



HAL
open science

Prédiction de l'évolution d'un système complexe dans un contexte non déterministe: architecture basée sur le couplage SMA / RàPC et la clusterisation.

Bruno Perez

► To cite this version:

Bruno Perez. Prédiction de l'évolution d'un système complexe dans un contexte non déterministe: architecture basée sur le couplage SMA / RàPC et la clusterisation.. Intelligence artificielle [cs.AI]. Université Bourgogne Franche-Comté, 2021. Français. NNT: 2021UBFCD036 . tel-03545248

HAL Id: tel-03545248

<https://theses.hal.science/tel-03545248v1>

Submitted on 27 Jan 2022

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.

SPIM

Thèse de Doctorat

UBFC



école doctorale **SPIM** sciences pour l'ingénieur et microtechniques
UNIVERSITÉ DE BOURGOGNE FRANCHE - COMTE

Prédiction de l'évolution d'un système complexe dans un contexte non déterministe : architecture basée sur le couplage SMA/RàPC et la clusterisation.

■ Bruno PEREZ

N°



**UNIVERSITÉ DE
FRANCHE-COMTÉ**

THÈSE présentée par

Bruno PEREZ

pour obtenir le
Grade de Docteur de
l'Université de Bourgogne/Franche-Comté

Spécialité : **Informatique**

**Prédiction de l'évolution d'un système complexe dans un
contexte non déterministe : architecture basée sur le couplage
SMA/RàPC et la clusterisation**

Unité de Recherche :
Femto-st/DISC

Soutenue publiquement le 21/09/2021 à Besançon,
devant le Jury composé de :

Julie DUGDALE	Rapporteur	MCF HDR, Université de Grenoble Alpes (UGA)
Stéphane CHRÉTIEN	Rapporteur	Prof., Université Lyon2
Lionel DROZ-BARTHOLET	Examineur	Ing. Recherche, MAINCARE Solutions Besançon
Christophe CAMBIER	Examineur	MCF HDR, Université de La Sorbonne UPMC-IRD-UCAD
Frédéric AUBER	Examineur	Prof., Centre Hospitalier Régional Universitaire de Besançon (CHRUB)
Laurent PHILIPPE	Directeur	Prof., Université de Bourgogne Franche-Comté
Julien HENRIET	Co-encadrant	MCF HDR, Université de Bourgogne Franche-Comté
Christophe LANG	Co-encadrant	MCF, Université de Bourgogne Franche-Comté

Remerciements

Cette étape de remerciements est empreinte d'une légère pression tant il est difficile de citer de manière exhaustive toutes les personnes qui de près ou de loin ont été un soutien pour ce travail de thèse.

Je remercie tout d'abord Julie DUGDALE et Stéphane CHRETIEN pour avoir accepté de rapporter mon travail ainsi que l'ensemble des membres du jury pour avoir accepté de l'évaluer.

Je remercie Laurent Philippe, mon directeur de thèse qui a su distiller avec une grande subtilité ce que l'on attendait d'un travail de recherche. Ce furent de beaux échanges qui m'ont grandement enrichi et ce malgré mon grand âge.

Je remercie Julien Henriet et Christophe Lang pour la qualité de leur encadrement. Au-delà de vos précieux conseils, j'ai ressenti tout au long de ce travail de thèse une bienveillance scientifique et humaine.

Je remercie les membres de l'équipe du DISC (Département d'Informatique des Systèmes Complexes) pour leur accueil chaleureux et pour tous les bons conseils indispensables pour passer une bonne journée : fonctionnement de la machine à café, bon plan pour la restauration ... Merci à vous pour les échanges que nous avons eu et au travers desquels j'ai pu profiter de votre expertise scientifique.

Je remercie Frédéric Aubert, Professeur et chef de service de chirurgie pédiatrique du CHRU (Centre Hospitalier Régional Universitaire) de Besançon qui est à l'origine de ce travail de thèse intégré dans le projet MASSAI (Modeling Aids for Safe Surgery using Artificial Intelligence). Ton enthousiasme et la pertinence de tes choix pour la mise en application du simulateur furent pour moi un moteur.

Je remercie également les personnels du CHRU de Besançon en espérant n'oublier personne : Docteur Amélie Jurine, Docteur Christelle Studer, Docteur Loic Barrot, tous, médecins anesthésistes qui m'ont aidé pour la typologie des risques anesthésiques et l'identification des patients, Melle Mélina Noël, étudiante ISIFC (Institut Supérieur d'Ingénieurs de Franche Comté), qui a fait les premières extractions de données à partir du logiciel AMI destiné au bloc, Docteur David Ferreira, pour son aide lors l'extraction de ces données, Docteur Nicolas Boulard, qui m'a aidé pour la typologie des risques chirurgicaux, Mme Alexandra Monot, du Service Informatique du CHU qui m'a beaucoup aidé au début de ce travail, Monsieur Alex Lopez, PDG de l'entreprise "AEGLE" (qui développe le logiciel AMI), Jérémy Gutierrez, étudiant ISIFC qui travaille actuellement sur l'environnement du bloc, Lorédane Salvi qui travaille

actuellement sur MASSAI et tous les "inconnus" du bloc de chirurgie pédiatrique ou du service biomédical qui ont participé comme "cobaye" ou conseils pour le développement du projet.

Je remercie la région Bourgogne-Franche-Comté qui a permis l'obtention du financement de ce projet de recherche grâce aux Fonds Européen de Développement Régional (FEDER).

Je remercie enfin mes enfants et en particulier mon épouse qui m'a soutenu et suivi dans cette aventure quelque peu atypique au regard de mon parcours.

Table des matières

1	Introduction	8
1.1	Contexte	9
1.2	Problématique et contribution	9
1.3	Plan	11
I	Contexte d'étude, contexte scientifique et état de l'art	12
2	Contexte d'étude : Survenue d'évènements indésirables au bloc opératoire	14
2.1	Introduction	15
2.2	Facteurs de risques liés au patient	16
2.3	Facteurs liés au personnel	17
2.4	Facteurs liés à l'environnement	17
2.5	Facteurs liés au processus opératoire	18
2.6	Bilan du chapitre	21
3	Contexte scientifique et état de l'art	24
3.1	Introduction	25
3.2	Prognostic and Health Managment	25
3.2.1	Architecture du PHM	26
3.2.2	Approche du PHM au bloc	27
3.2.3	Bilan de l'approche PHM	28
3.3	La modélisation SMA	28
3.3.1	Fondamentaux de la modélisation	29
3.3.2	La notion d'agent	30
3.3.3	Le système multi-agents	32
3.3.4	Analyse et bilan d'une modélisation SMA	37
3.4	Les systèmes à base de connaissances	38
3.4.1	Les réseaux de neurones artificiels	38
3.4.2	Le raisonnement à partir de cas	41
3.4.3	Couplage avec RàPC : quelques exemples significatifs	47

3.5	Les méthodes de clusterisation	48
3.5.1	Méthode basée sur la <i>Partition</i>	49
3.5.2	Méthode basée sur la <i>Hiérarchie</i>	50
3.5.3	Méthode basée sur la <i>Théorie floue</i>	51
3.5.4	Méthode basée sur la <i>Distribution</i>	51
3.5.5	Méthode basée sur les <i>Noyaux</i>	52
3.5.6	Méthode basée sur la théorie des <i>Graphes spectraux</i>	52
3.5.7	Méthode basée sur l' <i>Affinité de propagation</i>	52
3.5.8	Bilan des méthodes de clusterisation	53
3.6	Bilan du chapitre	53

II Nos contributions 55

4	Couplage SMA/RàPC : une architecture comme réponse à la prévention des risques dans un bloc opératoire 57
4.1	Introduction 59
4.2	Modélisation SMA : intérêts et motivations 59
4.2.1	Les systèmes complexes 59
4.2.2	Modèle et modélisation 60
4.2.3	Simulation à base d'agents 61
4.2.4	Limite d'une modélisation SMA 62
4.3	RàPC : un apprentissage par analogie 63
4.4	Modélisation SMA : formalisme et description du modèle 64
4.4.1	Modèle SMA pour le bloc opératoire : formalisme 65
4.4.2	Description du Modèle SMA pour le bloc opératoire : protocole ODD . 67
4.4.2.1	Vue globale : Overview 67
4.4.2.2	Éléments de conception : Design concepts 69
4.4.2.3	Détails 70
4.4.2.4	Description des agents BDI 71
4.5	Modèle RàPC pour le bloc opératoire 74
4.5.1	Élaboration 74
4.5.2	Remémoration 75
4.5.3	Réutilisation 76
4.5.4	Révision 77
4.5.5	Mémorisation 77
4.6	Couplage SMA/RàPC 77
4.6.1	Vision globale du couplage SMA/RàPC 77
4.6.2	SMA : une organisation holonique 78
4.6.3	Organisation de la mémoire collective 79

4.7	Synthèse de la présentation de l'architecture	81
4.8	Simulateur pour MASSAI	82
4.8.1	Plateforme de simulation CoupSR_PredRB : un outil de prédiction des risques au bloc	82
4.8.2	Exploitation des données : une approche par analogie	84
4.8.3	Simulateur et émergence	85
4.8.4	Comparaison <i>CoupSR_PredRB</i> avec avec des plateformes simulant la salle d'opération	86
4.9	Bilan du chapitre	88
5	Clusterisation : comparatif entre plusieurs méthodes	90
5.1	Introduction	91
5.2	Clusterisation : une méthode pour identifier des comportements similaires . . .	92
5.2.1	Un regroupement non supervisé	92
5.2.2	<i>similVar</i> et <i>similVar+</i> : des outils d'analyse d'interactions	94
5.2.3	<i>similVar</i> : Clusterisation des données pour évaluer les interactions entre agents	94
5.2.4	<i>similVar+</i> : Clusterisation des données pour évaluer les interactions entre agents en décalage dans le temps	97
5.3	Résultats	99
5.3.1	Méthode	101
5.3.2	Présentation des résultats	103
5.3.3	Discussion	104
5.4	Bilan du chapitre	108
6	Analyse des interactions entre agents et prédictivité de l'évolution de l'état du sys- tème	109
6.1	Introduction	110
6.2	Évaluation de la plausibilité des interactions entre les attributs	110
6.2.1	Matrice de corrélation : matrice d'adjacence	111
6.2.2	Détermination des liaisons entre attributs à partir de la matrice d'adjacence	113
6.3	Prédictivité	114
6.4	Résultats	117
6.4.1	Plausibilité des influences résultant d'interactions entre agents	118
6.4.2	Prédictivité de l'évolution de l'état du système	125
6.5	Bilan du chapitre	127
7	Conclusion	129
7.1	Contexte	130
7.2	Contributions	130

7.3	Résultats	131
7.4	Perspectives	132

Chapitre 1

Introduction

Sommaire

1.1	Contexte	9
1.2	Problématique et contribution	9
1.3	Plan	11

1.1 Contexte

Les interventions chirurgicales au sein des centres hospitaliers sont effectuées dans un contexte médical de plus en plus innovant et performant, multipliant ainsi la complexité des actes chirurgicaux et les risques induits en cours d'opération. Une intervention chirurgicale ne peut être dissociée de la sécurité du patient, et donc des risques induits par un environnement technique de plus en plus sophistiqué et pluridisciplinaire. La sécurité des patients est en conséquence un objectif ancré au cœur des pratiques des professionnels et du système de santé. Ainsi, elle a engendré des protocoles de sécurité standards relativement efficaces, mis en œuvre entre autres par l'Organisation Mondiale pour la Santé (OMS).

Le projet MASSAI (Modeling Aids for Safe Surgery using Artificial Intelligence) s'inscrit dans cette logique de sécurisation du bloc opératoire. Il a été initié par Frédéric Auber, Professeur et chef du service de chirurgie pédiatrique au CHRU (Centre Hospitalier Régional Universitaire) de Besançon, en collaboration avec l'équipe DISC (Département d'Informatique des Systèmes Complexes) du laboratoire FEMTO-ST (Franche-Comté Electronique Mécanique Thermique et Optique – Sciences et Technologies). Ce projet intègre notre travail de recherche dont la finalité vise à sécuriser le bloc opératoire dans un contexte complexe qui nécessite la gestion simultanée de nombreux éléments (ressources matérielles, ressources humaines, environnement technique ...). Il s'agit d'être en mesure de générer des alertes, de prédire un événement indésirable et d'en déterminer les éléments déclencheurs.

Le travail de recherche proposé dans cette thèse consiste donc à élaborer un modèle prédictif évaluant les risques induits par les actions et les états des différents éléments du bloc opératoire enrichi par des outils d'aide à la décision. L'important consiste au final à étoffer l'arsenal sécuritaire du bloc opératoire pour, in fine, épargner des vies.

1.2 Problématique et contribution

Si l'on considère le bloc opératoire comme un système complexe, il convient avant tout de définir ce dernier. Il n'existe pas de définition "normalisée" d'un système complexe, mais plutôt une convergence de propriétés le caractérisant. La définition proposée par Le Moigne [54] les récapitule assez clairement : *"Un système complexe est, par définition, un système construit par l'observateur qui s'y intéresse. Celui-ci postule la complexité du phénomène sans pour autant s'astreindre à croire à l'existence naturelle d'une telle propriété dans la nature ou dans la réalité. La complexité, qui implique toujours quelque forme d'imprévisibilité, ne peut aisément être tenue pour déterministe. De façon générale, en revanche, la complexité est représentée par un enchevêtrement d'interactions en inter-relations"*. Ainsi par exemple, que ce soient notre propre environnement physiologique caractérisé par nos cellules ou les réseaux de communication collectifs (réseaux sociaux), l'ensemble de ces entités interagissent entre elles et sont en constante évolution.

Comprendre de tels systèmes nécessite leur observation soumise à la représentation de la réalité avec le maximum de précision, sans pour autant compliquer exagérément le modèle. L'analyse des simulations qui en découle produit au moins deux niveaux d'observation. Le premier, qui peut être représenté (entre autres) par les entités réifiées sous forme d'agents, révèle les trajectoires, les interactions et les comportements individuels, alors que le second mesure les effets produits sur le système. Ces deux niveaux d'observation permettent d'établir une relation de causalité entre l'évolution des agents (comportements, interactions, trajectoires) dans un environnement donné, et l'évolution de l'état du système. Dès lors, notre questionnement scientifique porte sur la possibilité de prédire, d'anticiper et d'analyser les risques dans un système complexe pourvu d'entités non déterministes. La littérature nous montre qu'il existe dans le monde industriel des solutions très abouties en maintenance préventive. Le paradigme Prognostics and Health Management (PHM) fait l'objet de recherches actives et offre des outils permettant de proposer des solutions à la prédiction dans le domaine de l'industrie. Toutefois, ces solutions restent à notre connaissance cloisonnées au monde de l'industrie, des automates et semblent encore difficilement transposables à des univers non programmables tel que les environnements humains.

Afin d'aller au-delà de cette limite, nous avons choisi une modélisation multi-agents. En effet, la complexité des différents mécanismes et une approche globale caractérisent les systèmes multi-agents (SMA) : ils couvrent l'interactivité de tous les éléments matériels et immatériels impliqués dans un environnement, mais aussi les actions propres à chacun d'eux. Ainsi, nous espérons exploiter les propriétés d'autonomie et d'interaction des agents, pour aborder le concept d'émergence et d'adaptation en lien avec notre problématique. La thèse portant sur la prédiction d'une défaillance dans un système complexe, nous focalisons aussi notre réflexion sur l'intérêt du couplage du SMA avec d'autres paradigmes tels que le raisonnement à partir de cas (RàPC). Il s'agit, à travers cette association, de capitaliser sous forme de cas les simulations issues de notre modèle. Cela signifie dans un même temps que la recherche de solution à un problème (exemple : recherche du niveau de risque) consiste à réutiliser grâce à un raisonnement par analogie des solutions issues d'expériences passées (base d'expériences). Ainsi, trouver une solution, ne se limite pas au simple parcours d'une base de données. Cet outil supplémentaire pallie le défaut d'enrichissement des connaissances des systèmes multi-agents, indispensable à toutes les formes d'intelligences distribuées. Nous pensons que notre architecture qui couple SMA et RàPC est certes une contribution qui répond à notre problématique d'identification d'événements indésirables a priori imprévisibles, mais reste insuffisante en terme d'analyse. En effet, la détermination de seuils de risque et la prédictivité d'événements indésirables sont dépendantes de l'analyse de l'évolution des différentes variables du système. C'est pourquoi nous avons intégré deux outils d'aide à la décision au raisonnement à partir de cas. Le premier permet de classer les variations de nos variables sous forme de clusters. Nous pensons à ce sujet qu'une même variation entre deux variables peut traduire un même comportement et/ou une interaction entre elles. Le parcours de ces clusters est le second outil destiné à prédire l'évolution de notre système dans le temps. La viabilité de nos trois contributions (architecture, outils d'aide à la

décision) comme réponse à nos problématiques résulte des recherches et des expérimentations menées dans cette étude.

1.3 Plan

Nous avons structuré ce mémoire en deux parties. La première, composée de deux chapitres, présente notre contexte et l'état de l'art. Le Chapitre 2 consacré à l'univers du bloc opératoire décrit son environnement à travers le prisme des différents protocoles de sécurité mis en œuvre. Nous mettons aussi en évidence les causes possibles de survenue d'événements indésirables. Le Chapitre 3 décline quant à lui notre contexte scientifique et les différentes approches de la prédiction des risques extraites de la littérature. Il s'agit du PHM, de la modélisation multi-agents (SMA), du raisonnement à partir de cas (RàPC), des réseaux de neurones artificiels et enfin d'outils de clusterisation.

Dans la seconde partie de ce manuscrit, nous présentons nos contributions en réponse à notre problématique de prédiction des risques dans un contexte non déterministe, mais aussi en réponse aux verrous scientifiques qui y sont attachés. Le Chapitre 4 est axé sur notre architecture. Nous décrivons de manière formelle le couplage SMA/RàPC. Nous y exposons aussi notre implémentation qui est illustrée par une vision opérationnelle du simulateur. Nous comparons ensuite dans le Chapitre 5, différentes méthodes de clusterisation avec nos propres outils de regroupement de données. Cette phase a conforté l'implémentation de nos algorithmes (*similVar* et *similVar+*) destinés à la clusterisation des données. Le dernier chapitre (Chapitre 6) est au cœur de notre problématique sachant qu'il traite de l'analyse des interactions entre agents et de la prédictivité de l'évolution de l'état du système. Nous y défendons l'idée que notre simulateur est un modèle de prédiction qui va au-delà de la production d'alertes.

Les perspectives de ce type de modélisation prédictive, conceptualisées par une approche de couplage SMA/RaPC et enrichie par nos outils d'aide à la décision sont les arguments défendus en conclusion. Ces perspectives soulèvent essentiellement les questions liées aux notions d'émergence et de prédictivité dans un contexte réel : les données de notre modèle sont issues de différents capteurs intégrés dans le bloc opératoire. Peut-on dès lors continuer à considérer la simulation multi-agents comme un outil adapté ou faut-il le repenser ?

