, 1991, pp. 2886–2892.
- [53] J. Arlat, K. Kanoun, *Guide de la sûreté de fonctionnement*, Edition Cépaduès. 1999.
- [54] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part I: Quantitative model-based methods," *Comput. Chem. Eng.*, vol. 27, no. 3, pp. 293–311, Mar. 2003.
- [55] W. Hamscher, L. Console, and J. de Kleer, Eds., *Readings in Model-based Diagnosis*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1992.
- [56] Q. Gaudel, "Approche intégrée de diagnostic et de pronostic pour la gestion de santé des systèmes hybrides sous incertitude," phdthesis, INSA de Toulouse, 2016.
- [57] Q. Zhang and X. Zhang, "Distributed sensor fault diagnosis in a class of interconnected nonlinear uncertain systems," *Annu. Rev. Control*, vol. 37, no. 1, pp. 170–179, Apr. 2013.
- [58] J. Zhang, M. Qi, and B. Li, "Hybrid diagnosis of fault tree and Bayesian network in BIW automatic welding production line," in *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 2016, pp. 297–300.

- [59] J. A. Martin, "Knowledge-based supervision and diagnosis of complex process," in *Proceedings of the 1999 IEEE International Symposium on Intelligent Control Intelligent Systems and Semiotics (Cat. No.99CH37014)*, 1999, pp. 225–230.
- [60] Dubuisson B., *Diagnostic et reconnaissance des formes*, Hermès. 1990.
- [61] M. Partridge and M. Jabri, "Robust principal component analysis," in *Neural Networks for Signal Processing X. Proceedings of the 2000 IEEE Signal Processing Society Workshop (Cat. No.00TH8501)*, 2000, vol. 1, pp. 289–298 vol.1.
- [62] R. g Li, Y. q Zhu, H. f Xu, and D. x Liu, "Statistical multiple regression analysis' application in the cure on diabetes mellitus," in *The 2nd International Conference on Information Science and Engineering*, 2010, pp. 2383–2386.
- [63] M. M. Rahman, C. Charoenlarnnoppa, and P. Suksompong, "Classification and pattern recognition algorithms applied to E-Nose," in *2015 2nd International Conference on Electrical Information and Communication Technologies (EICT)*, 2015, pp. 44–48.
- [64] A. E. Bakri, Y. Berrada, and I. Boumhidi, "Bayesian regularized artificial neural network for fault detection and isolation in wind turbine," in *2017 Intelligent Systems and Computer Vision (ISCV)*, 2017, pp. 1–6.
- [65] F. Atalar and A. Kuntman, "Diagnosis of power transformers faults using fuzzy logic method," in *2016 National Conference on Electrical, Electronics and Biomedical Engineering (ELECO)*, 2016, pp. 28–32.
- [66] V. Atamuradov, K. Medjaher, P. Dersin, B. Lamoureux, and N. Zerhouni, "Prognostics and health management for maintenance practitioners - Review, implementation and tools evaluation," *Int. J. Progn. Health Manag.*, vol. 8, pp. 1–31, Dec. 2017.
- [67] El koujok, M., "Contribution au pronostic industriel : intégration de la confiance à un modèle prédictif neuro-flou.," Thèse de Doctorat. Université de Franche-Comté., 2010.
- [68] N. Herr, J. Nicod, C. Varnier, N. Zerhouni, and P. Dersin, "Predictive maintenance of moving systems," in *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, 2017, pp. 1–6.
- [69] T. Gullledge, S. Hiroshige, and R. Iyer, "Condition-based Maintenance and the product improvement process," *Comput. Ind.*, vol. 61, no. 9, pp. 813–832, Dec. 2010.
- [70] G. Rizzoni, S. Onori, and M. Rubagotti, "Diagnosis and Prognosis of Automotive Systems: motivations, history and some results," *IFAC Proc. Vol.*, vol. 42, no. 8, pp. 191–202, Jan. 2009.
- [71] J. Luo, K. R. Pattipati, L. Qiao, and S. Chigusa, "Model-Based Prognostic Techniques Applied to a Suspension System," *IEEE Trans. Syst. Man Cybern. - Part Syst. Hum.*, vol. 38, no. 5, pp. 1156–1168, Sep. 2008.
- [72] Bengtsson, M., "Condition Based Maintenance On Rail Vehicles, Possibilities for a more effective maintenance strategy.," *Tech. Rep. Eskilstuna Swed.*, 2003.
- [73] J. Mbuli, D. Trentesaux, and T. Dailly, "Specifying a Condition-Based Maintenance Decision Support System of a Fleet of Cyber-Physical Systems," in *Service Orientation in Holonic and Multi-Agent Manufacturing*, Springer, 2019, pp. 285–294.
- [74] J. Barbosa, P. Leitão, D. Trentesaux, A. W. Colombo, and S. Karnouskos, "Cross benefits from cyber-physical systems and intelligent products for future smart industries," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 504–509.
- [75] W. Mechri, C. Simon, B. lung, D. Theilliol, and F. Peysson, "Investigation for defining a first functional architecture of an intelligent embedded diagnosis system," in *14th International Workshop on Advanced Control and Diagnosis, ACD 2017*, Bucharest, Romania, 2017.
- [76] G. Wang, A. Gunasekaran, E. W. T. Ngai, and T. Papadopoulos, "Big data analytics in logistics and supply chain management: Certain investigations for research and applications," *Int. J. Prod. Econ.*, vol. 176, pp. 98–110, juin 2016.
- [77] H. Panetto, N. Boudjlida, and H El Haouzi, *Interoperability for Enterprise Software and Applications: Proceedings of the Workshops and the Doctoral Symposium of the Second IFAC/IFIP I-ESA International Conference: EI2N, WSI, IS-TSPQ 2006*. John Wiley & Sons, 2010.

- [78] A. M. Gergely and B. Crainicu, "The Concept of a Distributed Repository for Validating X.509 Attribute Certificates in a Privilege Management Infrastructure," *Procedia Technol.*, vol. 22, pp. 926–930, Jan. 2016.
- [79] J. Chen *et al.*, "A Big Data Analysis and Application Platform for Civil Aircraft Health Management," in *2016 IEEE Second International Conference on Multimedia Big Data (BigMM)*, 2016, pp. 404–409.
- [80] A. Núñez, J. Hendriks, Z. Li, B. D. Schutter, and R. Dollevoet, "Facilitating maintenance decisions on the Dutch railways using big data: The ABA case study," in *2014 IEEE International Conference on Big Data (Big Data)*, 2014, pp. 48–53.
- [81] Sammouri, W, "Data mining of temporal sequences for the prediction of infrequent failure events : application on floating train data for predictive maintenance.," Université Paris-Est, 2014.
- [82] Q. Shi and M. Abdel-Aty, "Big Data applications in real-time traffic operation and safety monitoring and improvement on urban expressways," <http://www.sciencedirect.com/science/article/pii/S0968090X15000777>, p. Transportation Research Part C: Emerging Technologies Volume 58, Part B, Pages 161-428, 2015.
- [83] M. V. Corazza, S. Magnalardo, D. Vasari, E. Petracchi, and M. Tozzi, "The EBSF_2 innovative predictive maintenance system for buses: A case study to synergetically improve operational and environmental bus performance," in *2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*, 2017, pp. 1–6.
- [84] Hao Wang, O. L. Osen, Guoyuan Li, Wei Li, Hong-Ning Dai, and Wei Zeng, "Big data and industrial Internet of Things for the maritime industry in Northwestern Norway," in *TENCON 2015 - 2015 IEEE Region 10 Conference*, <http://ieeexplore.ieee.org/document/7372918/>, 2015, pp. 1–5.
- [85] E. Jantunen, J. Junnola, and U. Gorostegui, "Maintenance supported by cyber-physical systems and cloud technology," in *IEEE Xplore, 2017 4th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2017, pp. 0708–0713.
- [86] G. Klas, "Edge Computing and the Role of Cellular Networks," *Computer*, vol. 50, no. 10, pp. 40–49, 2017.
- [87] W. Qiu and R. Kumar, "Decentralized failure diagnosis of discrete event systems," *IEEE Trans. Syst. Man Cybern. - Part Syst. Hum.*, vol. 36, no. 2, pp. 384–395, Mar. 2006.
- [88] C. E. Balin and J. Stankūnas, "Investigation of Fault Detection and Analysis Methods for Central Maintenance Systems," *Aviat. Technol.*, vol. 1, no. 1, Jan. 2013.
- [89] "624-1 Design Guidance for Onboard Maintenance System," *SAE ITC*, 16-Oct-2017. [Online]. Available: <https://www.aviation-ia.com/products/624-1-design-guidance-onboard-maintenance-system-2>.
- [90] O. Nasri, H. Shraim, P. Dague, O. Heron, and M. Cartron, "Modeling and Deployment of Model-Based Decentralized Embedded Diagnosis inside Vehicles: Application to Smart Distance Keeping Function," *International Journal of Vehicular Technology*, 2012. [Online]. Available: <https://www.hindawi.com/journals/ijvt/2012/103530/>.
- [91] S. Tambe, A.-M. E. Bayoumi, A. Cao, R. McCaslin, and T. Edwards, "An Extensible CBM Architecture for Naval Fleet Maintenance Using Open Standards | Data Distribution Service (DDS) Community RTI Connex Users," 2016. [Online]. Available: <https://community.rti.com/paper/extensible-cbm-architecture-naval-fleet-maintenance-using-open-standards>.
- [92] I. Roychoudhury, G. Biswas, and X. Koutsoukos, "Designing Distributed Diagnosers for Complex Continuous Systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 2, pp. 277–290, Apr. 2009.
- [93] J. Barbosa, P. Leitão, E. Adam, and D. Trentesaux, "Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution," *Comput. Ind.*, vol. 66, pp. 99–111, Jan. 2015.
- [94] Koestler, A, *The ghost in the machine. 1967. London: Hutchinson. 1967.*