Première partie

Contexte d'étude, contexte scientifique et état de l'art

Chapitre 2

Contexte d'étude : Survenue d'évènements indésirables au bloc opératoire

Sommaire

2.1	Introduction	15
2.2	Facteurs de risques liés au patient	16
2.3	Facteurs liés au personnel	17
2.4	Facteurs liés à l'environnement	17
2.5	Facteurs liés au processus opératoire	18
2.6	Bilan du chapitre	21

2.1 Introduction

Le bloc opératoire est un lieu dédié à des actes invasifs, qui réunit toutes les compétences médicales pour assurer la sécurité des patients. Or, le nombre considérable d'interventions combiné à leur complexité peut produire des événements indésirables associés aux soins (EIAS). La sécurité des patients qui est une préoccupation majeure des praticiens nécessite donc des environnements qui intègrent cette spécificité. Or ce type d'implantation se heurte à l'optimisation des blocs opératoires qui doivent maintenir un taux d'occupation maximal. L'enjeu consiste donc à concevoir des structures à la fois polyvalentes tout en maintenant un niveau de sécurité maximum pour les patients [42]. Le bloc opératoire, qui comprend cinq secteurs principaux :

- la zone d'arrivée et de contrôle des patients et du personnel ;
- les locaux du personnel (vestiaires, services sanitaires, salles de repos) ;
- le bloc chirurgical qui inclut les salles d'opération ;
- la salle de réveil ;
- les locaux de soutien (local d'anesthésie, local de nettoyage et préparation des instruments, local d'entretien ménager) ;

doit être conçu selon ce principe de polyvalence avec un niveau très élevé de sécurité. Il est donc essentiel de prendre en compte le nombre d'interventions, le type de pathologies ainsi que les équipements et matériaux lors de la conception ou lors de la rénovation du bloc opératoire. A l'environnement physique s'ajoutent aussi les comportements humains lorsqu'il est question de sécurité au bloc opératoire. Ainsi, l'une des pièces maîtresse préalable à toute intervention chirurgicale est caractérisée par la check-list qui a sensiblement réduit la morbi-mortalité postopératoire. Le mode d'emploi proposé par la haute autorité de santé stipule : *“La check-list HAS (Sécurité du patient au bloc opératoire) comporte les éléments indispensables et non modifiables à vérifier ensemble et en équipe au bloc opératoire afin de prendre des décisions sur la poursuite ou non d'une intervention chirurgicale. Cependant, elle peut donner lieu à tout développement souhaité par les professionnels, dans le cadre de leurs collèges professionnels/organismes agréés d'accréditation.”*. Cette check-list qui se présente sous la forme de recommandations a pour objet la vérification de différents éléments impactant la sécurité du patient. Celle-ci comporte 11 points de vigilances, déclinés en 3 phases : pré-induction anesthésique, pré-intervention chirurgicale et post-intervention chirurgicale. La Figure 2.1 décline ces différentes préconisations validées à l'oral par le personnel. Elles permettent de vérifier de manière systématique l'identité du patient, le type d'intervention à venir, les antécédents du patient, mais aussi les stratégies contre le risque infectieux. Tout écart entraîne une concertation entre les intervenants et éventuellement l'interruption de la procédure chirurgicale.

Malgré l'optimisation de la conception des blocs opératoires, l'amélioration des techniques médicales et des processus de contrôle particulièrement rigoureux, la survenue d'événements indésirables persiste. En effet, selon le rapport de la HAS [40], 820 événements indésirables graves associés aux soins (EIGS) par million d'habitants ont été recensés en 2018 et pourraient être évités dans la moitié des cas. Notre travail de thèse qui porte sur la prédictivité de ces

CHECK-LIST « SÉCURITÉ DU PATIENT AU BLOC OPÉRATOIRE »
Version 2018
« Vérifier ensemble pour décider »

Bloc : Salle :
Date d'intervention : Heure (début) :
Chirurgien « intervenant » :
Anesthésiste « intervenant » :
Coordonnateur(s) check-list :

Identification du patient
Étiquette du patient ou
Nom, prénom, date de naissance

AVANT INDUCTION ANESTHÉSIQUE Temps de pause avant anesthésie	AVANT INTERVENTION CHIRURGICALE Temps de pause avant incision (appelé aussi time-out)	APRÈS INTERVENTION Pause avant sortie de salle d'opération
<p>1 L'identité du patient est correcte <input type="checkbox"/> Oui <input type="checkbox"/> Non</p> <p>2 L'autorisation d'opérer est signée par les parents ou le représentant légal <input type="checkbox"/> Oui <input type="checkbox"/> Non</p> <p>3 L'intervention et le site opératoire sont confirmés : • idéalement par le patient et, dans tous les cas, par le dossier ou procédure spécifique <input type="checkbox"/> Oui <input type="checkbox"/> Non • la documentation clinique et para clinique nécessaire est disponible en salle <input type="checkbox"/> Oui <input type="checkbox"/> Non</p> <p>4 Le mode d'installation est connu de l'équipe en salle, cohérent avec le site / l'intervention et non dangereux pour le patient <input type="checkbox"/> Oui <input type="checkbox"/> Non</p> <p>5 La préparation cutanée de l'opéré est documentée dans la fiche de liaison service / bloc opératoire (ou autre procédure en œuvre dans l'établissement) <input type="checkbox"/> Oui <input type="checkbox"/> Non</p> <p>6 L'équipement / le matériel nécessaires pour l'intervention sont vérifiés et adaptés au poids et à la taille du patient • pour la partie chirurgicale <input type="checkbox"/> Oui <input type="checkbox"/> Non • pour la partie anesthésique <input type="checkbox"/> Oui <input type="checkbox"/> Non <i>Acte sans prise en charge anesthésique</i> <input type="checkbox"/> N/A</p> <p>7 Le patient présente-t-il un : • risque allergique <input type="checkbox"/> Non <input type="checkbox"/> Oui • risque d'inhalation, de difficulté d'intubation ou de ventilation au masque <input type="checkbox"/> Non <input type="checkbox"/> Oui • risque de saignement important <input type="checkbox"/> Non <input type="checkbox"/> Oui</p>	<p>1 Vérification « ultime » croisée au sein de l'équipe en présence des chirurgiens(s), anesthésiste(s), IADE-IBODE/IDE • identité patient confirmée <input type="checkbox"/> Oui <input type="checkbox"/> Non • intervention prévue confirmée <input type="checkbox"/> Oui <input type="checkbox"/> Non • site opératoire confirmé <input type="checkbox"/> Oui <input type="checkbox"/> Non • installation correcte confirmée <input type="checkbox"/> Oui <input type="checkbox"/> Non • documents nécessaires disponibles (notamment imagerie) <input type="checkbox"/> Oui <input type="checkbox"/> Non</p> <p>2 Partage des informations essentielles oralement au sein de l'équipe sur les éléments à risque / étapes critiques de l'intervention (time-out) • sur le plan chirurgical (temps opératoire difficile, points spécifiques de l'intervention, identification des matériels nécessaires, confirmation de leur opérationnalité, etc.) <input type="checkbox"/> Oui <input type="checkbox"/> Non • sur le plan anesthésique (Acte sans prise en charge anesthésique [risques potentiels liés au terrain (hypothermie, etc.) ou à des traitements éventuellement maintenus, etc.]) <input type="checkbox"/> Oui <input type="checkbox"/> Non</p> <p>3 L'antibioprophylaxie a été effectuée selon les recommandations et protocoles en vigueur dans l'établissement La préparation du champ opératoire est réalisée selon le protocole en vigueur dans l'établissement <input type="checkbox"/> Oui <input type="checkbox"/> Non</p>	<p>1 Confirmation orale par le personnel auprès de l'équipe : • de l'intervention enregistrée <input type="checkbox"/> Oui <input type="checkbox"/> Non • du compte final correct des compresses, aiguilles, instruments, etc. <input type="checkbox"/> Oui <input type="checkbox"/> Non • de l'étiquetage des prélèvements, pièces opératoires, etc. <input type="checkbox"/> N/A • si des événements indésirables ou porteurs de risques médicaux sont survenus : ont-ils fait l'objet d'un signalement / déclaration ? <input type="checkbox"/> Oui <input type="checkbox"/> Non <i>Si aucun événement indésirable n'est survenu pendant l'intervention cochez N/A</i></p> <p>2 Les prescriptions et la surveillance post-opératoires (y compris les seuils d'alerte spécifiques) sont faites conjointement par l'équipe chirurgicale et anesthésique et adaptées à l'âge, au poids et à la taille du patient <input type="checkbox"/> Oui <input type="checkbox"/> Non</p> <p style="text-align: center;">Décision concertée et motivée en cas de réponse dans une case marquée d'un *</p> <p style="text-align: center;">ATTENTION SI ENFANT !</p> <p> <input type="checkbox"/> Associer les parents à la vérification de l'identité, de l'intervention et du site opératoire. <input type="checkbox"/> Autorisation d'opérer signée. <input type="checkbox"/> Installation, matériel et prescription adaptés au poids, à l'âge et à la taille. <input type="checkbox"/> Prévention de l'hypothermie. <input type="checkbox"/> Seuils d'alerte en post-opératoire. </p> <p style="text-align: center;">SELON PROCÉDURE EN VIGUEUR DANS L'ÉTABLISSEMENT</p> <p>Attestation que la check-list a été renseignée suite à un partage des informations entre les membres de l'équipe</p> <p style="text-align: center;">Chirurgien Anesthésiste / IADE Coordonnateur CL</p>
<p>→ DÉCISION FINALE</p> <p><input type="checkbox"/> GO = OK pour incision <input type="checkbox"/> NO GO = Pas d'incision !</p> <p>Si No Go : conséquence sur l'intervention ? <input type="checkbox"/> Retard <input type="checkbox"/> Annulation</p>		

le rôle du coordonnateur de la check-list, aidé par le(s) chirurgien(s) et anesthésiste(s) responsables de l'intervention, est de cocher les items de la check-list : 1. si la vérification a bien été effectuée ; 2. si la vérification a été faite oralement en présence des membres de l'équipe concernée et 3. si les réponses marquées d'un * sont fait l'objet d'une concertation en équipe et d'une décision motivée.

N/A : Non Applicable pour cette intervention ; NR : Non Recommandé pour cette intervention

HAUTE AUTORITÉ DE SANTÉ

FIGURE 2.1 – Check-list bloc opératoire [HAS 2018]

événements nécessite au préalable une identification de la cause de leur survenue que l'on peut classer en cinq facteurs : facteurs de risque liés au patient, facteurs de risque liés au personnel, facteurs de risque liés à l'environnement, facteurs de risque liés au processus opératoire, facteurs de risque liés au post-opératoire.

Les Sections 2.2, 2.3 et 2.4 qui suivent décrivent succinctement les facteurs de risque liés au patient, au personnel et à l'environnement. La Section 2.5 explicite plus en détail les facteurs de risque liés au processus opératoire et en particulier à l'intervention chirurgicale qui correspond au système que nous modélisons. Les facteurs de risque liés au post-opératoire sont en dehors de notre périmètre d'étude et ne sont donc pas traités dans ce manuscrit.

2.2 Facteurs de risques liés au patient

Les facteurs de risques en relation avec le patient, tels que les infections cutanées, urinaire ou ORL ont pour cause un défaut de traitement de celles-ci en amont de l'intervention. Il en est de même pour le diabète ou l'obésité. La flore endogène quant à elle résulte d'une hygiène générale (douche au savon septique) imparfaite. La non détection de SARM (staphylocoque doré) conduit aussi à l'accentuation de la flore endogène. D'autres facteurs tels que le tabagisme, l'asthme, l'âge, la prématurité, les allergies, le degré d'urgence de l'intervention sont aussi des

causes possibles d'EIAS propres au patient.

2.3 Facteurs liés au personnel

Les risques liés au personnel, se déclinent en deux types : le risque infectieux et le risque comportemental. Dans le premier cas (microorganismes répartis dans l'air), une tenue vestimentaire inadaptée ou non stérile peut être la cause d'EIAS tout comme le non-respect scrupuleux des protocoles de désinfection exigés pour l'hygiène des mains. Dans le second cas, les événements indésirables peuvent être la conséquence d'un nombre trop important ou insuffisant de personnes. Cela induit respectivement du stress lié à l'agitation des va et vient ou de la nervosité face à une surcharge à répartir sur un nombre insuffisant de personnes. Cette situation anxieuse s'accroît sur des personnels inexpérimentés. Outre ces causes, figurent la fatigue humaine et la baisse de vigilance non négligeables dans des contextes d'interventions longues et complexes.

2.4 Facteurs liés à l'environnement

L'agencement de l'environnement dans lequel évoluent les praticiens et dont dépendent les équipements, l'air, l'eau et les surfaces est déterminant en termes de réduction des risques pour les patients. L'un des principaux objectifs visés étant la minimalisation des risques de contamination, une attention particulière est apportée au contrôle des personnes et matériels se déplaçant entre des zones d'asepsie différentes. Des circuits ont été conçus dans ce sens. Parmi les nombreuses possibilités qui existent, l'un des principes fondamentaux est la "marche en avant" [53] qui consiste à aller du plus sale vers le plus propre. C'est une stratégie essentielle pour lutter contre les infections au bloc opératoire. Cinq zones d'asepsie différentes et croissantes sont définies allant de l'extérieur du bloc opératoire jusqu'à la table d'opération. L'asepsie croissante est le résultat d'une surpression de 15 pascals entre les différentes zones. La Figure 2.2 nous montre cette progression délimitée par des sas (douanes). Celles-ci sont destinées à réduire l'introduction de micro-organismes au fil du déplacement de la zone 5 (la moins stérile) vers la zone 1 (la plus stérile).

Un défaut d'architecture, ou une salle inadaptée à la spécificité d'une chirurgie sont des sources d'incidents directs (erreur dans la pratique lors de l'intervention) ou indirects (stress généré par un cadre non fonctionnel). Concernant l'aspect physique de l'environnement, une déficience dans le traitement de l'air (surpression insuffisante) ou bien encore la présence de germes dans l'eau sont des sources d'EIAS. Citons enfin les matériels dont la désinfection (ou stérilisation) imparfaite conduisent à des contaminations.

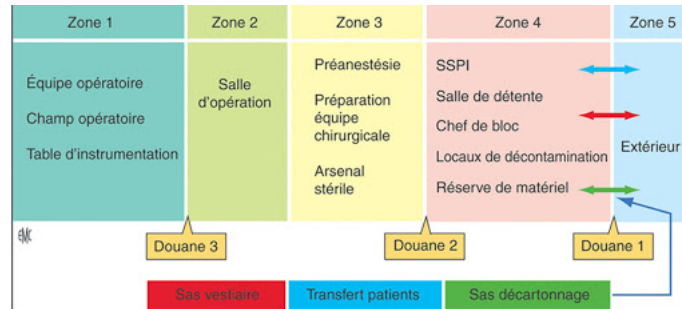


FIGURE 2.2 – Circuit du bloc opératoire [15]

2.5 Facteurs liés au processus opératoire

Le processus opératoire est le cheminement du patient allant de la prise de rendez-vous pour la consultation chirurgicale jusqu'à la sortie du bloc opératoire. Il comprend : les consultations anesthésiques et chirurgicales, l'anesthésie, la préparation chirurgicale, l'intervention chirurgicale, le réveil et/ou la réanimation. Dans le cadre de cette thèse, nous porterons uniquement notre attention sur les étapes : de l'anesthésie, de la préparation du patient et de l'intervention chirurgicale. En effet, notre périmètre d'étude porte uniquement sur ces processus sachant que toutes les étapes du parcours patient peuvent impacter les EIAS.

Les trois paragraphes qui suivent exposent l'identification des risques d'EIAS possibles lors des phases d'anesthésie du patient, de préparation du patient et d'intervention chirurgicale. Les Figures 2.3, 2.4, 2.5 ont été réalisées en collaboration avec les chirurgiens du service de chirurgie pédiatrique du centre hospitalier régional universitaire de Besançon.

Défaillances lors de l'anesthésie du patient

L'anesthésie consiste à suspendre momentanément la sensibilité et en particulier la sensation de douleur. Elle peut être générale (perte de conscience) ou loco-régionale (sans perte de conscience). Garantir la sécurité de cette pratique de façon optimale impose un protocole strict et rigoureux décrit par la Figure 2.3. Les infirmiers anesthésistes diplômés d'état (IADE) préparent dans le bloc opératoire les différents matériels prévus pour l'ensemble des interventions programmées dans la journée. Le patient est transporté par les brancardiers dans l'ordre du planning ou des urgences traumatiques. Il est ensuite installé sur la table d'opération par l'anesthésiste et les aides-soignants. L'IADE et l'anesthésiste installent le moniteur et posent les voies veineuses périphériques. L'induction peut dès lors être réalisée.

Les principales défaillances observables lors des différentes phases anesthésiques peuvent se résumer ainsi :

- *Préparation du matériel* : matériel non conforme, matériel non stérilisé, matériel mal affecté.
- *Transport du patient* : non-respect des règles d'hygiène et de positionnement du patient, patient mal affecté.

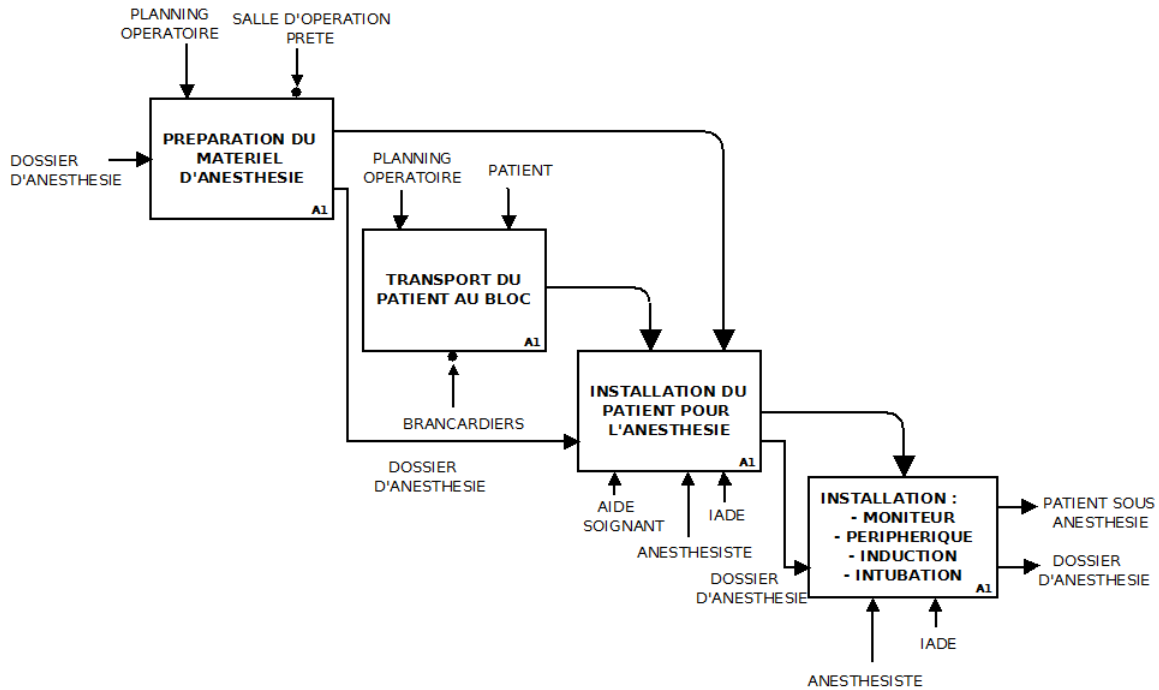


FIGURE 2.3 – Anesthésie du patient

- *Installation du patient* : non-respect des règles d’hygiène et de positionnement du patient, données du patient non conformes.
- *Installation du moniteur et des voies veineuses périphériques* : erreurs de produit, matériel altéré, panne de gaz médicaux, mauvaise manipulation.

Si l’efficience des machines entre dans le panel des causes de manière marginale grâce aux techniques préventives de plus en plus sophistiquées (PHM, auto-diagnostic ...), il n’en reste pas moins un élément à ne pas négliger. En effet, l’impact morbide est parfois critique en raison de la dépendance du patient par rapport à ces dernières. Au-delà de cet aspect, la sophistication accrue des appareils de monitoring exige pour leur utilisation optimale et sans danger pour le patient, une expertise croissante. Cela nécessite donc des temps de formation qu’il est parfois difficile de concilier avec des emplois du temps tendus.

Sur le plan physiologique, on recense, parmi les complications les plus graves, les chocs anaphylactiques (réactions allergiques exacerbées), les détresses respiratoires et les arrêts cardiaques. Une mauvaise appréciation des antécédents du patient et/ou une erreur de diagnostic pré-anesthésique sont déterminantes dans la survenue de ces incidents. Outre ces cas graves, la plupart des complications sont réversibles. Citons par exemple les extubations (retrait du dispositif tubulaire dans la trachée) accidentelles que l’on peut attribuer à une position inadéquate du patient ou à une maîtrise insuffisante des gestes techniques. Il est difficile de dissocier l’anesthésie de l’acte chirurgical et de ses conséquences à plus ou moins long terme sur l’état pathologique du patient. C’est pourquoi la modélisation que nous proposons dans l’étude inclue

de facto l'anesthésie comme élément de notre système à la fois collaboratif et autonome.

Défaillances lors de la préparation du patient

Les processus propres à la préparation du patient sont exprimés dans la Figure 2.4. En collaboration avec le chirurgien, les infirmiers de bloc opératoire diplômés d'état (IBODE) préparent les matériels nécessaires à l'intervention selon les préconisations du dossier médical du patient. Le patient peut être ensuite repositionné pour optimiser l'accès aux zones à opérer. L'IBODE aseptise soigneusement la peau et procède avec le chirurgien au drappage chirurgical (préparation du champ opératoire). En dernier lieu, le chirurgien énumère chaque point d'une check-liste qui doit être approuvée par tous les membres de l'équipe.

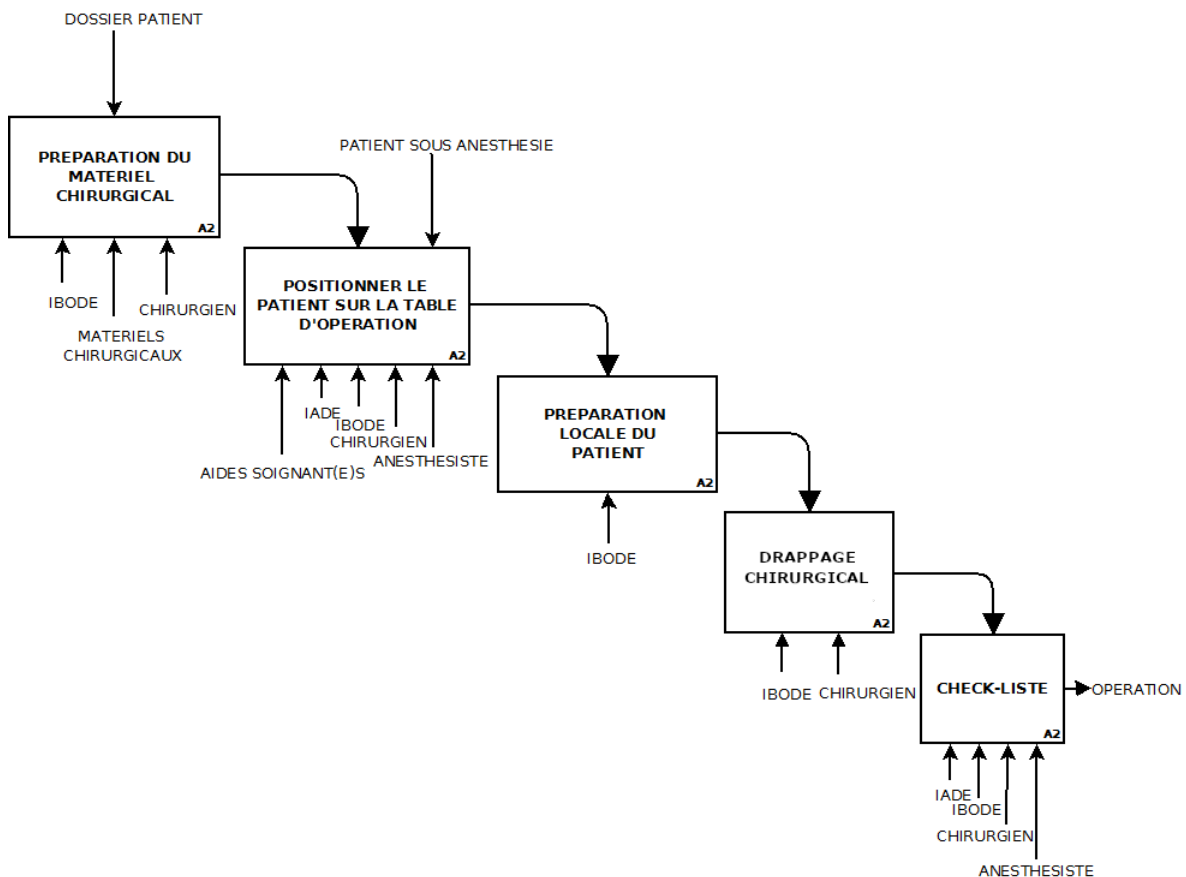


FIGURE 2.4 – Préparation du patient

Les défaillances potentielles lors des quatre processus de préparation du patient se répartissent comme suit :

- *Préparation du matériel de chirurgie* : matériel non opérationnel, système d'information en panne, éclairage défaillant.
- *Positionnement du patient* : table d'opération non adaptée, accessoires manquants, mauvais positionnement du patient, aérocontamination.

- *Préparation locale du patient* : zone opératoire trop contaminée, aérocontamination, erreur dans l'utilisation des produits.
- *Drappage chirurgical* : matériel absent ou inadapté, aérocontamination.

Les causes principales de ces défaillances proviennent souvent : d'un défaut de maintenance du matériel, des erreurs dans la programmation de l'intervention, des recommandations mal appliquées et enfin d'une mauvaise gestion des stocks (stock insuffisant).

Défaillances lors de l'intervention chirurgicale

Dans le cadre académique, l'intervention est réalisée par un chirurgien expert aidé d'un assistant "junior" (en cours d'apprentissage). Ils sont assistés par les IBODE qui passent les instruments de chirurgie et gèrent les transferts de matériels en garantissant leur stérilité. Dans un même temps, les anesthésistes et l'IADE gèrent l'anesthésie et contrôlent les constantes du patient captées et transmises au monitoring. Après l'intervention, l'anesthésiste et l'IBODE extubent le patient qui est ensuite transféré dans son lit par les aides soignants pour être acheminé en salle de réveil. Cet enchaînement est décrit dans la Figure 2.5.

La liste proposée ci-dessous (non exhaustive) énumère les défaillances observables lors du déroulement des quatre processus liés à l'intervention chirurgicale :

- *Intervention sur le patient* : panne matérielle, luminosité déficiente, patient infecté, infection aéroportée (particules présentes dans l'environnement), durée de l'intervention, fatigue des intervenants.
- *Extubation du patient* : lésions mécaniques (laryngotrachéales ...), extubation accidentelle.
- *Transfert du patient dans son lit* : problèmes ostéoarticulaires.
- *Transfert du patient en salle de réveil* : panne matérielle ou matériel manquant, couloir encombré.

Les principales causes inhérentes à ces défaillances impliquent souvent les matériels et consommables, le physique (air, eau), l'organisation du service et enfin les facteurs humains. Ainsi, une panne de matériel, un stock insuffisant de consommables ou un mauvais traitement de l'air peut conduire à un événement indésirable. Il en va de même lorsque le service est désorganisé ou lorsque l'intervention est trop longue. Citons enfin les mauvais gestes techniques, les extubations accidentelles ou bien encore les manipulations inadaptées souvent liées à une fatigue accrue des personnels.

2.6 Bilan du chapitre

Ce chapitre, où nous venons de présenter notre contexte d'étude, a pour objectif la mise en exergue de la sécurité du patient qui est au cœur de toutes les activités du bloc opératoire. Nous

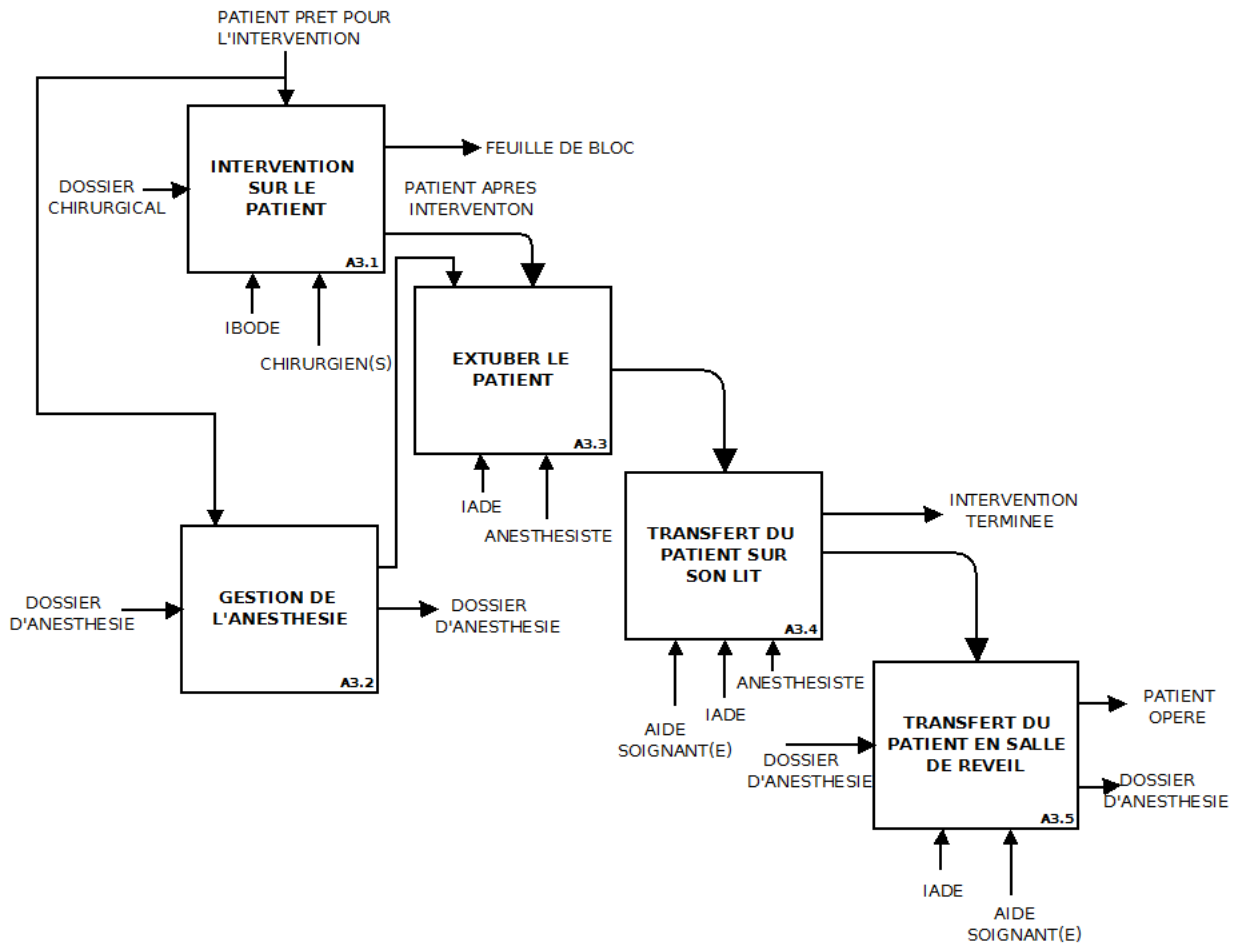


FIGURE 2.5 – Intervention chirurgicale

avons décrit et mis en évidence le caractère complexe de l’organisation d’un bloc opératoire qui ne se résume pas à une intervention chirurgicale. Il s’agit en réalité d’une succession de processus : consultation chirurgicale et d’anesthésie, anesthésie, préparation du patient, intervention chirurgicale. De nombreux protocoles de plus en plus élaborés de mise en sécurité du patient existent mais restent insuffisants au regard des défaillances et de leur causes (cf Section 2.5). C’est pourquoi, les professionnels de santé ont de plus en plus recours à différents paradigmes de l’IA afin d’analyser des situations à risques ou d’en anticiper de nouvelles au travers de modélisations [87].

Le travail d’identification d’EIAS et les causes de leurs survenues a été réalisé en concertation avec les experts impliqués dans le projet MASSAI et constitue les éléments de base utilisés pour la construction de notre modèle. Ce dernier est caractérisé par la capacité à faire émerger des situations à risques non prévisibles a priori, mais aussi par la capacité à anticiper la survenue d’EIAS.

Le chapitre suivant, décrit notre contexte scientifique puis les outils, les méthodes et les différents paradigmes qui, à notre connaissance, sont les plus usuellement utilisés pour prédire

l'évolution d'un système.

Chapitre 3

Contexte scientifique et état de l'art

Sommaire

3.1	Introduction	25
3.2	Prognostic and Health Managment	25
3.2.1	Architecture du PHM	26
3.2.2	Approche du PHM au bloc	27
3.2.3	Bilan de l'approche PHM	28
3.3	La modélisation SMA	28
3.3.1	Fondamentaux de la modélisation	29
3.3.2	La notion d'agent	30
3.3.3	Le système multi-agents	32
3.3.4	Analyse et bilan d'une modélisation SMA	37
3.4	Les systèmes à base de connaissances	38
3.4.1	Les réseaux de neurones artificiels	38
3.4.2	Le raisonnement à partir de cas	41
3.4.3	Couplage avec RàPC : quelques exemples significatifs	47
3.5	Les méthodes de clusterisation	48
3.5.1	Méthode basée sur la <i>Partition</i>	49
3.5.2	Méthode basée sur la <i>Hiérarchie</i>	50
3.5.3	Méthode basée sur la <i>Théorie floue</i>	51
3.5.4	Méthode basée sur la <i>Distribution</i>	51
3.5.5	Méthode basée sur les <i>Noyaux</i>	52
3.5.6	Méthode basée sur la théorie des <i>Graphes spectraux</i>	52
3.5.7	Méthode basée sur l' <i>Affinité de propagation</i>	52
3.5.8	Bilan des méthodes de clusterisation	53
3.6	Bilan du chapitre	53

3.1 Introduction

Le chapitre précédent montre que le bloc opératoire est un système complexe où sont pratiqués des actes médicaux susceptibles de produire des effets indésirables dont la nature et la gravité sont variables. L'une des préoccupations majeures consiste à prévenir ces risques non pas étape par étape mais de manière globale. Cette approche est d'autant plus difficile à mettre en œuvre qu'elle s'ajoute à d'autres éléments conjoncturels tels que :

- la diminution de l'acceptabilité des risques par les patients,
- une réglementation complexifiée,
- la maîtrise des coûts,
- la pénurie médicale,
- l'évolution technologique,
- des interventions de plus en plus sophistiquées donc plus risquées.

Face à cet enjeu, différentes pistes de réflexions ont guidé nos recherches. Parmi nos premières investigations figurent la détermination de l'état de santé d'un matériel et son efficience. En effet, toute machine se dégrade avec le temps jusqu'à la défaillance qui peut se produire à des moments inopportuns et amener des conséquences particulièrement négatives, voire critiques lorsque des vies humaines en dépendent. Dans une vision plus globale, la modélisation du bloc opératoire considéré comme un système où évoluent des entités qui interagissent entre elles est aussi une piste de recherche qui a retenue notre attention. A celle-ci s'ajoute la capitalisation de connaissances nécessaire à l'enrichissement de la mémoire du système et à la résolution de problèmes par analogie lorsque par exemple on cherche à déterminer des EIAS. Au-delà de la modélisation, l'analyse statistique des données possédant des caractéristiques communes a également été le sujet de nos travaux. Nous proposons dans les sections qui suivent de présenter les principaux paradigmes en lien avec les pistes de recherche que nous venons d'évoquer. Chacun des paradigmes est abordé selon ses caractéristiques, son formalisme et ses fonctionnalités. Les travaux connexes qui s'y rapportent sont aussi exposés.

3.2 Prognostic and Health Management

Les exigences de performance intrinsèques au monde industriel ont permis de développer de nouveaux outils renforçant l'anticipation des phénomènes de défaillance. Parmi les solutions, le *Prognostic and Health Management* (PHM) [36] prend une place prépondérante dans les activités de maintenance. Le pronostic affine les actions préventives, renforce l'anticipation des défaillances et augmente la performance globale des systèmes industriels, que ce soient sur leur cycle de vie, ou sur la qualité de leur exploitation. A ces processus s'ajoutent les mécanismes d'aide à la décision pour maintenir la disponibilité d'un système, tout en assurant un compromis entre la sécurité et les coûts d'exploitation. Certaines actions peuvent être entreprises comme

la planification ou la réalisation d’actions de maintenance suivant les impératifs d’exploitation associés au système. Le champ d’application du PHM est certes déterminant dans la maintenance industrielle [71], mais se positionne aussi dans des activités telle que l’e-maintenance [64]. L’architecture du PHM et l’approche de ce paradigme par rapport au bloc opératoire sont l’objet des sections 3.2.1, 3.2.2.

3.2.1 Architecture du PHM

Le PHM est un paradigme qui permet d’évaluer la fiabilité d’un système dans les conditions réelles de son cycle de vie afin de prévenir l’apparition de pannes [86]. Cela est matérialisé par l’observation et l’analyse des données émises par le système pour : détecter les anomalies, diagnostiquer les pannes et estimer la vie restante utile (RUL : *Remaining Useful Life*) des composants du système en temps réel. Cette méthode se décompose en six étapes énumérées dans la Figure 3.1. Chacune d’elle est présentée dans les paragraphes qui suivent.

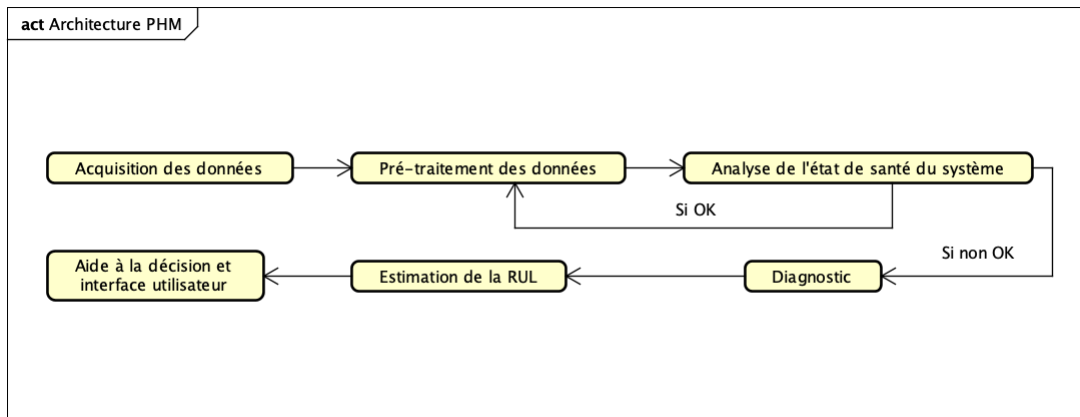


FIGURE 3.1 – Architecture PHM [80]

Acquisition des données

Les différents signaux retournés par le système sont observés en continu par des capteurs qui doivent garantir un niveau de fiabilité suffisant. Cela impose un contrôle de leur efficacité comme le montrent J. Xu *et al.* [93] qui proposent une évaluation des capteurs en parallèle de l’architecture. La pertinence du choix de ces sources de données est aussi déterminant dans un système complexe où les composants sont inter-connectés.

Pré-traitement des données

L’étape qui suit l’acquisition des données est destinée à réduire voire éliminer les bruits, les valeurs aberrantes et les vides, afin d’éviter les distorsions engendrées sur les mesures d’évaluation du système. Cette étape est d’autant plus importante qu’elle impacte aussi l’apprentissage propre à la détection d’anomalies.

Analyse de l'état de santé du système

Cette phase permet de détecter les anomalies du système grâce aux données collectées par les capteurs et pré-traitées. Trois méthodes de détection existent. La première est basée sur les seuils (déterminés pour chaque capteur) qui révèle une anomalie lorsqu'ils sont dépassés. La seconde utilise des modèles théoriques qui représentent le système dont les dégradations sont estimées à partir d'outils statistiques [70]. Enfin, la troisième méthode utilise le Machine Learning (ML) en s'appuyant sur des jeux de données d'apprentissage suffisants. Cette dernière méthode ne nécessite pas de connaissances spécifiques du système et permet une adaptation facile aux nouvelles pannes intégrées lors des apprentissages [94].