- [95] G. Zambrano Rey, C. Pach, N. Aissani, A. Bekrar, T. Berger, and D. Trentesaux, "The control of myopic behavior in semi-heterarchical production systems: A holonic framework," *Eng. Appl. Artif. Intell.*, vol. 26, no. 2, pp. 800–817, Feb. 2013.
- [96] F. Versteegh, M. A. Salido, and A. Giret, "A holonic architecture for the global road transportation system," *J. Intell. Manuf.*, vol. 21, no. 1, pp. 133–144, Feb. 2010.
- [97] C. H. J. Beckers, "A holonic architecture for the global automated transportation system (GATS)," 18-Sep-2008. [Online]. Available: <http://essay.utwente.nl/59154/>.
- [98] M. Diévert, "PhD Thesis : Architectures de diagnostic et de pronostic distribuées de systèmes techniques complexes de grande dimension (Distributed architectures for diagnosis and prognosis of large scale complex technical systems)," 03-Dec-2010. [Online]. Available: <http://ethesis.inp-toulouse.fr/archive/00001394/>.
- [99] D. Trentesaux and G. Branger, "Foundation of the Surfer Data Management Architecture and Its Application to Train Transportation," in *Service Orientation in Holonic and Multi-Agent Manufacturing: Proceedings of SOHOMA 2017*, T. Borangiu, D. Trentesaux, A. Thomas, and O. Cardin, Eds. Cham: Springer International Publishing, 2018, pp. 111–125.
- [100] G. Branger *et al.*, "Système Et Procédé Pour La Gestion D'actifs De Trains Ferroviaires," patent n° WO/2017/211593, 15-Dec-2017.
- [101] R. Bharadwaj, "Vehicle Integrated Prognostic Reasoner (VIPR) Final Report," NASA NTRS, <https://ntrs.nasa.gov/search.jsp?R=20130011181>, Mar. 2013.
- [102] D. Cornhill, "Vehicle Integrated Prognostic Reasoner (VIPR) Metric Report," Apr. 2013.
- [103] M. Thurston and M. Lebold, "Open Standards for Condition-Based Maintenance and Prognostic Systems," *Proc 5th Annu. Maint. Reliab. Conf. Gatlinburg USA 2001*.
- [104] "ISO 13374-1:2003 - Condition monitoring and diagnostics of machines -- Data processing, communication and presentation -- Part 1: General guidelines." [Online]. Available: <https://www.iso.org/standard/21832.html>.
- [105] O. Cardin and P. Castagna, "Using online simulation in Holonic manufacturing systems," *Eng. Appl. Artif. Intell.*, vol. 22, no. 7, pp. 1025–1033, Oct. 2009.
- [106] G. Hunter, *Metalogic: An Introduction to the Metatheory of Standard First Order Logic*. University of California Press, 1973.
- [107] B. Jukić, N. Jukić, and S. Nestorov, "Framework for integrating process and data logic: Connecting UML use cases and ER diagrams," in *Proceedings of the ITI 2013 35th International Conference on Information Technology Interfaces*, 2013, pp. 109–115.
- [108] P. Wegner and M. Israel, Eds., *ACM Comput. Surv., March 1996*, vol. 28. New York, NY, USA: ACM, 1996.
- [109] "EIF, 'European Interoperability Framework,' White Pages. pp. 1–40, 2004.,".
- [110] Z. Haibei and Y. Xu, "The Architecture Design of a Distributed Workflow System," in *2012 11th International Symposium on Distributed Computing and Applications to Business, Engineering Science*, 2012, pp. 9–12.
- [111] D. Aït-Kadi, Z. Simeu-Abazi, and A. Arous, "Fault isolation by test scheduling for embedded systems using a probabilistic approach," *J. Intell. Manuf.*, vol. 29, no. 3, pp. 641–649, Mar. 2018.
- [112] T. Kempowsky, "Surveillance de procédés à base de méthodes de classification : conception d'un outil d'aide pour la détection et le diagnostic des défaillances," phdthesis, INSA de Toulouse, 2004.
- [113] P. Salvaneschi, M. Cedei, and M. Lazzari, "Applying AI to structural safety monitoring and evaluation," *IEEE Expert*, vol. 11, no. 4, pp. 24–34, Aug. 1996.
- [114] M. Lazzari and P. Salvaneschi, "Embedding a Geographic Information System in a Decision Support System for Landslide Hazard Monitoring," *Nat. Hazards*, vol. 20, no. 2–3, pp. 185–195, Nov. 1999.
- [115] S. Lancini, M. Lazzari, A. Masera, and P. Salvaneschi, "Diagnosing Ancient Monuments with Expert Software," *Struct. Eng. Int.*, vol. 7, no. 4, pp. 288–291, Nov. 1997.