Diagnostic de la panne

Diagnostiquer une panne revient à la déterminer en fonction des anomalies décelées. Parmi les principales technologies utilisées dans cet objectif, se trouvent selon [55] : les réseaux bayésiens, le Machine Learning, le raisonnement à partir de cas et les réseaux de neurones artificiels. L'enjeu est dicté par la mise en relation des mesures collectées avec un type de panne diagnostiqué. A contrario des réseaux bayésiens qui nécessitent une connaissance fine du domaine, les réseaux de neurones artificiels ou le machine learning peuvent entraîner l'algorithme par apprentissage (à partir d'un jeu de données), sans connaissance préalable du domaine.

L'état de l'art relatif au PHM couvre essentiellement les environnements industriels. Qu'en est-il de cette méthode d'évaluation des défaillances dans le contexte de notre étude centrée sur la prédiction des risques au bloc opératoire ? Nous pouvons en effet considérer que le risque de défaillance matériel est corrélé à la sécurité du patient en cours d'intervention. D'ailleurs, l'expertise dans la pratique chirurgicale dépend en grande partie de l'efficacité des différents matériels (robots, appareils radiographiques, scanner, matériels de chirurgie . . .) qui sont utilisés durant le parcours du patient en chirurgie. L'approche du PHM au bloc est l'objet de la section suivante.

3.2.2 Approche du PHM au bloc

Selon la Direction Régionale des Affaires Sanitaires et sociales (DRASS), la "*maintenance des dispositifs médicaux (DM) est un sujet d'actualité, d'autant plus que l'arrêté du 3 mars 2003 rend obligatoire la maintenance de certaines catégories de dispositifs médicaux à partir du 1er janvier 2005*". La maintenance consiste donc à maintenir opérationnel un dispositif médical et entre dans le cadre du PHM, en particulier dans la stratégie prédictive des défaillances. Pour illustrer ce propos, considérons par exemple un moniteur de surveillance pour l'anesthésie illustré par la Figure 3.2. Ce type d'appareil, décèle les problèmes liés aux constantes du patient et garantit un dosage approprié d'anesthésiant. Une maintenance constructeur garantit des interventions préventives systématiques planifiées selon un calendrier annuel. A cela s'ajoute une position dynamique de la prévention par le biais de captures en temps réel d'indicateurs physiques (pression, fréquence . . .) de variations statistiques par rapport au mode opérationnel.

Dans le contexte de notre étude dont la finalité consiste à centraliser et traiter informatiquement toutes les informations induisant un risque, ces captures simultanées des défauts semblent réalisables voire indispensables.

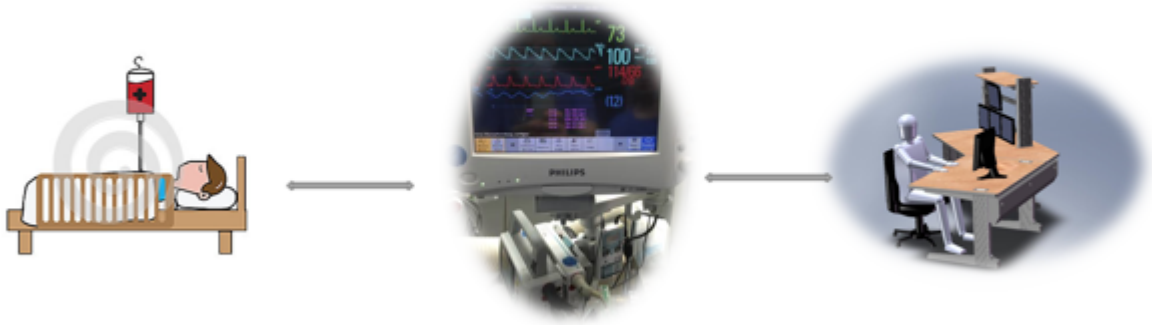


FIGURE 3.2 – Appareil d’anesthésie

3.2.3 Bilan de l’approche PHM

Le paradigme PHM montre un intérêt majeur dans l’industrie où l’anticipation des défaillances dans les systèmes est vitale. Le diagnostic et le pronostic de l’état d’un système en sont les clés de voute. Une transposition est envisageable dans l’environnement matériel du bloc opératoire (appareil de monitoring, bistouri électrique ...). Cependant, il n’existe pas à notre connaissance d’études sur la prédiction des risques des systèmes interactifs complexes fonctionnant dans des contextes non-déterministes tels que la chirurgie et l’environnement de la salle d’opération.

Nous pensons que les systèmes multi-agents sont adaptés à ce problème puisqu’ils sont construits selon le concept d’interactions. Dans la suite de ce chapitre, nous abordons ces techniques d’intelligence artificielle (IA) destinées à l’élaboration de modèles.

3.3 La modélisation SMA

La modélisation SMA intègre comme toute modélisation des principes fondamentaux qu’il convient de prendre en compte afin de structurer le modèle avec le plus de précision possible. Ce thème est donc abordé en premier lieu dans la suite de cette section suivi d’une description spécifique de la modélisation SMA.

3.3.1 Fondamentaux de la modélisation

La modélisation est, par définition, la représentation d'un système par un autre plus facile à appréhender. Plus formellement, la modélisation se caractérise par deux points essentiels :

- L'abstraction, qui permet de ne pas prendre en compte les détails inutiles pour mieux raisonner ; ces derniers sont introduits au fur et à mesure des besoins lors de l'élaboration ou de la conception du modèle.
- La classification, qui regroupe les informations basées sur des propriétés communes.

En outre, il est souvent plus facile de communiquer à partir de représentations (schéma, simulateur ...) qu'à partir de discours ou de textes. De même la simulation est reproductible à l'infini contrairement aux phénomènes réels et présente un intérêt économique majeur (crash test de voiture simulé et crash test réel). Ces différents principes répondent à notre principal objectif qui consiste à comprendre voire maîtriser l'évolution d'un système complexe et justifie notre choix pour la modélisation. Différentes approches de modélisation existent et sont évoqués dans les trois paragraphes qui suivent.

Approche ascendante ou descendante

Les approches ascendantes (bottom-up) et descendantes (top-down) sont deux approches de modélisation fondamentales. En effet elles sont déterminantes par rapport aux résultats attendus comme le montre [89]. Lorsque l'approche est descendante, la conception du modèle débute par la spécification de l'état global du système qui est ensuite affecté à chacun de ses composants. Ils possèdent donc une connaissance globale du système. Le développement d'application en informatique est un exemple d'approche descendante car il requiert une maîtrise préalable de ses concepts.

Dans une approche ascendante, les spécifications (capacités, comportements, exigences attendues) pour chacun des composants constituent un préalable à la construction du modèle. Le comportement global émerge quant à lui des interactions entre les différents composants. Cette approche est largement utilisée en physiologie humaine. A titre d'exemple, le fonctionnement du foie est expliqué grâce aux fonctionnements constitutifs de cet organe, tels que la biosynthèse de cholestérol et la formation de bile. Ces approches qui présentent des principes opposés peuvent cependant être complémentaires.

Continuité dans les simulations

La simulation d'un système est dépendante des notions de discrétisation ou de continuité. Lorsqu'une simulation est continue, les variables changent continuellement d'état en fonction du temps et non en fonction d'événements. Ces simulations sont largement représentées par des fonctions mathématiques telles que les équations différentielles. Les simulations de phénomènes physiques qui nécessitent une observation continue de l'évolution de l'état du système utilisent souvent la simulation continue.

La principale différence entre la simulation continue et la simulation à temps discret est liée

à la notion d'événements. En effet, la simulation à temps discret est associée à des systèmes où les transformations résultent d'événements ponctuels, souvent matérialisés par des signaux ou par l'achèvement de tâches. Ces simulations sont caractéristiques des automates à états finis dont les états sont dénombrables. Les systèmes multi-agents qui intègrent les pas de temps permettent la simulation en temps discret [84].

Modélisation distribuée

La résolution distribuée de problèmes est un paradigme inventé pour palier les limites de l'IA lorsqu'elle ne concentrait son "intelligence" que sur un système unique. Cela s'est traduit par la distribution de capacités à des agents capables d'autonomie, de coopération et d'actions indépendamment d'un système centralisé [26]. La vision de l'approche distribuée dans la résolution de problèmes est donc très étroitement liée à la notion d'agents et de système multi-agents qui sont l'objet des deux sections qui suivent.

3.3.2 La notion d'agent

Un système multi-agents se singularise des autres outils de modélisation en intégrant dans sa dialectique le groupe et donc les notions d'interaction, de coopération, de conflit, de collaboration et de coordination qui en découlent. Il est composé d'entités artificielles appelées agents que nous allons définir et décrire dans les quatre paragraphes suivants.

Définition

Il est difficile de trouver des définitions qui fassent l'unanimité auprès de la communauté scientifique, cependant, celles qui suivent sont relativement consensuelles :

- Un agent est une entité qui perçoit son environnement et agit sur celui-ci [58].
- C'est un système informatique situé dans un environnement et qui agit d'une façon autonome pour atteindre les objectifs pour lesquels il a été conçu [46].
- "Un agent est un objet informatique (au sens des langages objets) dont le comportement peut être décrit par un « script », qui dispose de ses propres moyens de calcul (un agent est alors associé à un processus léger), et qui peut se déplacer de place en place (une place pouvant être un site informatique distant du site originel de l'agent) pour communiquer avec d'autres agents." [27].

Jacques Ferber [26] nous donne quant à lui une définition qu'il considère "minimale" et "commune" et qui reprend toutes les fonctions majeures d'un agent : "Un agent est une entité physique ou virtuelle :

- qui est capable d'agir dans un environnement,
- qui peut communiquer directement avec d'autres agents,
- qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels) ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser,
- qui possède des ressources propres,

- qui est capable de percevoir, mais de manière limitée, son environnement,
- qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),
- qui possède des compétences et des offres de services,
- qui peut éventuellement se reproduire,
- dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit."

De cette définition découle des propriétés qui confèrent aux agents les capacités résumées dans le Tableau 3.1.

Propriété	Capacité
Réactivité	Perception de l'environnement pour répondre en temps réel aux changements
Proactivité	Prise d'initiatives
Sociabilité	Interaction avec d'autres agents
Autonomie	Prise de décisions

Tableau 3.1 – Propriétés et capacités des agents

Le paragraphe suivant décrit le classement des agents en fonction de leur typologie.

Les types d'agents

Le classement des agents est intimement lié à leur capacité d'apprentissage, de raisonnement, de compréhension et de planification. Il dépend aussi de leur autonomie et flexibilité dans un environnement dynamique. Le classement ci-dessous hiérarchise les groupements selon une granularité cognitive croissante :

- Les agents réactifs : ce sont des agents passifs qui réagissent seulement à un stimulus. Ce type d'agent ne dispose pas de module de raisonnement interne. Par exemple, le robot nettoyeur change de direction s'il rencontre un obstacle (situation, action).
- Les agents proactifs : ce sont des agents dynamiques qui entreprennent, car ils possèdent en plus de leurs attributs et méthodes, des processus internes qui leur permettent de prendre des initiatives pour réaliser leurs buts. Un agent proactif est donc un agent "dirigé buts". Par exemple, le robot nettoyeur aspire la poussière, sinon il avance (actions en fonction de conditions).
- Les agents cognitifs tels que les agents BDI (Desires, Believes, Intentions) : un agent cognitif raisonne avant d'agir, il est souvent associé au trio bouclant : perception-raisonnement-action. Il possède, en plus de ses buts, des notions psychologiques qui peuvent être exprimées par le biais d'attitudes mentales comme les croyances (Belives), les intentions (Intentions) et les désirs (Desires).
- Les agents mutli-niveaux : il existe différents niveaux d'abstraction permettant différents niveaux de traitement.

Après avoir décrit ce qu'est un agent, reste à expliquer son fonctionnement.

Le fonctionnement d'un agent

Un agent est une entité autonome qui peut être assimilée à plusieurs fonctions. Il peut réaliser une action après réception d'un message, mais ne peut agir à la place d'un autre agent. Il possède un certain nombre de compétences telles que la prise de décisions, la capacité à planifier et à penser. Ces compétences varient selon l'architecture interne propre à deux familles d'agents définis par [32] : les agents "tropiques" et les agents "hystérétiques". Les agents tropiques (agents réactifs) ne possèdent pas d'état interne et agissent par réflexe selon leurs perceptions. Leurs décisions ne dépendent pas d'un historique. D'un point de vue formel, si φ représente l'ensemble des états du système, λ_a l'ensemble des perceptions d'un agent a et β_a l'ensemble des actions réalisables par l'agent a , alors le processus de décision d'un agent a résulte de deux fonctions exécutées consécutivement :

- $P_a : \varphi \mapsto \lambda_a$: une perception est associée à l'état du système.
- $R_a : \lambda_a \mapsto \beta_a$: une action est associée à chaque perception.

Les agents "hystérétiques" (agents proactifs, agents cognitifs), possèdent quant à eux un état interne. Cette distinction par rapport aux agents "tropiques" leur permet de mémoriser des informations et de prendre des décisions. Sur un plan formel, si E_a représente un état interne de l'agent a , alors le processus de décision d'un agent a résulte de trois fonctions exécutées consécutivement :

- $P_a : \varphi \mapsto \lambda_a$: une perception est associée à l'état du système.
- $M_a : \lambda_a \times E_a \mapsto E_a$: un nouvel état interne E_a est calculé et mémorisé en fonction de l'état interne et de la perception de l'agent a .
- $D_a : \lambda_a \times E_a \mapsto \beta_a$: une action est associée au nouvel état interne de l'agent a et à sa perception courante.

La différence majeure entre ces deux types d'agents, par rapport à leur processus décisionnel, se résume donc par leur capacité ou non à mémoriser ce qu'ils perçoivent.

Les agents sont conceptuellement des entités qui appartiennent à un environnement organisé de manière à permettre des échanges collaboratifs. Ce type de modélisation nommé communément SMA est détaillé dans le paragraphe qui suit.

3.3.3 Le système multi-agents

Selon Ferber [26] un système multi-agents est un système distribué, constitué d'un ensemble d'agents tels que :

- chaque agent possède des informations et des capacités à résoudre des problèmes limités ;
- le contrôle global du SMA soit impossible ;
- les données soient décentralisées ;

- les calculs soient asynchrones (chaque agent peut effectuer des calculs indépendamment des autres).

L'interaction se caractérise par la mise en relation dynamique de deux ou plusieurs agents qui réalisent un ensemble d'actions réciproques. Elle peut être directe, indirecte, à travers l'environnement, coopérative ou compétitive. La Figure 3.3 illustre les interactions entre agents, leur perceptions et leur actions ; chacun d'eux possédant des objectifs et des connaissances.

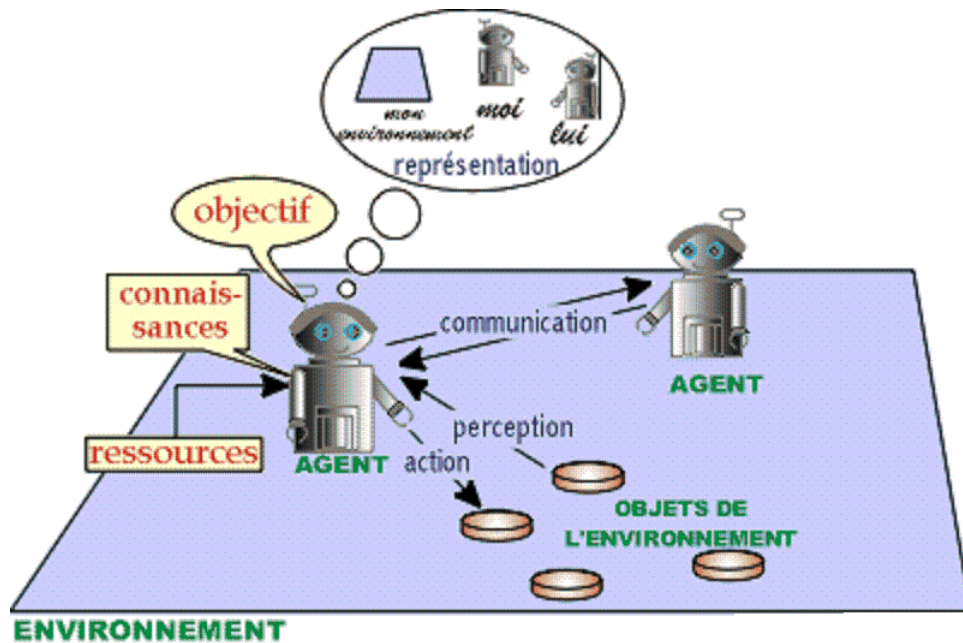


FIGURE 3.3 – SMA selon Ferber [26]

Les propriétés et rôles d'un système multi-agents

Dans un contexte SMA, la connaissance et le savoir-faire sont attribués aux différents agents. Chacun d'eux possède sa propre représentation de l'environnement et ses propres compétences, et c'est leur mutualisation qui permet d'atteindre les objectifs d'optimisation comme par exemple la réduction d'un risque [21]. Par cette propriété un système multi-agents est autonome, et n'est pas contrôlé par un système extérieur. Chaque agent est responsable de ses propres compétences et exécutions qui sont souvent parallèles entre elles [26]. Un système SMA est ouvert lorsque les agents peuvent apparaître ou disparaître, sinon il est fermé (pas de communication avec l'extérieur). Cette caractéristique couplée à l'autonomie confère à certains agents la capacité d'en créer d'autres et éventuellement de "périr" par suicide. D'un point de vue global, un système est défini comme homogène lorsque tous les agents possèdent le même type de perception (vue des praticiens dans notre contexte ...) et d'actions (contamination par des agents pathogènes dans notre contexte ...) et ne diffèrent que par des caractéristiques propres (forme, taille dans notre contexte ...) [24]. Dans le cas contraire, le système est hétérogène. Aux propriétés que nous venons de donner, s'ajoutent les rôles que l'on peut classer en trois branches :

- La résolution de problèmes distribués : les agents “rationnels” transforment les objets liés à la description d’un problème (exemple : agent contaminé → agent non contaminé).
- Simulation de problèmes complexes : les agents “réactifs” simulent des actions physiques, biologiques ou sociales qui produisent des modifications du monde (écosystème : proies/prédateurs ...).
- Gestion et maintien d’un environnement de travail : agents “cognitifs” dont les actions évoluent dans le temps et modifient l’environnement (robots ...).

Chacun de ces rôles est assuré grâce aux capacités et propriétés des entités qui collaborent. En effet chaque agent est réactif, proactif ou autonome, assurant un lien dynamique et collaboratif entre lui et ses pairs. La différence entre les trois rôles dépend de l’environnement qui est un élément majeur que nous allons aborder dans la section suivante.

L’environnement

L’environnement est un point clé des systèmes multi-agents [31]. C’est un espace commun aux agents qui supporte les activités des agents et diffuse l’information. C’est une donnée subjective qui varie suivant la focale d’observation. Par exemple, dans un contexte de prédiction des risques dans un bloc opératoire, l’environnement des agents infectieux (groupés en une espèce) comprend les praticiens, le matériel chirurgical, le bloc opératoire mais pas eux-mêmes.