- [116] N. Chomsky, "Three models for the description of language," *IRE Trans. Inf. Theory*, vol. 2, no. 3, pp. 113–124, Sep. 1956.
- [117] L. Li and X. Zhang, "Study of data mining algorithm based on decision tree," in *2010 International Conference On Computer Design and Applications*, 2010, vol. 1, pp. V1-155-V1-158.
- [118] A. F. A. Pinem and E. B. Setiawan, "Implementation of classification and regression Tree (CART) and fuzzy logic algorithm for intrusion detection system," in *2015 3rd International Conference on Information and Communication Technology (ICoICT)*, 2015, pp. 266–271.
- [119] S. L. Salzberg, "C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993," *Mach. Learn.*, vol. 16, no. 3, pp. 235–240, Sep. 1994.
- [120] R. Pandya and J. Pandya, "C5. 0 Algorithm to Improved Decision Tree with Feature Selection and Reduced Error Pruning," *Int. J. Comput. Appl.*, vol. 117, no. 16, pp. 18–21, May 2015.
- [121] A. H. C. Tsang, "Condition-based maintenance: tools and decision making," *J. Qual. Maint. Eng.*, vol. 1, no. 3, pp. 3–17, Sep. 1995.
- [122] A. C. Márquez, *The Maintenance Management Framework: Models and Methods for Complex Systems Maintenance*. London: Springer-Verlag, 2007.
- [123] Salonen, A, "Formulation of Maintenance Strategies – A Simplified Process," presented at the International Journal of COMADEM, 15(3), 9-18., 2012.
- [124] Andréasson, N., "Optimization of opportunistic replacement activities in deterministic and stochastic multi-component systems," Chalmers University of Technology, 2004.
- [125] H. Wang, "A survey of maintenance policies of deteriorating systems," *Eur. J. Oper. Res.*, vol. 139, no. 3, pp. 469–489, 2002.
- [126] R. P. Nicolai and R. Dekker, "Optimal Maintenance of Multi-component Systems: A Review," in *Complex System Maintenance Handbook*, Springer, London, 2008, pp. 263–286.
- [127] Bouvard, K, "Planification de la maintenance base sur la dégradation de composants. Application à un véhicule industriel," Université de Lille 1, France, 2010.
- [128] C. J. Hockley, "Design for success:," *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.*, Nov. 2005.
- [129] U. Dinesh Kumar, J. Knezevic, and J. Crocker, "Maintenance free operating period – an alternative measure to MTBF and failure rate for specifying reliability?," *Reliab. Eng. Syst. Saf.*, vol. 64, no. 1, pp. 127–131, Apr. 1999.
- [130] M. A. Brown and C. J. Hockly, "Cost of specifying maintenance/failure free operating periods for Royal Air Force aircraft," in *Annual Reliability and Maintainability Symposium. 2001 Proceedings. International Symposium on Product Quality and Integrity (Cat. No.01CH37179)*, 2001, pp. 425–432.
- [131] T. D. Quynh, L. T. H. An, and K. H. Adjallah, "DCA for Minimizing the Cost and Tardiness of Preventive Maintenance Tasks under Real-Time Allocation Constraint," in *Intelligent Information and Database Systems*, 2010, pp. 410–419.
- [132] P. F. Wilson, L. D. Dell, and G. F. Anderson, *Root cause analysis : a tool for total quality management*. Milwaukee, Wis. : ASQC Quality Press, 1993.
- [133] A. Fadil, D. Trentesaux, A. Bekrar, and G. Branger, "Architecture de surveillance à base d'agents intelligents pour l'aide à la maintenance d'une flotte de systèmes mobiles," presented at the 25èmes journées STP du GdR MACS, Arts et Métiers campus de Lille, Lille-France, 2018.
- [134] A. Fadil, D. Trentesaux, A. Bekrar, and G. Branger, "Event management architecting for the monitoring and diagnosis of a fleet of trains," *J. Mod. Transp. JMTR*, 2018.
- [135] B. Markelj and I. Bernik, "Safe use of mobile devices arises from knowing the threats," *J. Inf. Secur. Appl.*, vol. 20, no. Supplement C, pp. 84–89, Feb. 2015.
- [136] S. Blanch-Torné, F. Cores, and R. M. Chiral, "Agent-based PKI for Distributed Control System," in *2015 World Congress on Industrial Control Systems Security (WCICSS)*, 2015, pp. 28–35.
- [137] S. Ahmad, K. M. R. Alam, H. Rahman, and S. Tamura, "A comparison between symmetric and asymmetric key encryption algorithm based decryption mixnets," in *2015 International Conference on Networking Systems and Security (NSysS)*, 2015, pp. 1–5.

- [138] N. Dragoni *et al.*, “Microservices: Yesterday, Today, and Tomorrow,” in *Present and Ulterior Software Engineering*, Springer, Cham, 2017, pp. 195–216.
- [139] M. Villamizar *et al.*, “Evaluating the monolithic and the microservice architecture pattern to deploy web applications in the cloud,” in *2015 10th Computing Colombian Conference (10CCC)*, 2015, pp. 583–590.
- [140] “An Introduction to Service-Oriented Architecture from a Java Developer Perspective - O’Reilly Media,” 01-Nov-2017. [Online]. Available: <http://www.onjava.com/pub/a/onjava/2005/01/26/soa-intro.html>.
- [141] X. Li, Z. Liu, and H. Jifeng, “A formal semantics of UML sequence diagram,” in *2004 Australian Software Engineering Conference. Proceedings.*, 2004, pp. 168–177.
- [142] “IEC 61375-1:2012 | IEC Webstore.” [Online]. Available: <https://webstore.iec.ch/publication/5397>.
- [143] “IEC,” *IEC 61375-2-5 IEC Webstore*, 2014. [Online]. Available: <https://webstore.iec.ch/publication/5400>.
- [144] D. Trentesaux *et al.*, “The Autonomous Train,” in *2018 13th Annual Conference on System of Systems Engineering (SoSE)*, IEEE, 2018, pp. 514–520.
- [145] A. Diaz and D. Trentesaux, “Servitization in Train Transportation,” in *Service Orientation in Holonic and Multi-Agent Manufacturing*, Springer, 2019, pp. 273–284.
- [146] B. Dubuisson, *Diagnostic, intelligence artificielle et reconnaissance des formes*. Paris: Hermes Science Publications, 2001.
- [147] S. Tufféry, *Data mining et statistique décisionnelle - 3ème Edition*, 3e édition revue et augmentée. Paris: Editions Technip, 2010.
- [148] N. Chomsky, “Three models for the description of language,” *IRE Trans. Inf. Theory*, vol. 2, no. 3, pp. 113–124, Sep. 1956.
- [149] “Chomsky hierarchy,” *Wikipedia*. 12-Aug-2017.
- [150] J. Ayling, R. Moore, and G. Tu, “A high-performance monolithic store,” in *1969 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, 1969, vol. XII, pp. 36–37.
- [151] D. Shadija, M. Rezai, and R. Hill, “Towards an understanding of microservices,” in *2017 23rd International Conference on Automation and Computing (ICAC)*, 2017, pp. 1–6.

Liste de figures

Figure 1. 1 : La maintenance dans le cycle de vie du système[25]	20
Figure 1. 2: Différents types de maintenance[7], [33].....	22
Figure 1. 3: Fonctions de surveillance dynamique et classique, adaptée de[7], [47].....	25
Figure 1. 4: Principe des méthodes de diagnostic à base de modèles[7]	27
Figure 1. 5: Classification des méthodes de diagnostic[53].....	28
Figure 1. 6: Une architecture hiérarchique générique à trois niveaux : fonctionnel, exécutif et décisionnel[53].....	31
Figure 1. 7: Une architecture orientée-santé générique[53].....	31
Figure 1. 8: Une flotte de systèmes complexes mobiles homogènes maintenus par un CM.....	32
Figure 2. 1: Architecture de surveillance centralisée.....	38
Figure 2. 2: Une illustration des services centralisés de surveillance dans le transport ferroviaire, l'Alstom Traintracer[76]	39
Figure 2. 3: Le processus de surveillance centralisé des systèmes mobile[78]	40
Figure 2. 4 : Architecture typique d'un système PHM à l'échelle de la flotte	40
Figure 2. 5: Architecture de surveillance Edge-centralisé	41
Figure 2. 6: Architecture de capteurs d'un train de la SNCF.....	42
Figure 2. 7: Architecture de surveillance décentralisée	43
Figure 2. 8: Vue d'ensemble du CMS[83].....	43
Figure 2. 9: Diagnostic de pannes décentralisé basé sur un modèle du système SDK[85].....	44
Figure 2. 10: Architecture de bus de données pour la maintenance conditionnelle utilisant DDS et OSA-CBM[86]	44
Figure 2. 11: Architecture de surveillance décentralisé et coopératif.....	45
Figure 2. 12: Le contrôle du comportement myope dans les systèmes de production: un cadre holonique[90].....	46
Figure 2. 13: Structure hiérarchique du système de gestion de la route	47
Figure 2. 14: Architecture VIPR, adaptée de[95].	48
Figure 2. 15: Architecture EMH (Embedded Monitoring Hierarchy) [4]	48
Figure 2. 16: Modèle d'architecture ISO 13374[97].....	49
Figure 2. 17: Architecture organisationnelle IEDS en lien avec le système surveillé[70]	50
Figure 2. 18: Un modèle holonique d'un train avec application à la surveillance de porte[4].....	52
Figure 3. 1: Conception de l'architecture de surveillance	54
Figure 3. 2: Surveillance coopératives des Holons de l'architecture	55
Figure 3. 3: Surveillance récursives des Holons de l'architecture	56
Figure 3. 4: Description des couches d'abstraction de SurfEvent (inspiré de la norme ISO 13374).....	58
Figure 3. 5: Modèle générique d'un SurfEvent	58
Figure 3. 6: Modèle d'un cycle de vie SurfEvent.....	59
Figure 3. 7: Convertisseur d'évènements "SurfConvert"	60
Figure 3. 8: Un niveau Holonique dans l'EMH ²	61
Figure 3. 9: Les niveaux Holoniques dans l'EMH ²	63
Figure 3. 10 : Architecture d'un system Expert[105].....	65
Figure 3. 11: Description algorithmique du chaînage arrière du système expert d'un holon.....	67
Figure 3. 12 : Echange et coopération entre Holons de « pronostic local » et « pronostic global »	71
Figure 3. 13: Motivations pour la MFOP[123]	73

Figure 3. 14: Niveau Holonique « Maintenance Dynamique »	75
Figure 3. 15: Holon « politique de maintenance dynamique » d'une flotte des systèmes mobiles	77
Figure 3. 16: Synthèse de la politique de maintenance proposée[18].	80
Figure 3. 17: Processus «pm » pour la gestion des opérations de maintenance et REX	83
Figure 4. 1: modèle de la simulation d'un ensemble de SurfEvents.....	87
Figure 4. 2: modèle de la situation actuelle de référence	88
Figure 4. 3: Mesure de de la dispersion de x et y	88
Figure 4. 4 : Application « fonction générique ».....	89
Figure 4. 5 Résultat de l'expérimentation pour la situation de référence	89
Figure 4. 6 : modèle obtenu avec l'outil IBM® SPSS®	91
Figure 4. 7: Script de calcul de la durée d'expérimentation	91
Figure 4. 8 : Arbre de décision généré à partir du modèle d'expérimentation	92
Figure 4. 9 : comparaison entre la durée moyenne de calcul pour la situation de référence et celle de notre expérimentation.....	93
Figure 4. 10: Déploiement progressif de l'architecture EMH ²	95
Figure 4. 11: Une Implémentation possible de l'architecture EMH ² proposée en utilisant des microservices.	99
Figure 4. 12: Articulation des étapes composant le guide de déploiement proposé.....	101
Figure 5. 1: Instanciation d'un SurfEvent pour une flotte de trains	106
Figure 5. 2: Stratégie de raisonnement du moteur d'inférence mise en œuvre.	107
Figure 5. 3: protocole TCN d'un Train	108
Figure 5. 4: protocole ETB d'un Train.....	109
Figure 5. 5: microservices d'implémentation pour les 10 trains	109
Figure 5. 6: Editeur de connaissances de l'architecture EMH ²	111
Figure 5. 7: Outil d'aide au dépannage: interface utilisateur	112
Figure 5. 8: Evolution du KPI#1 sur plusieurs semaines.....	113

Annexes

Figure 1: Événement Entité-Association (EA)	131
Figure 2: vue schématique du processus de la formulation de la stratégie de maintenance[116]....	134
Figure 3: Classification of Noam CHOMSKY[140].....	136
Figure 4: Architecture monolithique.....	139
Figure 5: Architecture orientée services.....	139
Figure 6: Architecture microservice.....	140

Liste de Tableaux

Tableau 1. 1: Prise de décisions pour les nouvelles stratégies de maintenance[27]	23
Tableau 2. 1 : Évaluation des types d'architecture de surveillance par rapport aux spécifications demandées.....	51
Tableau 5. 1 : Situation actuelle pour chaque KPI	104
Tableau 5. 2: Objectifs visés pour chaque KPI.	105
Tableau 5. 3: Un échantillon de la base de connaissances des holons.....	107
Tableau 5. 4: Résultats atteints.....	112

Annexes

Annexe A : Modèle Entité-Association(EA) d'un SurfEvent

Le modèle EA est une représentation graphique d'un système d'information qui décrit les relations entre des objets, des concepts ou des événements. Un EA est une technique de modélisation de données pouvant aider à définir des processus métier et être utilisée comme base d'une base de données relationnelle[107]. Il vise à éliminer les redondances et les anomalies mises à jour. La modélisation d'un EA repose sur deux concepts:

- **Entités**, une entité est un objet dans le monde réel avec une existence indépendante qui peut être différenciée des autres objets. Elle dispose d'un identifiant, des attributs et des opérations.
- **Associations**, définies comme les relations ou les interactions entre les entités, elles peuvent disposer d'attributs et d'opérations. On distingue trois types d'association : association un à un (1: 1); association un à plusieurs (1: M); et association plusieurs à plusieurs (M: N).

La représentation graphique du modèle fournit une description détaillée et une vue globale et des classifications des entités utilisées et de leurs relations pouvant être utilisées dans les phases de conception de systèmes informatiques. En appliquant le modèle EA sur les paramètres d'un SurfEvent, on obtient les entités suivant :

- Types : cette entité dispose d'un identifiant et d'un attribut renseignant sur le terme « quantitatif » ou « qualitatif » ;
- Evènement « SurfEvent »: cette entité dispose d'un identifiant « Nom » et d'un attribut « calculé » ;
- Niveau : cette entité dispose d'un identifiant ;
- Statut : cette entité dispose d'un identifiant ;
- Emetteur : cette entité dispose d'un identifiant ;
- Contexte : cette entité dispose d'un identifiant et d'une valeur;

En ce qui concerne les associations entre entités:

- Un évènement « SurfEvent » dispose d'un type et un type est associé à un SurfEvent(1 :1) ;
- Un évènement « SurfEvent » dispose d'un statut et un statut est associé à un SurfEvent(1 :1) ;
- Un évènement « SurfEvent » appartient à un couches d'abstraction(Niveau), et dans une couche d'abstraction on peut avoir plusieurs SurfEvent (1 : N) ;
- Un évènement « SurfEvent » peut être émis par un émetteur, et un émetteur peut générer plusieurs SurfEvent (1 : N) ;
- Un émetteur peut avoir un parent et pour un parent on peut avoir plusieurs émetteurs, disposant chacun d'une liste de contextes (1 : N);

La Figure 1 suivant illustre la représentation graphique du modèle EA pour un SurfEvent. Après la normalisation du EA (en appliquant les règles d'association), nous obtenons ainsi un format générique d'évènement d'un SurfEvent qui tient compte des différentes couches d'abstraction et de la source émettrice (système cible), cela garantit à la fois la transition du monde réel à la représentation virtuelle du système, le respect des spécificités, ses relations, ses contextes et ses règles de fonctionnement, ainsi que la génération de métadonnées des systèmes cibles.

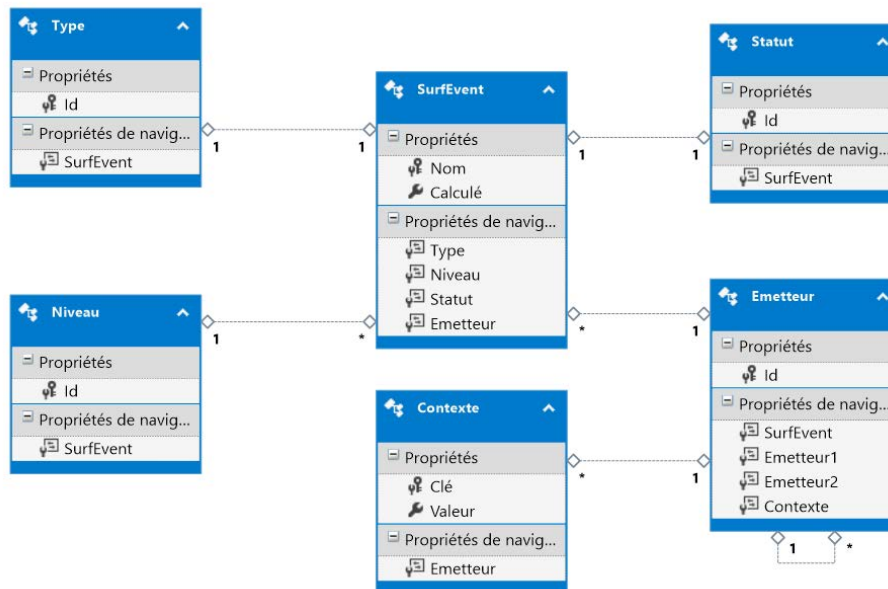


Figure 1: Événement Entité-Association (EA)

Annexe B : Approches de modélisation des fonctions de surveillance

Le choix des approches de modélisation des fonctions de diagnostic, pronostic et d'élaborations de conseils qui ont été implémentés dans les holons de surveillance de l'architecture EMH² est justifié dans cette annexe

1. Fonctions de diagnostic et de pronostic

Actuellement, plusieurs communautés de recherche scientifique (automatique, productique, informatique et IA...), s'intéressent au domaine de la supervision, et plus particulièrement au diagnostic et au pronostic, dans le but d'assister les opérateurs dans la prise de décision. Comme évoqué dans la partie (2.2) du chapitre I, du point de vue de la modélisation, les méthodes de diagnostic et de pronostic sont classifiées en 3 catégories : méthodes basées sur des modèles, basées sur des règles (base de connaissances) et basée sur les historiques des évènements. Nous précisons que l'état de l'art n'est pas exhaustif, car notre objectif est seulement de situer ces méthodes par rapport à leur application potentielle, et par rapport à nos objectifs en termes de réactivité, généricité et adaptabilité des mécanismes de surveillance sur des évènements de différente nature (quantitatif, qualitatif), dans un environnement distribué et multi-acteurs. Ces méthodes sont présentées ci-dessous :