Dans le cadre de systèmes hétérogènes, les capacités et perceptions varient d’un agent à l’autre (cf 3.3.3). L’agent qui agit en modifiant l’environnement n’est pas forcément celui qui perçoit cette modification. De plus, plusieurs environnements propres à l’agent co-existent : d’un côté l’environnement physique constitué d’un espace avec une métrique et de l’autre l’environnement social composé des autres agents connus. L’éventail des propriétés propres aux environnements des SMA coïncide avec l’immense variété d’applications à traiter [44]. Ainsi l’environnement peut-être :

- Accessible / Inaccessible : un agent accède à l’état complet de l’environnement pour (entre autres) choisir une action ; l’environnement est accessible à l’agent (inutile de conserver les changements de l’environnement).
- Déterministe / Non déterministe : le prochain état de l’environnement est déterminé par l’état courant et par l’action de l’agent.
- Épisodique / Non épisodique : les prochaines évolutions ne dépendent pas des actions déjà réalisées.
- Statique / Dynamique : l’environnement ne change pas pendant que l’agent réfléchit.
- Discret / Continu : le nombre de perceptions et d’actions est limité.
- Avec / Sans adversaires rationnels.

On ne peut parler de système complexe et interactif sans évoquer l’organisation. Un SMA entre dans cette logique qui est abordée dans le paragraphe qui suit.

L’organisation

Le fonctionnement d’un SMA est directement dépendant de l’organisation des agents intégrés

au SMA. Il s'agit de définir les acteurs, leurs rôles, et de quelle manière ils évoluent dans le système. De telles structures révèlent le caractère dynamique de ces organisations et les relations possibles entre groupes d'agents [66]. En effet il existe une mobilité des agents qui leur confère la capacité d'acquérir et de conserver de nouvelles aptitudes lorsqu'ils passent d'un groupe à l'autre. A cela s'ajoute la multiplicité des rôles attribuables aux agents qui évoluent dans plusieurs groupes. A titre d'exemple, un enseignant chercheur en IA (intelligence artificielle) peut également être passionné de football. Il peut moduler ses compétences et son langage respectivement en fonction du milieu dans lequel il évolue et de la population côtoyée. Il existe de nombreuses organisations et implémentations possibles qui définissent le comportement et l'autonomie des agents [43]. Le Tableau 3.2 donne une synthèse de celles qui sont les plus significatives.

Organisation	Description
Hiérarchies	Un agent est lié à un autre agent par une relation d'autorité
Holarchies	Les agents sont organisés en holons : un sous groupe d'agents est subsumé par un autre agent ou un groupe
Coalitions	Collaboration temporaire entre agents pour optimiser les intérêts individuels
Équipes	Collaboration collective au sein d'une même équipe dans un objectif commun
Congrégations	Les congrégations sont des collaborations entre agents destinées à la réalisation de plusieurs objectifs. Les agents peuvent entrer et sortir des congrégations
Sociétés	Les agents sont hétérogènes mais soumis aux mêmes normes
Fédérations	Les interactions se réalisent uniquement via un délégué
Matrices	Un agent peut être soumis à l'autorité de plusieurs agents
Combinaisons	Combinaisons entre plusieurs organisations

Tableau 3.2 – Organisations SMA

L'émergence est une caractéristique importante des systèmes multi-agents. Ce thème est abordé dans le paragraphe qui suit.

Émergence

Un phénomène est qualifié d'émergent lorsqu'il est perceptible au niveau global suite aux interactions locales du système [83]. Ce sont donc des phénomènes soudains et imprévisibles qui possèdent les propriétés suivantes [33] :

- ostensible : le phénomène est observable au niveau macro ;
- radicalement nouveau : le phénomène est inexplicable au niveau micro ;
- cohérent et corrélé : le phénomène a sa propre identité tout en étant lié aux parties du niveau micro ;
- dynamique particulière : le phénomène n'est pas prédéfini, il s'auto-crée et s'auto-maintient.

Ces propriétés caractérisent l'aspect dynamique et relativement stable dans la durée d'un phénomène qui lui confèrent une identité propre. Notons aussi l'importance de la notion de nouveauté du phénomène au niveau micro. Il en découle une décentralisation du contrôle au niveau macro, substituée par un contrôle passant par les entités micro, qui rend difficile, voire impossible, la prévision globale d'un phénomène. Cette impossibilité (description de la chaîne de causalité entre les niveaux) est le propre de l'émergence.

Cette notion ne prend tout son sens que si l'expertise de l'observateur s'avère insuffisante, car dans le cas contraire il est possible de définir toute la chaîne de causalité entre les niveaux macro et micro. Au final, l'émergence peut être définie comme un outil face à notre incapacité à déterminer tous les phénomènes complexes d'interactions au niveau micro. La Figure 3.4 illustre le cycle caractérisé par l'effet de l'émergence sur l'organisation des agents, elle-même régulée par des contraintes.



FIGURE 3.4 – Relation micro-macro dans les SMA [26]

Autant l'émergence montre qu'il n'est pas utile pour les agents de connaître le fonctionnement global, autant la question se pose concernant les méthodes à activer pour aboutir à une fonction nouvelle. L'auto-organisation est une piste possible sachant que par définition, un système est auto-organisé lorsqu'il modifie lui-même son organisation dans le but de s'adapter à son environnement. Or, comme nous l'avons vu précédemment, l'organisation détermine la fonction globale du SMA. C'est donc par un ajustement des différents paramètres liés aux agents tels que leurs comportements, leurs inter-relations, leurs créations ou leurs suppressions, que l'on optimisera l'organisation. Quelques exemples de modélisation de type SMA proches de notre thématique (prédiction des risques dans le monde médical) sont maintenant présentés.

SMA et prédiction des risques dans le monde médical

L'ensemble des modélisations destinées à prédire les risques dans le monde médical portent sur l'analyse d'organisations afin d'en éviter les écueils. Les environnements englobent soit des structures (un hôpital, un service ...) soit des natures d'individus (personnels soignants, non soignants ..) et souvent les deux. Dans ce premier exemple [12], l'auteure propose une approche de modélisation de patients inspirée de l'approche centrée produit du monde industriel. Le SMA implémenté est un outil de pilotage réactif, dynamique et distribué du parcours patient per-

mettant de minimiser les délais d'attente et la durée des séjours tout en assurant une meilleure gestion des ressources hospitalières. Dans [79], le travail de recherche consiste à élaborer un outil de modélisation de l'organisation des secours d'urgence. Cette implémentation, qui se veut opérationnelle, intègre des données télétransmises en temps réel afin d'évaluer différents modes organisationnels possibles.

L'étude [90] montre qu'une modélisation multi-agents est adaptée à l'amélioration des comportements nutritionnels. L'ontologie choisie comprend un agent de connaissances personnelles, un agent d'inférence floue (décrit la base de connaissances) et un agent de génération sémantique (évalue les régimes alimentaires). Grâce à cette approche les utilisateurs obtiennent des informations sur leur alimentation et améliorent ainsi leur hygiène de vie alimentaire. Ce dernier exemple [78] met en avant les systèmes de surveillance des patients à distance (SPD) encore trop sous-exploités actuellement. L'apport d'une modélisation multi-agents est selon les auteurs, un outil précieux en matière d'identification des obstacles et contraintes des systèmes SPD. Enfin, nous renvoyons le lecteur à la présentation des plateformes "Tour de contrôle" et "3D Virtual Operating Room" se trouvant dans la Section 4.8.4 du Chapitre 4. En effet, ces plateformes orientées toutes les deux sur la prédiction des risques sont des éléments de comparaison que nous avons jugés utiles pour le lecteur.

3.3.4 Analyse et bilan d'une modélisation SMA

Dans le cadre de structures complexes nous avons montré dans les précédents paragraphes que le système multi-agents présente de nombreux avantages. Parmi eux, figurent les interactions entre agents de types très différents. Ainsi, en lien avec notre étude, une relation au sens mathématique est envisageable entre des univers aussi distincts que sont par exemple la mesure de la fatigue humaine ou l'efficacité d'un matériel chirurgical. D'autre part, parce que les SMA sont facilement compréhensibles par l'ensemble des acteurs mobilisés dans des projets à structure complexe, les systèmes multi-agents entrent dans des démarches participatives essentielles à l'élaboration de décisions collectives.

On peut émettre cependant une réserve en ce qui concerne les difficultés soulevées par l'identification de phénomènes émergents et des processus qui conduisent à leur survenue. A cette réserve s'ajoute l'acquisition de connaissances d'un système considéré comme fermé par définition. C'est un des axes majeurs de notre réflexion qui nous a conduit à coupler notre SMA à des paradigmes tel que le raisonnement à partir de cas (RàPC) dont le principe consiste à résoudre des problèmes par analogie à partir d'expériences passées. Ce couplage permet aussi de délester le SMA de la résolution de problèmes et de l'auto-organisation qui en découle. Cette dernière peut en effet conduire à des choix parfois trop complexes entre un comportement maîtrisé de l'enchaînement des agents dans une réflexion *bottom-up*, et un comportement *top-down*, certes plus libre pour les agents mais déstabilisant pour le SMA.

Après avoir présenté les systèmes multi-agents aussi bien dans leur aspect fonctionnel que formel, nous abordons dans la section suivante les différents systèmes à base de connaissances.

Nous pensons en effet qu'il est difficile de dissocier les spécifications opérationnelles d'un système complexe de la notion de connaissances.

3.4 Les systèmes à base de connaissances

Le dictionnaire Larousse nous donne la définition de l'informatique suivante : "*Science du traitement automatique et rationnel de l'information considérée comme le support des connaissances et des communications*". A la lecture de cette définition et au regard des progrès de l'intelligence artificielle, nous pouvons convenir de la place prépondérante de l'apprentissage et des processus d'acquisition des connaissances propres aux systèmes actuels dédiés à la modélisation. Nous présentons dans les paragraphes qui suivent les principaux d'entre eux.

3.4.1 Les réseaux de neurones artificiels

Le cerveau humain possède une fabuleuse capacité de traitement de l'information qui confère à l'homme la faculté de comprendre, de s'adapter aux situations, de retenir, de créer, de résoudre des problèmes ... Cette capacité qui s'apparente à l'intelligence a été modélisée pour donner naissance à l'intelligence artificielle (IA), dont le but est de créer des systèmes capables d'avoir un comportement intelligent. Parmi les différentes techniques développées, les réseaux de neurones artificiels (RNA) sont aujourd'hui l'un des domaines qui suscite le plus d'intérêt. Ils s'inspirent très largement du réseau de neurones biologiques et la littérature est prolifique sur le sujet.

En ce qui concerne notre étude, nous nous intéressons à leur capacité à prédire ou bien encore à classer les données. Les quelques exemples qui suivent sont centrés sur notre contexte d'étude à savoir le bloc opératoire. [47] utilise les RNA afin de détecter les artefacts spectraux générés par les éclairages en salle d'opération qui limitent la robustesse de la spectroscopie Raman (utilisé dans la détection de cellules cancéreuses). Le couplage spectromètre RNA permet de détecter ces caractéristiques non linéaires complexes. La classification des arythmies cardiaques est essentielle lors de la surveillance des patients en salle d'opération. Dans cette optique, [65] propose une classification précise des signaux extraits d'électrocardiogrammes. Les réseaux de neurones sont aussi associés au sein du bloc opératoire comme outil de prévention. L'article [67] présente l'implémentation d'un RNA dans le but d'entraîner les respirateurs à distinguer une situation normale d'une alerte. Citons enfin [11] qui analyse plusieurs outils d'IA dont les RNA, utilisés afin d'optimiser l'organisation du bloc. C'est un point crucial qui impacte de manière transversale tous les processus liés à une intervention chirurgicale. Les deux paragraphes qui suivent décrivent le principe général d'un RNA.

Du neurone au réseau de neurones

Le principe de fonctionnement d'un neurone présenté dans la Figure 3.5 suit toujours les mêmes

étapes : les informations sont introduites sous la forme de signaux (en général des valeurs ≤ 1) via les entrées x_i ($i \in \mathbb{N}$). Le neurone k reçoit ces valeurs auxquelles sont affectés des poids W_{ki} . La fonction de combinaison C_k calcule ensuite le potentiel du neurone k qui correspond à la somme des poids à laquelle est ajouté un seuil s_k : $C_k = s_k + \sum_{(i,k) \in \mathbb{N}^2} x_i W_{ki}$. La fonction φ est ensuite appliquée au potentiel calculé par C_k et retourne la valeur de sortie y_k . Plusieurs types de fonction d'activation existent. Parmi les principales se trouvent :

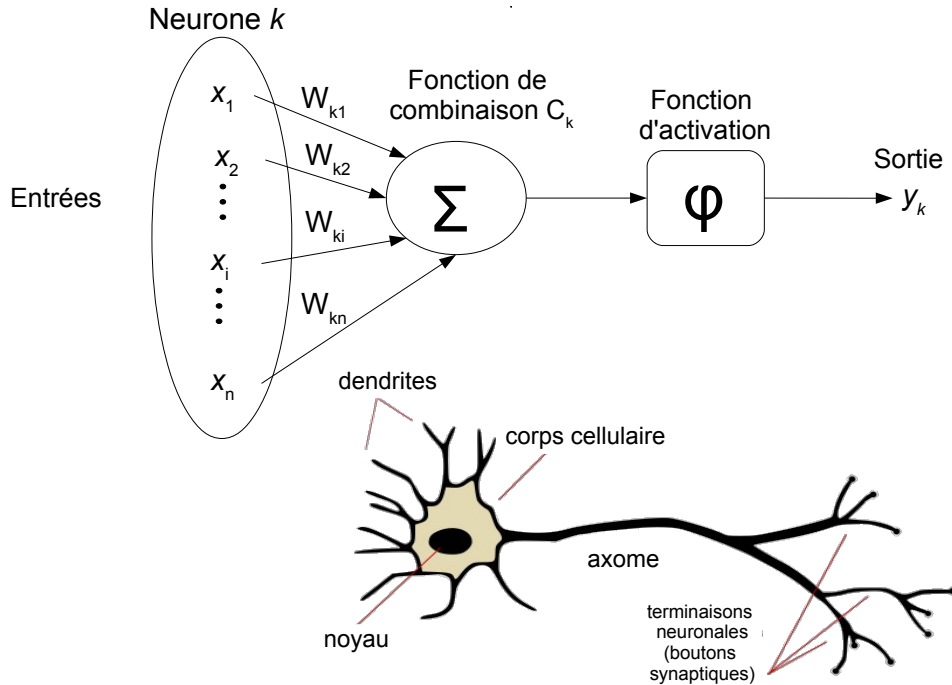


FIGURE 3.5 – Neurone artificiel

- fonction seuil : $\varphi(x) = \begin{cases} 1 & \text{si } \Sigma > \text{seuil} \\ 0 & \text{sinon} \end{cases}$.
- fonction Sigmoidé : $\varphi(x) = \frac{1}{1 + e^{-x}}$.
- fonction Tanh : $\varphi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.
- fonction ReLU : $\varphi(x) = \max(0, x)$.

Hormis la fonction seuil, toutes les autres sont continues et dérivables et permettent un apprentissage automatique qui est explicité dans le paragraphe qui suit. Le principe décrit pour un neurone reste le même lorsque le modèle inclut plusieurs neurones qui forment dès lors un réseau. La Figure 3.6 montre un réseau de deux neurones en parallèles. Nous retrouvons comme pour la présentation d'un neurone : une couche d'entrées (x_1, x_2, x_3), des fonctions de

combinaison C_k , des poids W_{ki} , des fonctions d'activation φ et une couche de sortie (y_1, y_2) . La plupart des réseaux actuels comportent des couches intermédiaires de neurones dont le nombre varie en fonction de la complexité du problème à résoudre.

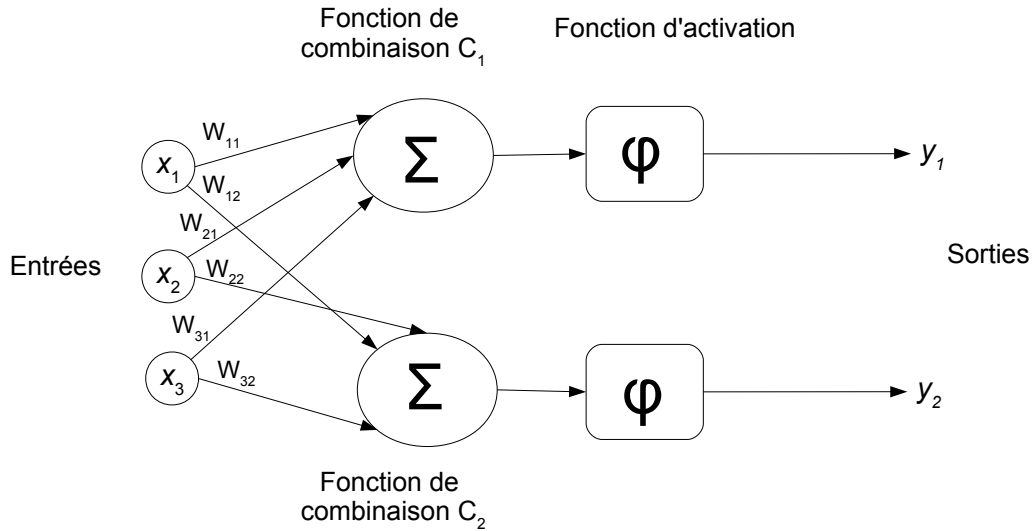


FIGURE 3.6 – Réseau de deux neurones

Apprentissage

L'apprentissage automatique au sens large consiste à acquérir la capacité à apprendre à partir de données. Il s'agit donc pour une machine d'augmenter ses compétences dans le traitement de certaines tâches sans programmation spécifique de ces dernières. Deux types d'apprentissages existent : l'apprentissage supervisé et l'apprentissage non supervisé. Dans le premier cas, l'apprentissage est dépendant de la comparaison à la sortie du réseau, de la valeur prédite avec la valeur réelle. Ainsi, comme le montre la Figure 3.7, une valeur de coût notée $C = \frac{(Y - W)^2}{2}$ est calculée puis injectée dans un neurone qui va ensuite ajuster les poids synaptiques. Y correspond à la valeur prédite et W à la valeur réelle. L'apprentissage qui en découle est un processus itératif visant à minimiser la valeur de C . C'est l'apprentissage utilisé avec le paradigme Perceptron. Dans le second cas, l'apprentissage se fait de manière autonome sans que l'on fournisse au système des données réelles à comparer en utilisant par exemple la règle de Hebb ("les neurones qui déchargent ensemble forment entre eux des circuits préférentiels"). Sachant que la méthode choisie pour notre étude est la méthode supervisée, l'apprentissage non supervisé dans le contexte du RNA ne sera pas développé davantage dans ce manuscrit. En effet dans notre contexte d'utilisation, nous avons utilisé comme jeux de données d'entraînement quelques cas

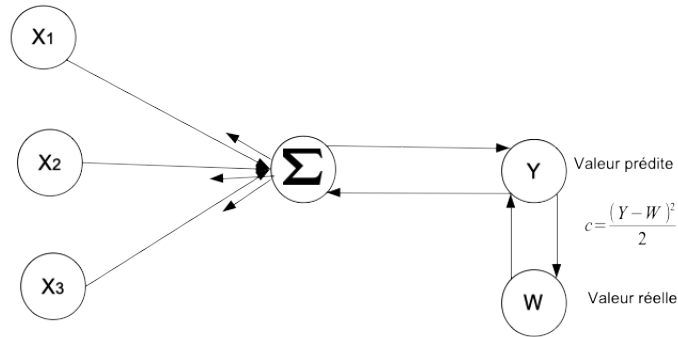


FIGURE 3.7 – Apprentissage d’un Perceptron

réels suffisamment significatifs.

Bilan RNA

Le réseau de neurones est un paradigme éprouvé et efficace en matière de prédictivité. Le caractère non linéaire de ces systèmes permet d’intégrer des environnements tel que le bloc opératoire (données avec des échelles et métriques différentes). A cela s’ajoute l’adaptabilité et la souplesse des RNA, nécessaires dans des environnements fluctuants. Cependant, l’efficacité de ces systèmes, qui dépend de l’apprentissage à partir de données massives, risque d’être un frein dans un contexte où nous ne disposons que de peu de données réelles. Le raisonnement à partir de cas palliant cet écueil, est décrit dans la section suivante.

3.4.2 Le raisonnement à partir de cas

Le raisonnement à partir de cas résulte d’une convergence entre l’intelligence artificielle et la psychologie cognitive. Son principe de fonctionnement consiste à résoudre des problèmes (appelés cas) par analogie avec des expériences passées. Il a émergé dans les années 80 lorsqu’un intérêt tout particulier est apporté sur le rôle de la mémoire humaine dans la résolution de problèmes [75]. L’idée générale de ce courant stipule que la résolution d’un problème est toujours issue d’expériences passées analysées et adaptées (si nécessaire). Plusieurs laboratoires ont alors apporté leurs contributions, et c’est Janet Kolodner qui fut à l’origine de la première implémentation (CYRUS) [50]. Trois principaux courants que nous allons expliciter dans les trois paragraphes qui suivent ont ensuite succédé à CYRUS.

RàPC interactif

Ce type de raisonnement est appliqué lorsqu'il existe une interaction entre l'utilisateur et le système [6]. Les exemples de raisonnement à partir de cas "conversationnels" en sont les plus représentatifs. Les cas sont composés de classifications textuelles de conversations, d'un ensemble de couples (question, réponse) et d'un ensemble d'actions [5].

RàPC comme systèmes de recommandation

Le raisonnement à partir de cas est considéré comme un système de recommandation lorsqu'il suggère un ou plusieurs produits en fonction des préférences de l'utilisateur. Le crible des produits résulte de l'analyse de données les plus proches possibles du problème cible (préférence utilisateur). Les bases de connaissances sont en générales assez volumineuses comme dans l'exemple [8] où l'auteur propose un choix musical à partir d'un titre de chanson.

RàPC textuels

Le raisonnement à base de cas textuels vise à résoudre des problèmes à partir d'expériences dont la description figure dans des documents textuels [52]. Les cas textuels peuvent être non-structurés ou semi-structurés. Dans le premier cas, leur description est libre et dans le second cas le texte est découpé en portions étiquetées ("problème", "solution"). Le souhait d'obtenir de manière automatique le résumé d'un jugement légal est un exemple de ce type de modélisation [63].

Nous proposons après ce bref historique de donner une définition et les principes du raisonnement à partir de cas.

Définition

Selon [75], raisonner à partir de cas revient à résoudre un nouveau problème nommé "*problème cible*" à partir d'un ensemble de problèmes déjà résolus. Le cas source représente une expérience de résolution de problèmes passés et appartient à un ensemble nommé "base de cas". Le cas cible correspond quant à lui au nouveau problème à résoudre. Le but d'un RàPC se résume à l'utilisation d'expériences passée(s) et à leur(s) éventuelle(s) adaptation(s) afin de résoudre un nouveau problème. De manière plus formelle, un cas source est un couple ($srce, Sol(srce)$) où $srce$ désigne un problème source et $Sol(srce)$ sa solution. Un cas cible est un couple ($cible, Sol(cible)$) où $cible$ désigne un problème $cible$ et $Sol(cible)$ sa solution.

Principe

Le principe du raisonnement à partir de cas est calqué sur le raisonnement par analogie. Il consiste à mettre en correspondance une situation en cours appelée "*cible*" avec une situation issue d'une expérience passée donc déjà existante appelée "*source*". Ainsi, si deux situations sont analogues, alors il est possible d'établir des relations entre le problème cible et le problème source afin d'utiliser la solution source en réponse à notre problème cible. Le carré d'analogie [62] (cf Figure 3.8) résume dans sa globalité le principe que nous venons d'évoquer. Dans ce

“carré”, un cas source est sélectionné en fonction de la valeur de descripteurs d’un problème cible. La similarité entre les deux cas résulte d’une mesure notée α . Sachant que les descripteurs du cas cible sont théoriquement similaires, il est possible d’utiliser les éléments de la solution source après d’éventuelles adaptations. Les éléments de la solution source sont reliés par des relations de dépendance β qui sont reproduites par analogie sur le cas cible. Dès lors, une adaptation de la solution source peut-être envisagée en fonction de l’écart α (mesure de similarité) pour déterminer la solution cible.

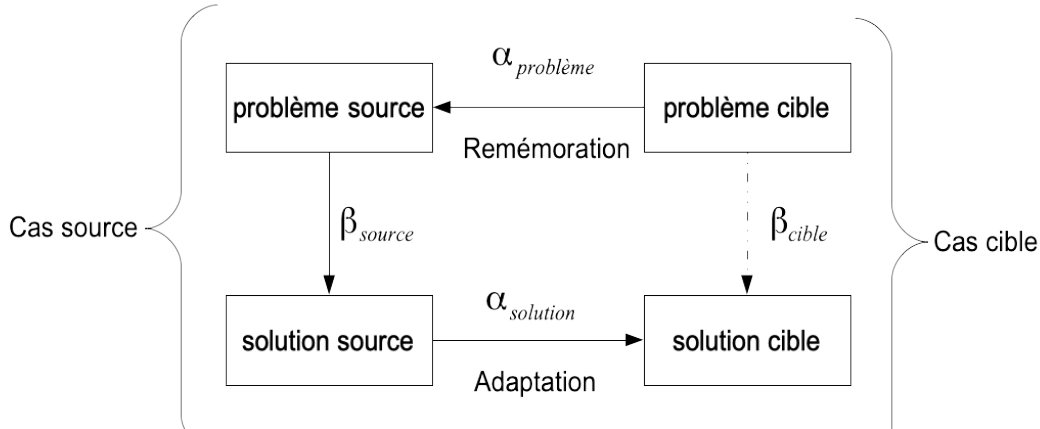


FIGURE 3.8 – Carré d’analogie [62]

Le raisonnement à partir de cas impose un certain formalisme que l’on retrouve notamment dans la représentation d’un cas. Celui-ci est scindé en deux parties : un problème souvent noté pb et sa solution souvent notée $Sol(pb)$. Que ce soient pb ou $Sol(pb)$, ils sont décrits par un ensemble de descripteurs (souvent notés d) et chacun d’eux est formalisé par un couple (*attribut, valeur*). Dès lors, un cas source est représenté par un tuple $(d_1^s, \dots, d_n^s, D_1^s, \dots, D_p^s)$ où d_i^s est le $i^{ème}$ descripteur du problème source et où D_k^s est la $k^{ème}$ solution du problème source. On retrouve le même formalisme avec le cas cible : $(d_1^c, \dots, d_n^c, D_1^c, \dots, D_p^c)$ où d_i^c est le $i^{ème}$ descripteur du problème cible et où D_k^c est la $k^{ème}$ solution du problème cible.

Le processus de résolution d’un problème propre au RàPC est caractérisé par un cycle à cinq étapes. Le premier cycle proposé par Aamodt et Plaza [1] comportait les phases Remémoration, Réutilisation, Révision et Mémorisation. Ce dernier à été enrichi par Mille [61] grâce à l’apport de la phase Élaboration. Les différentes phases illustrées par la Figure 3.9 sont décrites dans les cinq paragraphes qui suivent.

Élaboration

L’efficacité d’un RàPC est à rapprocher de l’élaboration d’un cas. C’est une étape qui consiste à formaliser la description d’un problème cible en amont de la phase de remémoration. Les descripteurs du cas sont choisis de manière à optimiser la recherche de cas similaires ainsi que leurs

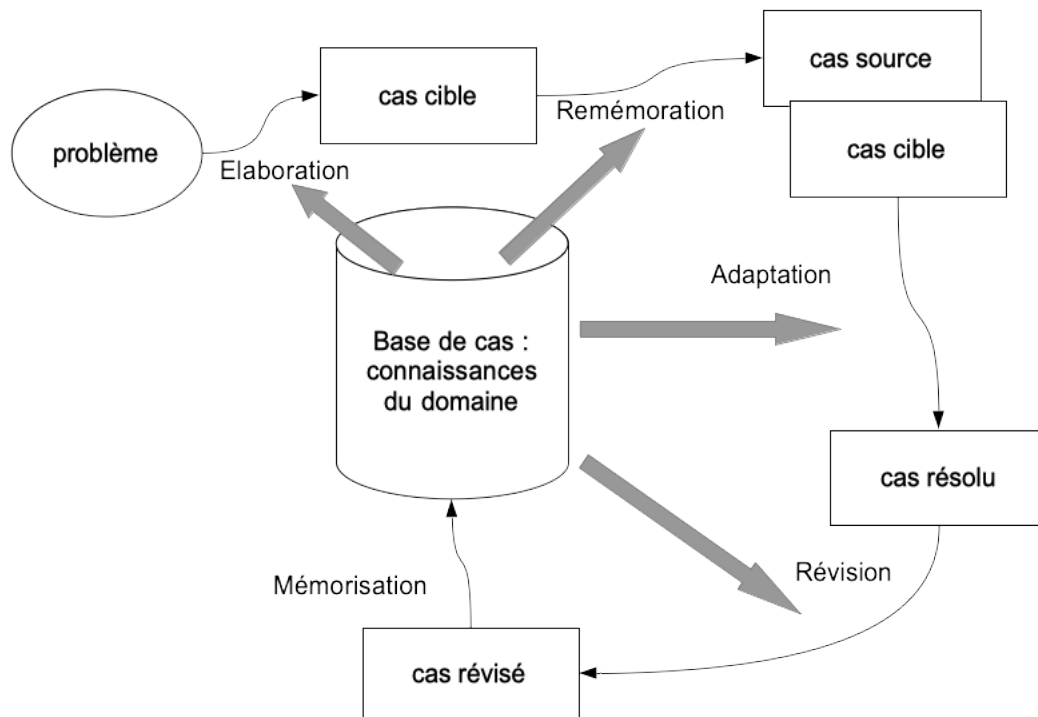


FIGURE 3.9 – Cycle RàPC

éventuelles adaptations. Le cas cible sera donc défini à partir des connaissances du domaine, nécessaires au renseignement des différents descripteurs considérés comme significatifs.

Remémoration

Le raisonnement à partir de cas a été construit selon l'hypothèse qu'un ensemble d'expériences est potentiellement utile à la résolution d'un problème similaire. Ces expériences enrichissent une base de cas au fur et à mesure de l'apprentissage. La phase de remémoration est destinée à chercher des cas sources issus de cette base et similaires au problème cible. Il s'agit plus précisément de sélectionner le meilleur cas source. Cette sélection résulte d'un calcul de similarité entre le cas cible et les cas extraits d'un ensemble de cas sources. Les mesures sont appliquées à leurs descripteurs. La littérature en recense un grand nombre qui convergent vers un calcul de similarités locales ou globales. Ces deux types de calculs sont décrits dans les deux paragraphes qui suivent.

Calcul de similarités locales Il s'applique sur les caractéristiques d'un cas. Elles peuvent être quantitatives (fréquence cardiaque d'un patient ...) ou qualitatives (type de bistouri utilisé ...). Ces calculs sont liés aux types de descripteurs et aux échelles considérées : une fréquence cardiaque égale à 150 pulsations/min est considérée comme similaire à une fréquence cardiaque

égale à 180 pulsations/min, en revanche une température corporelle égale à 38° n'est pas considérée comme similaire à une température égale à 40°. Ces calculs sont essentiellement basés sur les distances comme le montrent les exemples extraits de [76] :

- descripteurs numériques : $sim(a, b) = 1 - \frac{|a-b|}{range}$;
- descripteurs symboliques mono-valeurs : $sim(a, b) = \begin{cases} 1 & \text{pour } a = b \\ 0 & \text{pour } a \neq b \end{cases}$;
- descripteurs symboliques multi-valeurs : $sim(x, y) = \frac{card(x) \cap card(y)}{card(x \cup y)}$.

a et b sont les valeurs des descripteurs, $card(x)$ et $card(y)$ correspondant respectivement à la cardinalité de l'ensemble x et de l'ensemble y , et $range$ est la valeur absolue de la différence entre la borne supérieure et la borne inférieure de l'ensemble des valeurs traitées.

Calcul de similarité globale Il est appliqué sur un ensemble d'attributs. Cela revient donc en général à calculer la moyenne d'un ensemble de similarités locales en pondérant si nécessaire certains attributs. Les calculs de distances se distinguent les uns des autres en fonction de la granularité attendue. Citons principalement :

- Weighted Block city : $sim(A, B) = \sum_{i=0}^n w_i sim_i(a_i, b_i)$.
- la distance de Minkowski : $sim(A, B) = \left[\frac{1}{n} \sum_{i=0}^n [sim_i(a_i, b_i)]^r \right]^{\frac{1}{r}}$ (la distance est Euclidienne si $r = 2$ ou de Manhattan si $r = 1$).

n est le nombre d'attributs, w_i le poids du $i^{ème}$ attribut ($\sum_{i=1}^n w_i = 1$) et $sim_i(a_i, b_i)$ la similarité locale calculée pour les valeurs a_i et b_i du $i^{ème}$ attribut.

Réutilisation

Lors de la phase de réutilisation, une solution est proposée à un nouveau problème à partir d'une ou plusieurs solutions extraites d'un ou plusieurs cas remémorés. Dès lors, la solution peut-être utilisée telle quelle, ou bien elle doit nécessiter une modification. Dans le premier cas de figure la solution est simplement copiée et dans le second cas de figure une adaptation s'impose. On retrouve dans [7] trois types d'adaptations possibles :

- Réinstantiation : le cas source est suffisamment proche du cas cible pour que la solution issue du cas source soit copiée sans modification.
- Adaptation substitutionnelle : une substitution est opérée sur le descripteur de la solution source remémorée lorsqu'elle est considérée comme non pertinente en l'état.
- Adaptation transformationnelle : cette adaptation est un recours à l'adaptation substitutionnelle lorsqu'elle n'est pas applicable. La solution attendue est alors dérivée du ou des cas remémorés en fonction des contraintes et caractéristiques exigées.

Jean Lieber présente dans [56] une extension de cette notion d'apprentissage. Il décrit par exemple l'adaptation par généralisation/spécialisation et l'adaptation par abstraction/raffinement.

Il distingue la généralisation de l'abstraction ainsi : "La différence principale entre ces deux notions est que la connaissance d'une solution $Sol(abst)$ d'un problème plus abstrait que pb n'est généralement pas suffisante pour la résolution de pb alors que la connaissance d'une solution $Sol(gén)$ d'un problème $gén$ plus général que pb l'est."

Révision

La phase de révision permet de tester la solution produite (copiée ou adaptée) lors de l'étape de réutilisation. L'analyse et l'apprentissage portés sur les échecs non-prévus peuvent enrichir et corriger les connaissances du système [74]. La révision se compose de deux étapes : l'évaluation de la solution (par un système expert, par simulation, par comparaison avec le monde réel) et son éventuelle correction à partir des connaissances du domaine. Ce n'est que lorsque l'évaluation est satisfaisante que la nouvelle solution retenue peut être mémorisée.

Mémorisation

L'enrichissement de la base de cas est tributaire de l'enregistrement de nouveaux cas validés lors de la phase de révision. Cela implique parfois la ressemblance avec des cas déjà existants qui peut conduire à la création de doublons. Il convient donc de prendre en compte cette contrainte en amont de la sauvegarde d'un cas. En pratique un cas est enregistré dans la base lorsqu'il est issu d'un ancien cas qui subsume le nouveau cas ou lorsqu'il ne dépend pas de cas sources (remémoration infructueuse ou nouvelle proposition d'expert).

Les applications du raisonnement à partir de cas sont nombreuses que ce soient dans le domaine industriel [2], dans l'analyse de documents textuels [63], dans l'analyse d'images [59] ou bien encore dans le domaine médical [18] pour n'en citer que les principaux. Le RàPC tient d'ailleurs toutes ses promesses dans la prédiction et l'anticipation des risques ayant un impact sur la santé. La littérature prolifique sur le sujet révèle deux grandes tendances. La première met plutôt l'accent sur des maladies précises alors que la seconde relève plutôt des organisations. Nous citerons quelques exemples de ces deux mouvements dans les deux paragraphes qui suivent.

Les risques liés aux maladies

Dans [18] l'auteur propose un système auxiliaire de diagnostic des maladies du foie (principale cause de mortalité à Taïwan) dont la détection précoce serait profitable aux patients. L'objectif est d'intégrer le RàPC à différentes méthodes de classification (RNA, arbre de classification et de régression, régression logistique) pour les rendre plus précis afin d'éviter les faux diagnostics. L'étude [77] cherche à montrer l'efficacité de l'anticipation d'une chute d'une personne âgée. Les captures de mouvements sont intégrées dans un outil d'aide à la décision (système RàPC) afin de fournir au physiothérapeute un pronostic quant aux risques de chutes graves. Dans ce dernier exemple [45], l'auteur souhaite à l'aide d'une approche RàPC prédire l'effet de différents traitements sur les troubles anxieux. Cette aide à la décision est précieuse dans un contexte où l'équipe médicale administre le traitement ayant les meilleures taux de réussite.

Les risques liés aux organisations

L'organisation d'interventions d'urgence basée sur la téléphonie mobile est au cœur de l'article [39]. Actuellement, les notifications ne sont que des messages courts (SMS) et fournissent trop peu d'éléments relatifs aux catastrophes d'origines humaines. L'alternative RàPC propose de mettre en place un processus de récupération de catastrophes passées et répertoriées dans une base de cas en appui de l'ontologie SRUM (Système de Réponse d'Urgence Mobile). Parmi les pistes de recherche en très forte croissance figure la télémédecine. Cette pratique médicale, qui consiste à échanger des informations entre un professionnel de santé et son patient évite des déplacements inutiles et trouve tout son sens dans des situations exceptionnelles comme le confinement. Dans l'article [81] le cycle du RàPC est calqué sur une consultation type (établir le diagnostic, récupérer le traitement, appliquer les preuves, adapter, retenir) et s'appuie sur des connaissances antérieures (base de cas) avalisées par les preuves. Ce dernier exemple [72] présente un outil construit selon le paradigme RàPC, qui cartographie les pratiques en matière de soins de santé lorsque plusieurs spécialités sont sollicitées pour un pronostic final. L'objectif est de décliner toutes les collaborations possibles.

Bilan sur l'utilisation des systèmes RàPC

La recherche de solutions par analogie avec des problèmes résultants d'expériences similaires passées permet de prendre une décision malgré une connaissance limitée du domaine. C'est un atout majeur dans une configuration initialisée avec un faible jeu de données réelles. Cet avantage est à contrebalancer avec la rigidité qu'impose la structure de ces systèmes. C'est pourquoi nous avons opté pour un couplage de notre système multi-agents avec le raisonnement à partir de cas. Ainsi, nous avons associé l'intérêt d'une approche de résolution de problèmes par analogie à celui des dynamiques d'interactions et d'auto-organisation propres aux SMA. La section qui suit en donne deux exemples extraits de la littérature.

3.4.3 Couplage avec RàPC : quelques exemples significatifs

Dans les parties précédentes, nous avons présenté le raisonnement à partir de cas ainsi que les systèmes multi-agents. Nous rappelons que leur couplage permet selon nous, de combiner les notions d'émergence et d'interactions propres aux SMA avec la résolution de problèmes par analogie propre au RàPC. Les trois exemples qui suivent illustrent ce type de couplage.

Dans sa thèse, Addelhamid ZOUHAIR, [97] propose "*une architecture multi-agents multi-couches fondée sur le raisonnement à partir de cas dynamiques et incrémentables (RàPCDI) capable d'étudier les situations dynamiques (reconnaissance, prédiction, et apprentissage de situations)*". Sa principale contribution porte sur la transformation de cycles RàPC linéaires en cycle RàPC dynamiques structurés en étoile. La plateforme proposée IDCBR-MAS couple le RàPC au SMA pour palier les manques de dynamique dans les calculs de similarité, indispensables à l'adaptation du système. Cette vision est certes intéressante, mais elle vise essentiellement à enrichir la phase de "remémoration" du cycle RàPC grâce aux capacités d'adaptabilité

et d'autonomie des SMA et ne correspond pas à notre objectif. En effet nous ne cherchons pas à améliorer l'un des cycles du RàPC mais plutôt à utiliser l'ensemble de ses propriétés pour enrichir notre système multi-agents.