- Dans le cadre des **méthodes basées sur des modèles**, la surveillance consiste à comparer la grandeur déduite d'un modèle représentatif du fonctionnement prédit des systèmes avec des mesures directement observées sur le comportement réel d'un système sous surveillance. La présence d'une déviation fournit l'indication d'une faute pouvant aboutir à une défaillance[55]. Ces comportements anormaux sont par la suite analysés dans le but de localiser les fautes. Les principaux inconvénients de ces méthodes sont la nécessité d'avoir des modèles mathématiques assez précis et complets, ce qui n'est pas toujours facile, voire impossible, pour des processus complexes et hiérarchiquement dépendant d'une flotte des

systèmes mobiles; une mauvaise modélisation des perturbations peut également engendrer des erreurs dans le modèle. Leurs capacités explicatives sont presque inexistantes et sont difficiles à adapter aux changements intervenant durant tout le cycle de vie des systèmes. Ces méthodes sont difficilement adaptables en cas de grand nombre de source de données (Big Data).

- Dans le cadre des **méthodes basées sur des connaissances**, la surveillance consiste à utiliser les connaissances et l'expérience humaines pour définir les connaissances explicites des relations causales entre les symptômes, les défaillances et les fautes, dans le but de garantir le bon fonctionnement des systèmes sous surveillance[58]. Ces relations sont souvent déterminées durant la phase de conception et proviennent d'une analyse fonctionnelle ou structurelle du système[56]. Les principaux inconvénients de ces méthodes sont qu'ils dépendent fortement de l'expertise humaine et restent étroitement liées à leurs domaines d'étude (l'adaptabilité de ces méthodes est limitée car les règles doivent être réécrites lorsque des modifications sont apportées au système), il est impossible de surveiller les fautes non enregistrées et il est difficile pour ces méthodes d'être utilisées dans un environnement distribué.
- Dans le cadre des **méthodes basées sur les historiques** des événements, la surveillance consiste à exploiter la disponibilité de données historiques du système sous surveillance afin de déterminer des changements de caractéristiques par rapport à des valeurs de référence ou à des progressions de défaillances[62]. Ces méthodes ont l'avantage de ne pas exiger de connaissance au préalable des systèmes et la disponibilité de grandes quantités de données historiques du processus est seulement nécessaire. Néanmoins, Le principal inconvénient de ces méthodes est que les représentations obtenues sont invariantes dans le temps et doivent être mises à jour périodiquement. Ces méthodes doivent en outre être accompagnées de mécanismes d'apprentissage pour construire la connaissance, ce qui n'est pas immédiat.

Il est évident que toutes ces méthodes de surveillance ont leurs forces et leurs faiblesses. Ainsi, après les avoir étudiés par rapport à notre besoin, nous avons opté plutôt pour l'intégration d'un ensemble de méthodes de surveillance a dont le choix dépend du type de fonction concernée :

- **Pour une fonction de diagnostic**, les méthodes basées sur la connaissance (plus particulièrement, les systèmes experts) sont retenues car elles offrent des capacités de raisonnement dans l'incertitude ainsi que des capacités d'explication et de localisation des solutions fournies [59].
- **Pour une fonction de pronostic**, Les méthodes basées sur des données historiques sont retenues car elles ont l'avantage de ne pas nécessiter de connaissance préalable approfondie pour caractériser l'accumulation et la progression des défaillances de systèmes sous surveillance.

L'Analyse en Composantes Principales (ACP), les techniques de régressions, les arbres de décision, les méthodes de classification et de reconnaissance de formes, les classificateurs statistiques et les réseaux de neurones artificiels [54], [146] relèvent de ce type de méthode. Afin d'orienter notre choix sur une de ces méthodes, nous avons réalisé une étude comparative selon : la nature des événements observés (quantitatif, qualitatif)[147], le degré de généralité de la méthode, la capacité à être utilisée dans un objectif de prédiction et d'explication des événements qualitatifs et quantitatifs, voir Tableau 1 .

explicatives → ↓ à expliquer	1 quantitative (covariable)	n quantitatives (covariables)	1 qualitative (facteur)	n qualitatives (facteurs)	mélange
1 quantitative	rég. linéaire simple, régression robuste, arbres de décision	rég. linéaire multiple, rég. robuste, PLS, arbres, réseaux de neurones	ANOVA, arbres de décision	ANOVA, arbres de décision, réseaux de neurones	ANCOVA, arbres de décision, réseaux de neurones
n quantitatives (représentent des quantités ≠)	régression PLS2	régression PLS2, réseaux de neurones	MANOVA	MANOVA, réseaux de neurones	MANCOVA, réseaux de neurones
1 qualitative nominale ou binaire	ADL, régression logistique, arbres de décision	ADL, rég. logistique, reg. logistique PLS, arbres, réseaux de neurones, SVM	régression logistique, DISQUAL, arbres	régression logistique, DISQUAL, arbres, réseaux de neurones	régression logistique, arbres, réseaux de neurones
1 discrète (comptage)	modèle linéaire généralisé (régression de Poisson, modèle log-linéaire)				
1 quantitative asymétrique	modèle linéaire généralisé (régressions gamma et log-normale)				
1 qualitative ordinale	régression logistique ordinale (au moins 3 niveaux)				
n quantitatives ou qualitatives	modèle à mesures répétées (les n variables représentent des mesures répétées d'une même quantité)				

Tableau 1 : Classification de méthodes basées sur des données historiques selon la nature des évènements[147]

A partir du Tableau 1, on peut constater que les méthodes tel que les réseaux de neurones, l'arbre de décision, l'analyse de covariance (ANCOVA) et la régression logistique multinomiale offrent une bonne capacité d'explication (qualitatif) et de prédiction (quantitatif) d'un évènement à partir d'une ensemble d'évènements de natures variés. Dans le cadre de nos travaux, nous avons retenu la méthode de l'arbre de décision[117] car elle offre à la fois la capacité de classier les influences entre les événements, de prédire et d'expliquer des évènements quantitatifs et qualitatifs et à travailler sur un ensemble d'évènements de natures variées. Cette méthode présente également la capacité de générer directement des règles de prédictions à partir de ses feuilles. Ces règles peuvent exprimer les risques de défaillance future des systèmes considérés ainsi que les durées de vie utile restante (RUL).

2. Fonction d'élaborations de conseils pour la maintenance

La fonction d'élaboration de conseils, dresse au sein d'un système un bilan de surveillance issue de ses fonctions de diagnostic et pronostic, afin d'assister l'opérateur de maintenance dans le choix des actions de maintenance à entreprendre. Dans le contexte de nos travaux, cette fonction doit modéliser non seulement l'élaborations de conseils de chacun des sous-systèmes et systèmes mobiles, mais aussi traiter le niveau de la flotte. Cette fonction de surveillance doit représenter la partie visible (résultat) de la surveillance de la flotte pour les acteurs en charge des systèmes sous surveillance. Elle doit donc élaborer et partager les recommandations sur l'état de santé actuel et prévisionnel des systèmes mobile de la flotte, les opérations de maintenance à réaliser au moment opportun, les configurations à adopter pour que les systèmes mobiles de la flotte puissent achever leurs missions en évitant la panne le tout en se conformant aux objectif fixés en termes de coûts.

Dans la littérature, différents modèles de stratégie de maintenance sont présentés, par exemple [121], [122] et [123]. Ces contributions spécifient les étapes de définition des objectifs stratégiques pour le service de maintenance, l'identification des indicateurs de performance (KPI) pertinents, l'évaluation l'état actuel de la maintenance, la définition des objectifs pour chaque indicateur de performance (KPI) et l'élaboration d'un plan d'action. La Figure 2 illustre le modèle de Salonen[123] utilisé pour la formulation des stratégies de maintenance. Ce modèle comprend toutes les étapes mentionnées précédemment. De plus, pour l'analyse des écarts identifiés entre les états actuels et souhaités, le modèle propose l'utilisation de l'analyse S.W.O.T(*Strengths, Weaknesses, Opportunities, and Threats*). Néanmoins, ce modèle ne traite que le volet management et coût de la maintenance. Les résultats issus de la surveillance classique et dynamique ne sont intégrés, ce qui rend potentiellement la politique de maintenance inadaptée à l'évolution des états réels des systèmes mobiles sous surveillance.