Olivier Chator [19] décrit quant à lui un système collaboratif en charge de répondre aux attentes d'un maître d'œuvre. Ce système multi-agents doit donc être capable de définir un projet, d'identifier les acteurs et leurs compétences puis d'évaluer le projet à posteriori. Il présente dans ses travaux une entité logicielle (Agent Compétence) "*vivante et autonome*" intégrée au sein d'un SMA possédant son propre cycle de vie, ainsi qu'un modèle comportemental personnalisé et évolutif. A l'issue, une bourse de compétences enrichie par les expériences sera interrogée par les acteurs humains. Chacune des compétences n'est plus intégrée à une entité, mais peut être partagée par plusieurs d'entre elles. Ce système dynamique impose l'acquisition active de connaissances. Le raisonnement à partir de cas a été choisi dans cette perspective car il intègre dans son principe d'adaptabilité et de mémorisation la plasticité attendue. Cependant notre objectif de modélisation orientée agents diffère de cette idée dont le concept privilégie la mutualisation des compétences. En effet, même si ce partage de connaissances et de compétences existe dans notre concept, chaque agent possède ses propres comportements qu'il enrichit au fil des expériences grâce à une mémoire individuelle. En effet, nous pensons qu'il est essentiel que les buts et les objectifs définis puissent être individualisés pour certains agents car cette approche bottom-up caractérise les émergences.

Citons enfin la plateforme POMAESS [95], une modélisation multi-agents où chaque agent est expert en résolution de problème machine et se situe sur des sites industriels différents. Dans le but d'améliorer la capacité dans la prise de décision, la méthode de raisonnement à partir de cas a été intégrée au SMA. Même si ce contexte se rapproche dans l'esprit du nôtre, une différence significative existe. En effet, dans POMAESS, les agents coopèrent dans un objectif commun (résolution de problèmes), mais contrairement à notre modèle, ils ne produisent pas les événements qu'ils analysent.

L'architecture SMA/RàPC que nous avons implémentée a été enrichie d'outils d'aide à la décision inspirés de méthodes de clusterisation, dans le but de regrouper des données variant de manière similaire dans un intervalle de temps donné. La description de ces paradigmes est l'objet de la section suivante.

3.5 Les méthodes de clusterisation

Le regroupement de données est une question fondamentale dans l'apprentissage non supervisé. Plusieurs méthodes de clustering ont été proposées dans la littérature pour partitionner les ensembles de données indépendamment du domaine. La clusterisation est souvent réalisée en amont de l'apprentissage futur. Les méthodes classiques de clustering sont divisées en 8 catégories. Le Tableau 3.3 donne le type d'algorithmes, en résume leur principe de fonctionnement et cite quelques algorithmes typiques.

Type	Principe de la méthode	Algorithmes
<i>Partition</i>	Le centre de gravité des points est considéré comme le centre du cluster	<i>k-means</i> [57] k-medoids [69]
<i>Hierarchie</i>	Les données sont regroupées en fonction de leurs relations hiérarchiques	BIRCH [96] CURE [38]
<i>Théorie floue</i>	La valeur discrète de l'étiquette d'appartenance $\{0, 1\}$, est transformée en un intervalle continu $[0, 1]$, pour décrire de manière plus souple les relations d'appartenance entre les objets	FCM [14] FCS [22]
<i>Distribution</i>	Les données, générées à partir de la même distribution, appartiennent au même cluster sachant qu'il y a plusieurs distributions dans les données originales	GMM [73]
<i>Densité</i>	Les données dans la région à haute densité de l'espace de données appartiennent au même cluster	DBSCAN [25] Mean Shift [20]
<i>Grille</i>	L'espace de données original est transformé en une structure de grille dont la taille est définie pour la clusterisation	STING [91] CLIQUE [4]
<i>Théorie fractal</i>	L'espace est divisé en plusieurs parties qui partagent certaines caractéristiques communes avec l'ensemble	FC [9]
<i>Modèle</i>	Sélection d'un modèle particulier qui convient le mieux à chaque cluster	COBWEB [49] GMM [73]

Tableau 3.3 – Summary of typical clustering methods

En plus de ces algorithmes standards, des algorithmes plus récents basés sur des *Noyaux* [29], sur des *Graphes spectraux* [88] ou sur l'*Affinité de propagation* [30] ont retenu notre attention. En effet, ils sont adaptés à nos variables continues et/ou discrètes, robustes et peu sensibles au bruit. Dans notre étude, nous comparons les méthodes classiques de partitionnement qui nous semblent adaptées à notre contexte. Ceci exclut de nos choix les méthodes *Théorie fractale* et *Densité* plutôt destinées à la reconnaissance des formes, ainsi que la méthode *Grille* réservée aux représentations multidimensionnelles. Nous comparons également les trois méthodes récentes mentionnées précédemment. Ces différentes méthodes sont présentées dans les paragraphes qui suivent.

3.5.1 Méthode basée sur la *Partition*

La méthode de clustering *Partition* permet de classer automatiquement des groupes "natu- rels". Formellement, le principe consiste à déterminer pour un certain nombre de classes fixes, une distribution d'un ensemble de points qui minimise un critère appelé "inertie" ou variance "intra-classe". L'algorithme itératif du *k-means* [57] minimise la somme des distances entre

chaque individu et un centroïde (centre de gravité d'un ensemble de points). La répartition des points de chaque cluster évolue tant que cette somme diminue.

L'algorithme *k-medoids* [69] est une variante de *k-means* qui prend le point le plus proche du centre des points de données comme référence de cluster. Cette amélioration rend cet algorithme moins sensible aux valeurs extrêmes.

3.5.2 Méthode basée sur la *Hiérarchie*

Comme la méthode *k-means*, la classification hiérarchique ascendante [92] permet de déterminer des classes homogènes. C'est une méthode hiérarchique construite du niveau le plus fin (un seul individu) au niveau le plus agrégé (regroupement de tous les individus). Ainsi, chaque point est tout d'abord considéré comme un cluster, avant de rechercher les deux clusters les plus proches. Cette étape est répétée jusqu'au regroupement de tous les points en un seul cluster. Nous obtenons une décomposition qui minimise l'inertie intra-classe et maximise l'inertie inter-classe (dispersion des nuages) comme le montre la Figure 3.10. L'isolement des valeurs aberrantes au sein d'un cluster est ainsi évité. Le résultat de cet l'algorithme se présente sous la forme d'un dendrogramme.

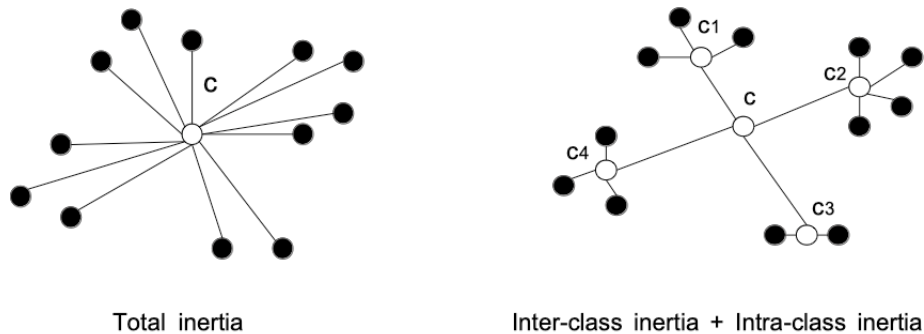


FIGURE 3.10 – Calcul de Ward

Dans cette méthode, la relation hiérarchique entre les clusters est facilement détectée, mais la complexité temporelle reste élevée. L'algorithme BIRCH [96] pallie cette limite. En effet, il réduit la taille des données initiales en les résumant dans une structure hiérarchique appelée *cluster caractéristique* comprenant : le nombre de points, la somme des coordonnées des points, les coordonnées du centre du groupe et son rayon. Cette pré-étape, qui permet le traitement de grands ensembles de données est suivie d'une classification hiérarchique ascendante (décrite ci-dessus) appliquée au centre du groupe.

3.5.3 Méthode basée sur la *Théorie floue*

Dans une classification floue non supervisée, la notion d'ensemble flou est introduite dans la définition des classes. Ainsi, l'appartenance de chaque point à un cluster est pondérée par un degré d'appartenance et chaque cluster a un centre de gravité. De manière équivalente aux autres algorithmes de clustering, les distances intra-classes sont minimisées et les distances inter-classes sont maximisées. La différence majeure réside dans le degré d'appartenance à un cluster attribué à chaque point. Cette méthode floue donne une probabilité d'appartenance réaliste et donc une précision de clustering relativement élevée. Cependant, la sensibilité aux valeurs initiales des paramètres nécessite la détermination préalable du nombre de clusters. L'algorithme FCM [14], qui est testé dans notre étude est certainement approprié mais nécessite une connaissance préalable du nombre de classes.

3.5.4 Méthode basée sur la *Distribution*

Dans cette approche, les données sont composées de distributions telles que les distributions gaussiennes. Les données sont regroupées selon une certaine distribution, sachant que la distance par rapport au centre d'un point est inversement proportionnelle à la probabilité qu'un point appartienne à la distribution. La figure 3.11 illustre une distribution gaussienne. Dans cet exemple, le nombre de cluster k est égal à trois, μ_i (espérance) désigne le centre du cluster $i \in [1 ; 3]$ et σ_i indique la variance.

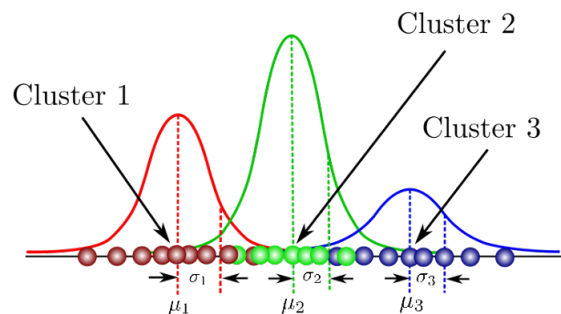


FIGURE 3.11 – Distribution gaussienne [16]

Comme la *Floue théorie*, cette méthode basée sur la distribution est efficace en termes de probabilité d'appartenance à un cluster. En revanche, la complexité temporelle peut être élevée et le paramétrage, parfois complexe a une influence sur le résultat et peut produire des aberrations.

L'algorithme GMM [73] applique cette méthode en utilisant des distributions gaussiennes.

3.5.5 Méthode basée sur les *Noyaux*

Le clustering basé sur les noyaux [29] a surmonté la difficulté pour séparer les groupes de manière non linéaire. Cette méthode de clustering cartographie les données d'entrée dans un espace multidimensionnel afin d'analyser les clusters. La figure 3.12 présente un exemple de mappage de données dans un espace caractéristique multidimensionnel, ce qui permet de les séparer facilement.

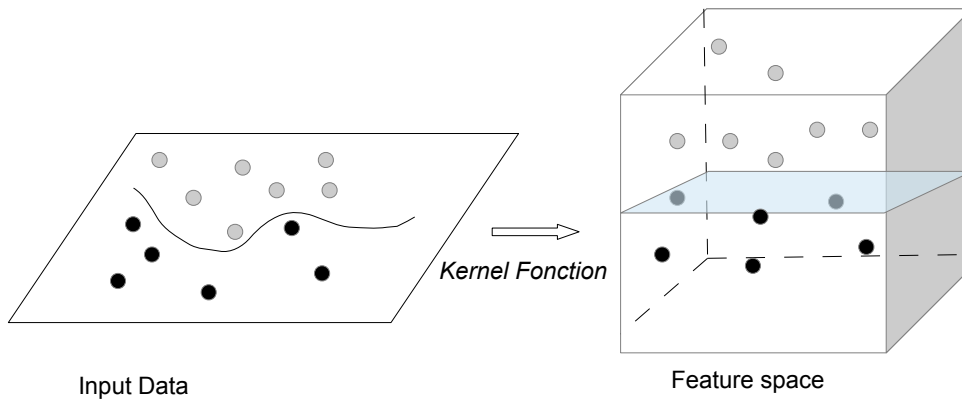


FIGURE 3.12 – Mappage de l'espace \mathbb{R}^2 vers l'espace caractéristique

3.5.6 Méthode basée sur la théorie des *Graphes spectraux*

La clusterisation basée sur les graphes spectraux consiste à identifier, dans un graphe, les ensembles de nœuds fortement connectés pour les regrouper au sein d'un cluster. Cette mise en cluster regroupe les nœuds fortement connectés entre eux et sépare les nœuds faiblement connectés entre eux. Dans l'exemple de la Figure 3.13, les clusters sont définis en fonction de la densité du graphe voisin par rapport à la densité globale du graphe.

Cette méthode fournit une clusterisation très précise, mais la complexité temporelle augmente considérablement lorsque les graphes deviennent plus complexes. L'algorithme CLICK [82], que nous avons comparé à notre Algorithme *similVar* (cf Section 5.3.2 du Chapitre 5), utilise la théorie des graphes spectraux et des techniques statistiques pour classer les données.

3.5.7 Méthode basée sur l'*Affinité de propagation*

Dans la méthode basée sur l'*Affinité de propagation*, tous les points sont considérés comme des centres potentiels du cluster et la valeur négative de la distance euclidienne entre deux points

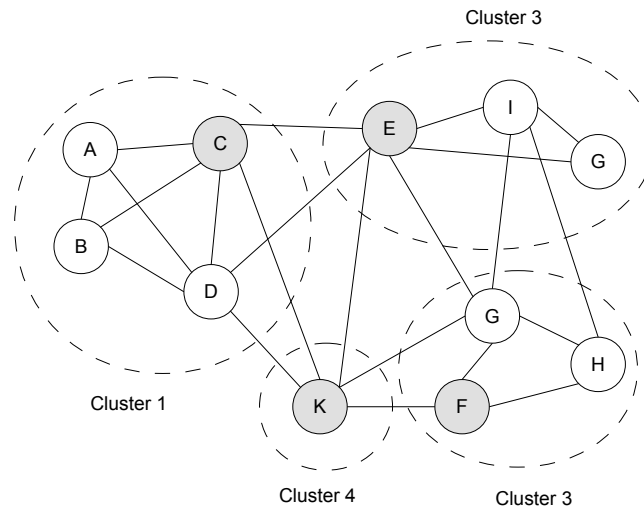


FIGURE 3.13 – Graph Clusterisation

détermine leur affinité. Ainsi, plus la somme des affinités est élevée, plus la probabilité que le point soit le centre du cluster est grande.

L’algorithme que nous avons choisi [30] applique cette stratégie, en maximisant la valeur de la fonction globale de clustering à chaque itération.

3.5.8 Bilan des méthodes de clusterisation

Les méthodes de clusterisation “classiques” que nous avons présentées offrent des implémentations faciles à mettre en œuvre et largement éprouvées. En revanche elles nécessitent la détermination : du nombre de clusters a priori (*Partition, Théorie floue*) ou du niveau hiérarchique des découpages souhaités (*Hiérarchie*). De plus, elles sont sensibles aux valeurs aberrantes et les complexités temporelles sont élevées. Les méthodes de clusterisation “récentes” offrent de meilleures performances en réduisant par exemple les valeurs aberrantes. Les limites se situent au niveau du paramétrage parfois complexe des outils et de la sensibilité des résultats par rapport aux nombres de variables (*Grappe spectral*). Tout comme les méthodes “classiques” la complexité temporelle est parfois élevée (*Affinité de propagation*).

3.6 Bilan du chapitre

Ce chapitre consacré à la présentation du contexte scientifique et de l’état de l’art propre à la modélisation prédictive a permis de décrire trois paradigmes que nous considérons comme

proches de notre projet d'étude : le système multi-agents, le raisonnement à partir de cas et la clusterisation.

Le choix du système multi-agents est motivé en grande partie par l'approche *Bottom-Up* (modéliser individuellement pour expliquer les phénomènes macro) et par la notion d'émergence. Ces deux concepts sont en effets adaptés à des situations non déterministes. Concernant le raisonnement à partir de cas, c'est un outil à base de connaissances conforme à nos attentes car il n'est pas nécessaire d'être exhaustif sur les connaissances du domaine. De plus il s'agit d'un système évolutif qui s'enrichit au fil des expériences. Il s'intègre aussi parfaitement au paradigme multi-agents considéré dans ce contexte comme "générateur" de cas. Enfin les outils de clusterisation que nous avons décrit montrent un intérêt car ils permettent de classifier des tendances analogues.

Ce chapitre décrit aussi certaines limites observées pour ces paradigmes par rapport à notre problématique de prédictivité dans un contexte non déterministe. Nos contributions en réponse à ces dernières sont exposées dans la seconde partie du manuscrit.

Deuxième partie

Nos contributions

La seconde partie de ce manuscrit est consacrée à la description de nos trois contributions. Elle s'appuie sur des contextes scientifiques exprimés dans l'état de l'art : paradigmes SMA, RàPC et méthodes de clusterisation. Nous soulignons à travers les 3 chapitres qui la composent notre positionnement et nos réponses face aux problématiques de prévention et de prédictivité d'EIAS.

Dans cette optique, notre première contribution exprimée dans le Chapitre 4 décrit et explique l'intérêt du couplage SMA/RàPC dans un contexte non déterministe tel que le bloc opératoire. Après avoir posé notre formalisme, nous montrons qu'une telle architecture capable de produire un grand nombre de situations (caractérisées par l'état du système) grâce au SMA, couplée à un raisonnement par analogie grâce au RàPC est une réponse possible pour prévenir les risques. Le Chapitre 5 expose les outils d'aide à la décision implémentés et donc notre seconde contribution destinée à analyser des interactions entre agents fortement impliquées dans l'évolution de l'état de notre système. Notre troisième contribution qui est l'objet du Chapitre 6 est aussi définie par des outils d'aide à la décision dont l'objet consiste à prédire l'évolution de notre système.

Chacun des chapitres est composé : d'une section implémentation, d'une section résultats et d'une section discussion.

Chapitre 4

Couplage SMA/RàPC : une architecture comme réponse à la prévention des risques dans un bloc opératoire

Sommaire

4.1	Introduction	59
4.2	Modélisation SMA : intérêts et motivations	59
4.2.1	Les systèmes complexes	59
4.2.2	Modèle et modélisation	60
4.2.3	Simulation à base d'agents	61
4.2.4	Limite d'une modélisation SMA	62
4.3	RàPC : un apprentissage par analogie	63
4.4	Modélisation SMA : formalisme et description du modèle	64
4.4.1	Modèle SMA pour le bloc opératoire : formalisme	65
4.4.2	Description du Modèle SMA pour le bloc opératoire : protocole ODD	67
4.5	Modèle RàPC pour le bloc opératoire	74
4.5.1	Élaboration	74
4.5.2	Remémoration	75
4.5.3	Réutilisation	76
4.5.4	Révision	77
4.5.5	Mémorisation	77
4.6	Couplage SMA/RàPC	77
4.6.1	Vision globale du couplage SMA/RàPC	77
4.6.2	SMA : une organisation holonique	78
4.6.3	Organisation de la mémoire collective	79
4.7	Synthèse de la présentation de l'architecture	81

4.8	Simulateur pour MASSAI	82
4.8.1	Plateforme de simulation CoupSR_PredRB : un outil de prédiction des risques au bloc	82
4.8.2	Exploitation des données : une approche par analogie	84
4.8.3	Simulateur et émergence	85
4.8.4	Comparaison <i>CoupSR_PredRB</i> avec avec des plateformes simulant la salle d'opération	86
4.9	Bilan du chapitre	88

4.1 Introduction

Notre travail de recherche, dont la finalité consiste à prévenir les risques dans un environnement complexe non déterministe soulève de nombreuses problématiques. Parmi elles, la caractérisation d'une situation en dehors du champ déterministe reste encore un défi. Notre projet, axé sur la détection de situations a priori non définies entre dans ce cadre. Malgré le développement de paradigmes orientées sur la modélisation prédictive, certaines questions restent en suspend. Il s'agit en l'occurrence de la caractérisation d'émergences. Doit-on les considérer comme : de nouveaux point de vue ? De nouvelles perceptions ? Des dérives ou non dérives de situations ? De même, comment peut-on envisager une adaptation voire une capitalisation de ces émergences, pour faire évoluer un contexte clos par définition. Cette clôture est certes une garantie quant à la maîtrise de l'aléa mais restreint l'évolutivité du système.