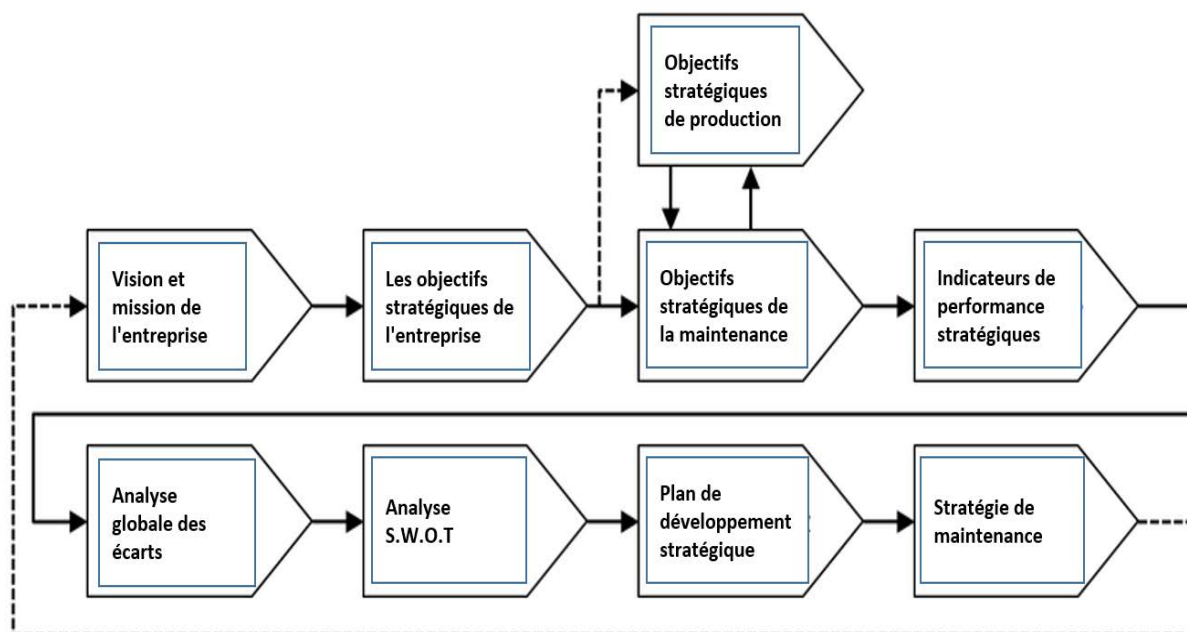


Figure 2: vue schématique du processus de la formulation de la stratégie de maintenance[123]

Dans [21], [124]–[127], les auteurs proposent des politiques de maintenance appliquées aux systèmes multi-composants (système composés de ses sous-systèmes). Ces politiques basées principalement sur le regroupement des interventions visent à réduire le coût de maintenance global d'un système[21]. Leurs processus de décision pour les déclenchements des opérations de maintenance se distinguent principalement au travers de deux catégories : statique ou dynamique. **Les politiques de regroupement statique** planifient les opérations de maintenance sur un horizon long terme en minimisant un critère donné, généralement un critère de coût. Ces politiques se basent exclusivement sur le comportement a priori des composants. La décision de maintenance se base sur des modèles de durée de vie ou de dégradation disponibles et raisonnent en termes de probabilité d'occurrence de défaillances prévisionnelles. En conséquence, les règles de maintenance restent statiques pendant la durée de vie du système. **Les politiques de regroupements dynamique** se base sur le suivi du niveau de dégradation des composants. Ainsi, selon cette politique, la décision de maintenance s'appuie sur les modèles de défaillance et les informations issue de la surveillance classique et dynamique des systèmes.

Nous avons retenu dans le cadre de nos travaux le modèle de politiques de regroupements dynamique car il permet de prendre en compte le niveau d'information propre à chaque composant contrairement aux politiques statiques qui ne permettent pas de tenir compte de l'état réel des composants. Ces politiques dynamiques permettent de développer une méthode de planification adaptée à l'utilisation réelle des composants et de profiter des opportunités (remplacements, des inspections, des pannes, etc.) de maintenance offertes pour notamment regrouper les opérations de maintenance et ainsi économiser des coûts logistiques.

Parmi les modèles de politiques de regroupements dynamique, nous nous intéressons plus particulièrement au modèle proposé dans[21] car il offre plus d'autonomie aux systèmes sous surveillance pour fournir des opportunités de maintenance, contrairement à la plupart des politiques de regroupement dynamiques où le regroupement est effectué autour des dates de maintenance des composants[21]. La mise en place du modèle retenu dans le cadre de nos travaux est basée sur la MFPO « Maintenance Free Operating Period » qui permet de définir des périodes à la fin de chaque mission où le système peut être maintenu. L'auteur a amélioré la politique de maintenance de la MFOP relativement aux processus de décision, d'évaluation et d'optimisation de maintenance pour qu'elle soit capable d'intégrer les informations disponibles en ligne pour adapter par conséquent la décision de maintenance.

Annexe C : choix et définition d'une grammaire et d'un automate pour le système expert

Dans cette annexe, nous étudions le domaine de la théorie des langages afin de choisir et définir des formalismes capables de décrire et de reconnaître les opérations à effectuer sur les SurfEvents, opérations qui sont de type logiques, mathématiques, textuelles, temporaires. Notre choix de sélection est basé sur la classification de Noam CHOMSKY[148], qui introduit des critères plus ou moins restrictifs sur la forme des règles d'une grammaire pour obtenir des classes de grammaires hiérarchiques, ordonnées par inclusion.

Un langage peut être décrit par un certain nombre de règles appelées "grammaire". L'objectif est de donner une description précise des règles de construction des expressions correctes du langage. La procédure (un algorithme) pour déterminer si une expression donnée appartient à une langue est appelée "Automate". Par la suite, nous présenterons la classification CHOMSKY.

Définition : une grammaire est un quadruplet $G = (T, N, S, R)$ tel que

- T est le vocabulaire terminal, c'est-à-dire l'alphabet sur lequel est défini la langue.
- N est le vocabulaire non terminal, c'est-à-dire l'ensemble des symboles qui n'apparaissent pas dans les expressions générées, mais qui sont utilisés pendant la génération. Un symbole non terminal fait référence à une « catégorie syntaxique ».
- R est un ensemble de règles appelé « réécriture » de la grammaire, ils sont de forme :
$$u1 \rightarrow u2, \quad \text{avec } u1 \in (N \cup T)^+ \text{ et } u2 \in (N \cup T)^*$$
- $S \in N$ est le symbole ou l'axiome de départ. C'est à partir de ce symbole non-terminal que nous commencerons la génération d'expressions au moyen des règles de la grammaire.

Types de grammaires:

- Type 0: grammaires générales. Aucune restriction sur les règles.
- Type 1: grammaires contextuelles. Les règles de « R » sont de la forme:

$$uAv \rightarrow uvw \text{ avec } A \in N, u, v \in (N \cup T)^* \text{ et } w \in (N \cup T)^+$$

En d'autres termes, le symbole non terminal A est remplacé par w si nous avons les contextes u à gauche et v à droite.

- Type 2: Grammaires sans contextes ou algébrique. Les règles de « R » sont de la forme:

$$A \rightarrow w, \text{ avec } A \in N \text{ et } w \in (N \cup T)^*$$

En d'autres termes, le côté gauche de chaque règle est constitué d'un seul symbole non terminal.

- Type 3: Grammaire régulières
 - À droite. Les règles de « R » sont de la forme :

$$A \rightarrow aB \text{ ou } A \rightarrow a \text{ avec } A, B \in N \text{ et } a \in T$$
 - À gauche. Les règles de « R » sont de la forme :

$$A \rightarrow Ba \text{ ou } A \rightarrow a \text{ avec } A, B \in N \text{ et } a \in T$$

Dans la classification de Noam CHOMSKY, les grammaires sont associées au type de langages et d'automates comme suit (voir tableau 2):

Grammaire	Langue générée	Automate de reconnaissance
Type 0	Toutes les langues reconnues « récursivement énumérables »	Machine de Turing
Type 1	Contextuelle	Automate linéairement borné
Type 2	Non contextuelle « algébrique »	Automate à pile
Type 3	Régulière	Automate à états finis

Tableau 2 : classification de Noam CHOMSKY des grammaires

Ces langages sont classés par inclusion: l'ensemble des langages générés par les grammaires de type n est strictement inclus dans celui des grammaires de type $n-1$ (pour $n \in \{1,2,3\}$). La Figure 3 suivant illustre ces relations d'inclusion.

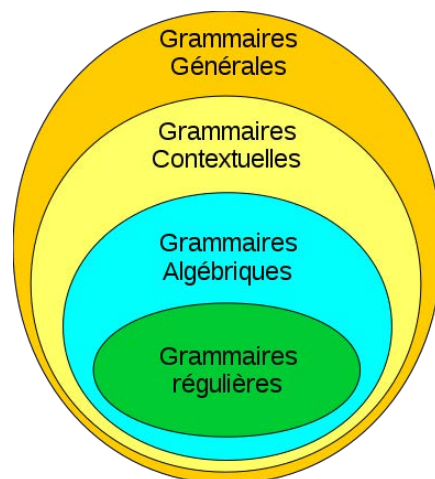


Figure 3: Classification of Noam CHOMSKY[149]

Dans nos travaux, la grammaire sans contexte (type 2) reconnue par son automate à pile est retenue pour définir et reconnaître les différentes expressions logiques, mathématiques, textuelles et temporelles des SurfEvents (voir Tableau 2). La raison principale est qu'elle admet moins de

restriction que la grammaire régulière et plus de contrôle que les grammaires de type 1 et 0. Nous spécifions ci-dessous notre grammaire.

Supposons les notations suivantes:

Token	Commentaire
Exp	Expression d'un SurfEvent
Lexp	Expression logique d'un SurfEvent
Mexp	Expression mathématique d'un SurfEvent
Dexp	Expression temporaire d'un SurfEvent
Texp	Expression logique d'expression textuelle d'un SurfEvent
FL	Fonction logique d'Expression logique d'un SurfEvent
FT	Fonction textuelle d'Expression logique d'un SurfEvent
FD	Fonction temporaire d'Expression logique d'un SurfEvent
FM	Fonction mathématique d'Expression logique d'un SurfEvent

Tableau 2 : notations pour la grammaire choisie

Considérons $G' = (T, N, S, R)$ une grammaire sans contexte qui définit les règles de fonctionnement logique, mathématique, textuelle et temporaire des SurfEvents comme suit:

$T = \{ (,), +, -, *, \%, [0..9], ., \cup, \setminus, \in, \exists, \emptyset, <, \leq, >, \geq, =, \neq, Or, And, \neg, True, False, ?, Event, EventList, [a \dots z], [A..Z], \varepsilon, FL, FT, FD, FM \}$

$N = \{ Exp, Lexp, Mexp, Dexp, Texp, FL, FT, FD, FM, Op1, Op2, Op3, Id1, Id2, Id3, Digit, Text, Ob \}$

$S = Exp$

$\varepsilon =$ designe un SurfEvent directement obtenu à l'aide d'un capteur

$R = \{$

$Exp \rightarrow Lexp \mid Mexp \mid Dexp \mid Texp \mid \varepsilon$

$Mexp \rightarrow Id1 \mid (Mexp) \mid Mexp \ Op1 \ Mexp \mid FM$

$Op1 \rightarrow + \mid - \mid * \mid / \mid \%$

$Id1 \rightarrow Digit \ Id1 \mid Digit$

$Digit \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \mid . \mid ,$

$Texp \rightarrow Id2 \mid (Texp) \mid Texp \ Op2 \ Texp \mid FT$

$Op2 \rightarrow \cup \mid \cap \mid \in \mid \setminus \mid \exists \mid \emptyset$

$Id2 \rightarrow Text \ Id2 \mid Text$

$Text \rightarrow a \mid b \mid c \mid d \mid e \mid \dots \mid x \mid y \mid z \mid A \mid B \mid C \mid D \mid F \mid G \mid H \mid I \mid J \mid K \mid L \mid M \mid N \mid O \mid P \mid Q \mid R \mid S \mid T \mid U \mid V \mid W \mid X \mid Y \mid Z$

$Dexp \rightarrow (Dexp) \mid FD$

$Lexp \rightarrow Id3 \mid (Lexp) \mid Lexp \ Op3 \ Lexp \mid Texp \ Op3 \ Texp \mid Mexp \ Op3 \ Mexp \mid FL$

$Op3 \rightarrow < \mid \leq \mid > \mid \geq \mid = \mid \neq \mid And \mid Or \mid \neg \mid$

$Id3 \rightarrow True \mid False \mid 0 \mid 1 \mid ?$

$FM \rightarrow$

$Exp(Ob) \mid Avg(Ob) \mid Count(Ob) \mid Max(Ob) \mid Min(Ob) \mid Ob. Single() \mid Sum(Ob) \mid Abs(Ob) \mid$

$Sin(Ob) \mid Cos(Ob) \mid Floor(Ob) \mid Log(Ob) \mid Log10(Ob) \mid Power(Ob, Id1) \mid Round(Ob) \mid Sqr(Ob) \mid$

$Tan(Ob) \mid Sign(Ob) \mid Round(Ob, Digit)$

$Ob \rightarrow Event \mid EventList$

$FL \rightarrow Iif(Lexp, Exp, Exp) \mid IsNull(Ob) \mid IsNullOrEmpty(Ob) \mid Exists(Ob)$

$FT \rightarrow$

$Ascii(Text) \mid Char(Id1) \mid Contains(Id2, Id2) \mid EndsWith(Id2, Id2) \mid Remove(Id2, Digit)$

$\mid Reverse(Id2) \mid StartsWith(Id2, Id2) \mid Trim(Id2) \mid Upper(Id2)$

$FD \rightarrow$

$DateTime \mid Between(DateTime, DateTime) \mid AddDays(DateTime, Id1)$

$\mid AddHours(DateTime, Id1) \mid AddMilliseconds(DateTime, Id1)$

$\mid AddMinutes(DateTime, Id1) \mid AddMonths(DateTime, Id1) \mid$

$AddSeconds(DateTime, Id1) \mid AddTimeSpan(DateTime, Id1) \mid AddYears(DateTime, Id1) \mid$

```

DateDiffHour(DateTime, DateTime)|GetDay(DateTime)|GetHour(DateTime)|
GetYear(DateTime)|Now()|Today()|UtcNow()
}

```

A partir de la grammaire G' définie ci-dessus, nous proposons un automate à pile $A = (T', P', Q', M', S')$ reconnaissant la même langue comme suit:

$$\begin{aligned}
T' &= T \\
P' &= T \cup N \cup \{\perp\} \\
Q' &= \{p', q'\} \text{ tels que } p' \text{ et } q' \text{ ne sont pas des symboles de } T \cup N \\
M' &= \{(p', \varepsilon, \perp) \rightarrow (q', [stack(S)]) \\
&\quad \cup \{(q', \varepsilon, B') \rightarrow (q', [unstack(B'), stack(S_n), \dots, stack(S_2), stack(S_1)]) / (B' \\
&\quad \rightarrow u = S_1 S_2 \dots S_n) \in R\}\} \\
&\quad \cup \{(q', a', a') \rightarrow (q', [unstack(a')]) / a' \in T\} \\
S' &= p'
\end{aligned}$$

L'automate à pile « A » fonctionne comme suit:

- Si le symbole en haut de la pile est un symbole non terminal B' , il est remplacé par un règle u tel qu'il existe une règle $B' = u$ dans R .
- Si le symbole en haut de la pile est un symbole terminal a' , et que le symbole en haut de l'expression à analyser est un a' , alors nous désempilons a' de la pile, et supprimons un mot à analyser.

Annexe D : choix d'une approche d'implémentation de l'architecture EMH²

Dans cette section, nous présenterons les différentes approche d'architectures pouvant supporter le développement des différents Niveaux Holonique de l'architecture EMH². Nous précisons que notre étude comparative de ces approche d'architectures n'est pas absolue, elle repose uniquement sur un besoin en réactivité, flexibilité, modularité, et évolutivité. À partir des contributions examinées dans la littérature, nous avons recensé trois approches: architecture monolithique ; architecture orientée services et architecture de microservice :

- **Architecture monolithique** : elle constitue une approche traditionnelle (voir Figure 4). Ce modèle est composé d'une module d'interface utilisateur, d'un module logique métier et d'une logique permettant l'accès aux données. Les composants de la solution sont interconnectés et interdépendants plutôt que faiblement couplés, comme dans le cas des solutions modulaires. Les avantages de cette approche d'architecture sont qu'une solution monolithique offre une capacité d'abstraction facilitant une déploiement cohérent et rapide. Les inconvénients majeurs sont liés à son manque de modularité et de flexibilité et à une forte complexité résultante. En effet, il est possible d'envisager des situations où aucun développeur (ou groupe de développeurs) ne peut comprendre la totalité de la solution, assurer la réutilisation de ses composants, et assurer l'évolutivité qui devient difficile. L'intégration ou le déploiement voit sa rapidement complexité accrue et toute modification prend du temps. Il devient difficile de changer de technologie, de langage ou de plateforme

parce que l'intégration est très forte (tout est étroitement couplé et dépendent les uns des autres) [150].

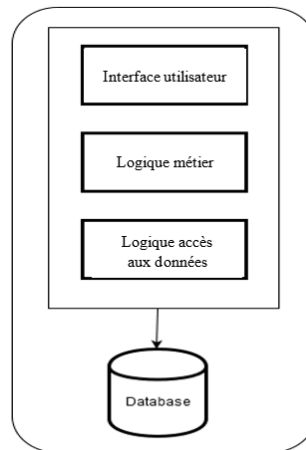


Figure 4: Architecture monolithique

- **Architecture orientée services** : cette approche d'architecture plus récente est dénommée SOA (Services oriented architecture). Elle favorise un couplage moyen entre ses composants afin d'optimiser leurs réutilisations[125]. Elle présente les caractéristiques suivantes:
 - Les services sont des composants ayant des contrats ou interfaces publiés et les consommateurs peuvent découvrir ces services de manière dynamique (voir Figure 5).
 - Les langages XML (Extensible Markup Language) et SOAP (Simple Object Access Protocol) sont des formats adoptés par l'approche SOA afin de fournir un format d'échange unifié entre les composants.

Les avantages de cette approche d'architecture sont la flexibilité, la facilité de mettre en place des tests et les débogages sont plus faciles. Ce type d'architecture présente une bonne évolutivité et une facilité de réutilisation de ses composants. Cependant, l'approche SOA ne convient pas aux solutions nécessitant un échange de données important, régulier. De plus, les solutions nécessitant une communication en temps réel sont difficilement implémentables avec la logique SOA.

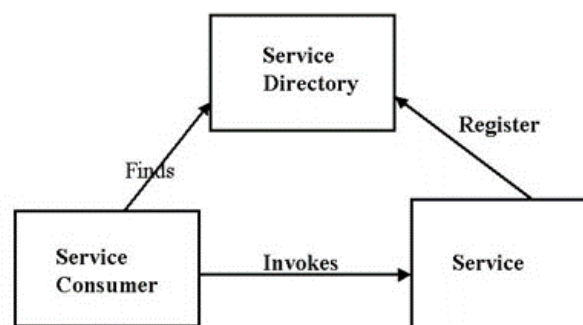


Figure 5: Architecture orientée services

- Architecture de microservice** : cette approche d'architecture offre des solutions distribuées dans laquelle tous ses modules sont des microservices (voir Figure 6). Chaque microservice étant « léger » (petit en terme de taille mémoire et codes), ils sont faiblement couplés et fortement dédiés pour une fonctionnalité spécifique[138]. Un microservice communique à l'aide de REST (Representational State Transfer), de Message Queue (MQ) ou de tout autre interface de communication. Les avantages de cette approche d'architecture sont qu'elle permet une mise à l'échelle (évolutivité) aisée, un traitement en temps réel, une modularité et un déploiement simple et flexible. L'inconvénient majeur de cette approche est sa complexité de son maintien dans un environnement où les sources de données dispersés, manquent de consistance [151].

Malgré ses inconvénients, l'approche d'architecture par microservices offre d'énormes avantages qui l'emportent sur les deux autres approches et a donc été retenue pour implémenter notre architecture de surveillance EMH².

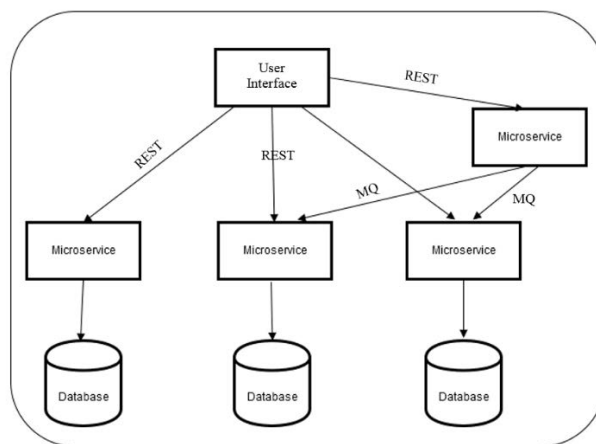


Figure 6: Architecture microservice

AHMAT FADIL, ADOUM.

Proposition d'une architecture de surveillance Holonique pour l'aide à la maintenance proactive
d'une flotte de systèmes mobiles : application au domaine ferroviaire

Directeur de thèse : Pr. Damien Trentesaux, Université Polytechnique Hauts de France

Co-encadrant de thèse : MCF. Abdelghani Bekrarm Université Polytechnique Hauts de France

Spécialité : Automatique et Génie Informatique

Résumé : La maintenance de flottes de systèmes mobiles dans le monde du transport et de la logistique revêt de nos jours une importance croissante de par l'augmentation des attentes des exploitants et opérateurs en termes de sécurité, de fiabilité, de suivi, de diagnostic et de maintenance de ces systèmes. Dans ce contexte, Les mainteneurs des flottes doivent souvent faire face à d'énormes quantités de données brutes, informations et événements de surveillance liés aux contextes de leurs systèmes. De plus, ces événements, données et informations manquent souvent de précision et sont souvent contradictoires ou obsolètes. Enfin, le degré d'urgence des décisions de maintenance est rarement pris en compte. Ce travail est consacré à la proposition et à la mise au point d'une architecture de surveillance pour l'aide à la maintenance d'une flotte de systèmes mobiles. Cette architecture, appelée EMH², est destinée à faciliter le diagnostic et le suivi de ce type de flotte. Elle est construite sur les principes holoniques, des plus bas (capteurs) aux plus hauts niveaux (ensemble d'une flotte de systèmes mobiles). Elle se base également sur une standardisation des événements traités afin de traiter les données de manière générique. Cette architecture, indépendante des types de systèmes surveillés et de leur niveau hiérarchique, peut devenir l'épine dorsale d'une stratégie efficace de maintenance proactive d'une flotte. Une méthodologie de déploiement est ainsi proposée. Une étude en simulation et une application sur une flotte de 10 trains actuellement en service est présentée.

Mots-clés : Système de gestion d'événements; Architecture holonique, Surveillance, Maintenance Proactive, flotte de systèmes mobiles, générations de connaissances, systèmes de transport ferroviaire.

Abstract: The maintenance of mobile systems fleets in the world of transport and logistics is of increasing importance today due to the increasing expectations of operators in terms of safety, reliability, monitoring, diagnosis and maintenance of these systems. In this context, fleet maintainers often have to deal with huge amounts of raw data, information and monitoring events related to the context of their systems. Moreover, these events, data and information are often lack precision and often contradictory or obsolete. Finally, the urgency of maintenance decisions is rarely taken into account. This work is devoted to the proposal and the development of a monitoring architecture to help maintain a fleet of mobile systems. This architecture, called EMH², is intended to facilitate the diagnosis and monitoring of this type of fleet. It is built on holonic principles, from the lowest (sensors) to the highest levels (a whole fleet of mobile systems). It is also based on a standardization of processed events in order to process the data generically. This architecture, independent of the types of systems monitored and their hierarchical level, can become the backbone of an effective

strategy for proactive fleet maintenance. A deployment methodology is thus proposed. A simulation study and an application on a fleet of 10 trains currently in service is presented.

Key words: Event management system; Holonic Architecture, monitoring, Proactive Maintenance, Fleet of Mobile Systems, Knowledge Generation, railway transportation systems.

Attribution - Pas d'Utilisation Commerciale 4.0 International