Face à cette double problématique, notre objectif consiste donc à construire un système capable de produire des situations qualifiées comme inédites et capable de s'enrichir de ces dernières en s'y adaptant. L'architecture fondée sur le couplage des paradigmes SMA et RàPC est une solution que nous proposons. Dans la suite de ce chapitre nous présentons tout d'abord ces deux paradigmes puis leur couplage. Le simulateur construit selon cette architecture est ensuite décrit puis comparé à deux plateformes dont le contexte s'avère être assez similaire. Un bilan vient clôturer ce chapitre.

4.2 Modélisation SMA : intérêts et motivations

Une des premières interrogations repose tout d'abord sur le type d'organisation que nous avons choisi et sur les motivations qui nous ont conduit à ce choix. Sachant que nous souhaitons décrire tout ou parti d'un système et en particulier son évolution, la modélisation est selon nous la plus adaptée des réponses. Notre ambition orientée sur l'émergence et sur la prédiction de l'évolution d'un système nous incite tout d'abord à opter pour une modélisation agents. La présente section décrit ce type de modélisation, ses intérêts et ses limites. Notre proposition en réponse à ces limites est l'objet de la Section 4.3.

4.2.1 Les systèmes complexes

Un système complexe est composé de plusieurs éléments qui interagissent entre eux simultanément dans un environnement borné, pour former une unité globale. Ces systèmes peuvent s'auto-organiser, produire des actions rétroactives, s'auto-créeer ou s'auto détruire. In fine ils génèrent des émergences observables au niveau macroscopique. Selon Aristote [41] : "la totalité est plus que la somme des parties". Si l'on se réfère à cette citation, l'émergence confère au système la capacité d'acquérir de nouvelles propriétés dont l'explication ou la déduction vont au-delà de la seule observation des parties. [10] propose un formalisme résumé ci-dessous.

Les structures $\{S_i\}$ (i appartient à l'ensemble d'indices I) sont des systèmes (considérés comme des agents dans notre cas) qui interagissent entre eux et Obs^1 permet d'observer et d'évaluer les propriétés en jeu de ces structures telles que les interactions notées Int . Ces interactions génèrent alors une nouvelle structure : $S^2 = R(S_i^1, Obs^1, Int^1)$. Il s'agit du résultat des processus d'interactions considéré comme une structure émergente avec pour mécanisme d'observation Obs^2 . Cela conduit à la définition suivante :

P est une propriété émergente ssi $P \in Obs^2(S^2)$, $P \notin Obs^2(S_i^1)$

Cette définition de l'émergence, montre que l'acquisition de nouvelles propriétés qui la caractérise, ne peut s'expliquer par la seule observation des structures. Ainsi par exemple, l'observation de la spécificité d'une cellule vivante ne s'explique pas uniquement par l'observation individuelle de chacune des réactions chimiques. L'émergence soulève donc les questions liées aux niveaux d'observation et aux interactions inhérentes aux systèmes complexes. La modélisation est l'une des réponses apportées à ces interrogations.

4.2.2 Modèle et modélisation

Les nombreuses définitions d'un modèle convergent sur le fait qu'il s'agit d'une représentation simplifiée de la réalité. Cela soulève des interrogations sur la nature du modèle. Est-il une représentation abstraite ou concrète du système ? Cherche t-on à globaliser une représentation ou à la sectoriser ? Cela nous conduit à nous interroger sur ce qui nous motive dans la modélisation.

La première motivation est dictée par la volonté de manipuler avec plus de facilité un concept réel à partir de sa représentation simplifiée. Répondre à des questions ou des problématiques est la seconde motivation. Plusieurs types de modèles sont susceptibles de répondre à ces motivations mais se classent dans des finalités très différentes. Les modèles statiques tels que les diagrammes de classe UML sont destinés à décrire les systèmes. Leur utilisation est fréquente en amont de l'implémentation d'un module logiciel. Les modèles dynamiques sont plutôt réservés à la prédiction souvent associée à l'évolution d'un système. Ils sont utilisés dans la gestion et le contrôle des systèmes et aussi dans des logiques de prédictions préventives. Notre problématique de caractérisation d'une situation émergente entre dans le cadre d'application d'une modélisation dynamique. Cependant, cette notion ne précise pas le déterminisme du modèle. Ce statut dépend exclusivement du type de variables utilisé. Lorsqu'elles ne sont pas aléatoires le modèle est déterministe et dans le cas contraire il est non déterministe. Dans notre contexte, la modélisation est non déterministe et le temps est une donnée non continue mais discrète. Ainsi, une variable u ne varie pas de manière continue en fonction du temps mais en fonction d'un ensemble de valeurs discrètes. Celles-ci décrites par l'ensemble $c = \{c_0, c_1, \dots, c_n\}$ sont telles que $c_{j+1} - c_j = C$, où C désigne un intervalle de temps constant. Les changements d'état des variables interviennent à intervalle de temps C . La Figure 4.1 reprend chacune des briques que

nous avons considérées comme essentielles en réponse à la problématique : “Caractérisation d’une situation en dehors du champ déterministe”.

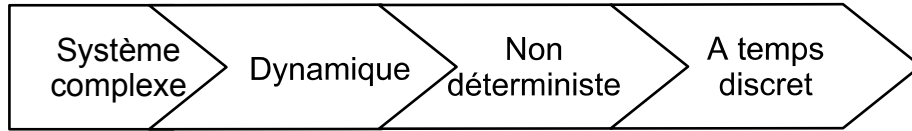


FIGURE 4.1 – Caractérisation du modèle

Le choix du modèle conceptuel, que nous venons de décrire et de résumer dans la Figure 4.1 est l’étape qui précède le choix du type de simulation. A ce stade nous devons envisager un simulateur qui permet d’exécuter un modèle avec : des transitions entre un état final et un état initial, des événements, des interactions et des collaborations. La simulation orientée agent qui répond à toutes ses exigences est l’objet du paragraphe suivant.

4.2.3 Simulation à base d’agents

Nous avons évoqué dans la Section 3.3 du Chapitre 3 les modes de simulation et distingué la modélisation ascendante, descendante, continue et discrète. Dans notre contexte de simulation les différentes entités intervenants dans le bloc opératoire sont hétérogènes. Elles interagissent entre elles et influencent le niveau de risque de survenue d’un événement indésirable grave. Toutes ces caractéristiques s’intègrent dans un mode de simulation centré objet où les entités interagissent entre elles. De plus, notre observation est portée sur l’évolution du comportement des entités et du système dans des pas temps donnés. La modélisation multi-agents qui intègre l’approche ascendante, la simulation à temps discret et une simulation centrée sur des entités indépendantes nommées agents est selon nous un mode de simulation adapté. A titre de contre exemple, la modélisation mathématique basée sur les équations différentielles et soumise à une approche descendante n’est pas adaptée à notre contexte de simulation.

Fondamentalement, l’agent est une entité en capacité d’agir dans un environnement donné. Il communique et interagit avec les autres agents et perçoit tout ou partie de son environnement. Ses comportements résultent de la programmation de modules imputées à des fonctions précises qui s’exécutent de manière synchrones ou asynchrones.

Les choix d’implémentation ont une répercussion sur l’évolution de l’état du système, l’accessibilité à son état, la dynamique (statique lorsque l’action de l’agent est nécessaire au changement ou dynamique dans le cas inverse) et la continuité (discret lorsque les actions appartiennent à un ensemble fini ou continu à contrario).

Dans notre environnement, ce sont les agents qui influencent l'évolution de l'état du système, et l'ensemble des actions sont définies dans la structure du modèle. L'environnement qui coïncide avec notre modèle intègre donc une dynamique statique et une continuité discrète. Son accessibilité est partielle ou totale selon les rôles, les buts et les actions attribués à chaque agent.

La singularité de la simulation multi-agents réside dans la capacité des agents à interagir. Qu'elle soit uni-destinataire ou pluri-destinataires, la communication propre à certaines interactions impose un mode de transmission, un langage et un protocole. Chaque agent peut alors émettre ou recevoir des messages volontairement ou involontairement, directement ou indirectement. Lorsqu'elle est directe la communication impose la connaissance de l'identité de l'émetteur et du récepteur. Dans le cas contraire, la communication est dépendante de l'environnement et de ses variations. La modification de l'environnement est interprétable comme un message borné à une topologie impliquant une classe d'agents. Ferber propose une synthèse des différents éléments qui caractérisent un SMA [26] :

- Un environnement E caractérisé par l'ensemble de ses états notés S et comme tout espace par une métrique.
- Un ensemble d'objets O localisables dans E .
- Un ensemble d'agents A tel que $A \subseteq O$.
- Un système I d'interactions et de communications : envoi de messages, de signaux ...
- Une organisation qui structure A et définit les fonctions de chacun des agents : rôle, perception, transformation et modification des objets O .
- Une éventuelle relation à des utilisateurs U agissant sur le SMA via des agents interfaces tel que $U \subseteq A$.

Les principes fondamentaux des SMA que nous venons d'exposer confortent nos motivations pour ce choix de paradigme évoquées en début de section. Ils argumentent en faveur de leur utilisation pour notre type de modélisation que nous souhaitons, non déterministe, à intervalle de temps discret et en capacité d'interactions. Le paradigme SMA que nous avons choisi offre une approche "individus centrés" autorisant des équations stochastiques individuelles alors que cette méthode n'est possible qu'au niveau macro dans la plupart des modèles mathématiques. Les techniques de micro-simulation sont certes venues palier cette lacune [34] mais n'intègrent pas à l'origine la notion fondamentale d'interactions. Le paradigme multi-agents qui englobe à la fois la modélisation individuelle et les interactions est donc conforme à nos attentes. Cependant, l'aspect prédictif au sens prévision de notre modélisation impose une structuration spécifique du module d'apprentissage qui fait défaut au SMA. Le paragraphe qui suit décrit cette limite.

4.2.4 Limite d'une modélisation SMA

L'apprentissage d'un système a pour vocation l'amélioration d'une ou de plusieurs pratiques lorsqu'une adaptation est possible. Dans un système multi-agents, un agent est en capacité d'ap-

prentissage induisant la possibilité d’analyser son environnement pour s’adapter à une situation et réagir en conséquence afin d’atteindre des objectifs et/ou des buts selon sa typologie. Dans son article [51] Marc-Olivier Labarre pose les bases théoriques de l’apprentissage d’un SMA qu’il classe en deux catégories : les modèles de décision de Markov et les jeux stochastiques [13]. Un problème de décision de Markov (MDP) produit des résultats aléatoires lorsque l’agent agit. Il est formalisé par un tuple (S, A, R, T) où S représente l’ensemble des états, A l’ensemble des actions, T la matrice $S \times A \times S$ des probabilités de transitions et R la matrice $S \times A$ des bénéfices. Sur un plan intuitif, ce processus est comparable à un automate probabiliste. L’objectif consiste à rechercher le meilleur état possible, *bénéfique*, par évitement d’états *néfastes*. L’efficacité est à rapprocher des gains et pénalités acquis au long de l’expérience. Nous n’évoquerons pas les jeux stochastiques qui est certes un sujet de recherche très riche et très approfondi, mais qui dépasse le cadre de cette thèse. Ces méthodes d’apprentissage sont révélatrices d’une limite majeure si l’on envisage d’inclure la prédictivité. En effet, les mécanismes nécessaires sont plus complexes car ils doivent intégrer les notions de hiérarchisation et de classification. Celles-ci sont indispensables à l’agent pour agir et décider avec toute l’objectivité attendue. Nous avons expliqué que la solution SMA a été choisie car elle répond à nos attentes de modélisation d’un contexte non déterministe tel que le bloc opératoire. Nous n’avons donc pas recherché une autre solution mais plutôt une méthode que nous pourrions agréger à notre SMA afin d’enrichir son processus d’apprentissage. Ce choix est l’objet de la section qui suit.

4.3 RàPC : un apprentissage par analogie

On ne peut dissocier l’apprentissage d’un système d’acquisition de connaissances. En la matière, les Systèmes de Gestion de Bases de Donnée (SGBD) sont des candidats naturels. Ils permettent de gérer une base de données composée d’une collection de données en relation. Celles-ci sont alors partagées par plusieurs applications, stockées avec un minimum de redondance et indépendantes des applications. Dans un mode de fonctionnement classique, les dépendances fonctionnelles sont des propriétés conçues entre autres pour éviter les redondances. Une dépendance fonctionnelle peut se formaliser de la manière suivante : soient R un schéma de relations, $X = \{x_1, x_2, \dots, x_n\}$ et $Y = \{y_1, y_2, \dots, y_n\}$ des sous-ensembles de R . $X \rightarrow Y$ ssi $\forall r \in R, \forall$ la paire de n-uplets t et $s \in r, t[X] = s[X] \Rightarrow t[Y] = s[Y]$ ($t[X]$ représente la projection du n-uplets t sur les attributs X). Cette propriété qui stipule qu’une valeur de X détermine au plus une valeur de Y garantit une certaine intégrité à la base de données mais écarte la possibilité d’une approche de fouille de données par similitude. Or, les notions d’émergence et de prédictivité imposent aux bases de connaissances un relâchement de ces contraintes.

La méthode d’apprentissage profond RNA, que nous avons décrit dans la Section 3.4.1 du Chapitre 3 est une piste intéressante mais ne montre son efficacité que sur un grand nombre de données. En effet, un réseau de neurones n’est en mesure de prédire que s’il a été préalablement entraîné. Notre contexte d’étude étant très limité en nombres de cas réels exploitables, nous

avons aussi écarté cette solution mais conservé l'idée d'un apprentissage au fil de l'eau. Notre choix s'est donc orienté sur une base de connaissances robuste, évolutive et suffisamment souple dans ses instances.

Le raisonnement à partir de cas répond à ces critères car il permet :

- d'enrichir une base de connaissances à partir d'expériences passées,
- de résoudre des problèmes par analogies en utilisant des problèmes déjà résolus,
- d'adapter des solutions aux spécificités spatio-temporelles.

Le couplage du RàPC avec notre modélisation SMA permet de relier deux paradigmes qui interagissent tout en fonctionnant de manière indépendante. La démarche est alors orientée dans un processus de réponse à un problème global qui implique des traitements différents et une coopération entre les systèmes impliqués. Les échanges sont bidirectionnels avec une mémorisation de l'évolution de l'état du système. Chacun des systèmes peut subir l'influence de l'autre de manière synchrone ou asynchrone. Une vision résumée et formelle de notre couplage est donnée dans la figure 4.2. Dans cette architecture, la couche *langage* est la partie programmable de l'agent qui intègre des appels à la *couche interactions*. Cette dernière gère les communications entre agents sous couvert de protocoles. Les événements émergent de la *couche interactions* et sont gérés par la *couche langage*. La *couche apprentissage* est couplée avec le RàPC. Les comportements, les prises de décision de l'agent en dépendent et sont contrôlées par la *couche langage*. La *couche simulation* qui exécute le code est dépendante des trois autres couches.

Après avoir motivé notre choix d'architecture, nous présentons dans la suite de ce chapitre notre implémentation.

4.4 Modélisation SMA : formalisme et description du modèle

En préambule à cette section, nous posons la terminologie relative à la modélisation du bloc opératoire. Les variables d'état sont nommées attributs ; elles définissent le comportement de chaque agent indispensable à l'atteinte des objectifs de simulation. Leurs valeurs sont modifiées à chaque pas de temps de la simulation appelé cycle. Celles-ci confèrent au système un état (état global du système) : *normal*, *alert* ou *failure*. En ce qui concerne l'acquisition des données, nous distinguons deux modes au sein de notre architecture : le mode non connecté et le mode connecté également appelé "mode temps réel". Dans le mode non connecté, les données résultent d'une fonction ou de données statistiques et sont affectées à chaque cycle. Dans le mode connecté, les données sont issues d'une capture dynamique : les capteurs connectés (bracelet pour la fatigue, capteurs électrochimiques pour le taux d'infection...) surveillent le dispositif. Quel que soit le mode considéré, ces sources d'information sont considérées comme des ressources "à disposition" des agents. Ainsi un agent peut solliciter suivant ses droits une ou plusieurs instances d'une ressource (par exemple le niveau sonore d'un capteur). Dans cette thèse, nous ne traitons que le mode non connecté qui correspond à celui actuellement mis en œuvre. Cette terminologie étant posé, nous abordons à présent l'implémentation de notre archi-

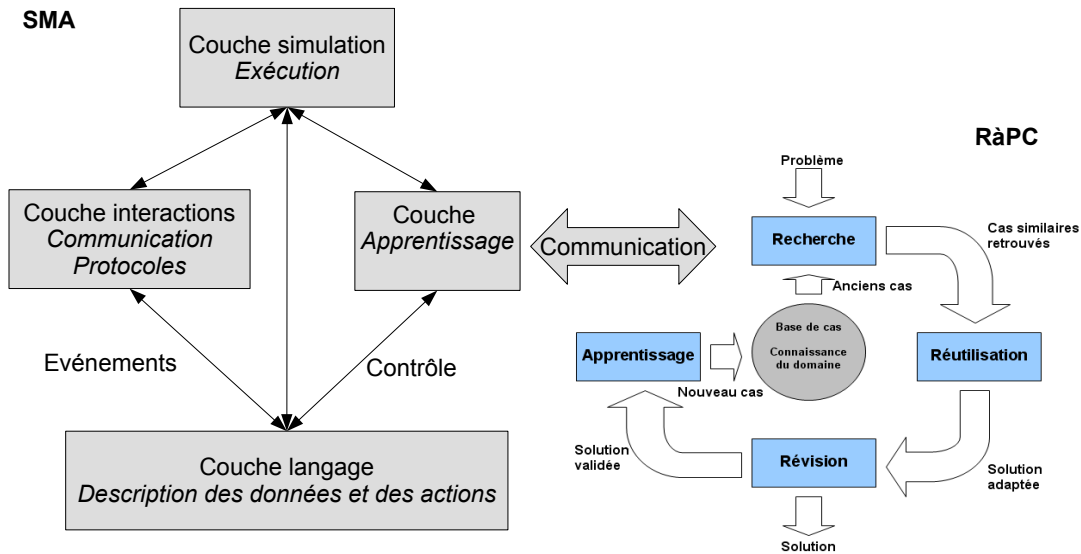


FIGURE 4.2 – Structure du couplage SMA/RàPC

tecture.

4.4.1 Modèle SMA pour le bloc opératoire : formalisme

Nous avons décrit dans la première partie, le contexte scientifique de notre thèse et présenté les paradigmes SMA et RàPC dans un cadre général. Nous exposons à présent notre modélisation SMA utilisée dans le contexte du bloc opératoire.

Pour rappel, un agent perçoit son environnement grâce à des capteurs internes et agit sur celui-ci à partir d’effecteurs. Chaque agent agit indépendamment et l’état du système résulte de la sommation de l’ensemble des actions individuelles. L’état du système E est donc un ensemble de partitions d’états individuels E_i avec $E = E_0 \cup E_1 \cup \dots \cup E_n$. Chaque agent est autonome et connaît son état e_i mais ignore celui des autres agents ainsi que leur position. Il n’a donc pas connaissance de l’état global du système. Dans la Figure 4.3 l’agent est défini comme une entité qui perçoit une fraction env_k de l’environnement (incluant aussi les autres agents) sur lequel il agit après décision. Ainsi, un agent est en capacité d’évaluer l’évolution locale de l’état du système.

Cet état est aussi pour partie dépendant des interactions entre agents classées en interactions “embarquées” ou “non embarquées”. Une interaction “embarquée” est une capacité endogène

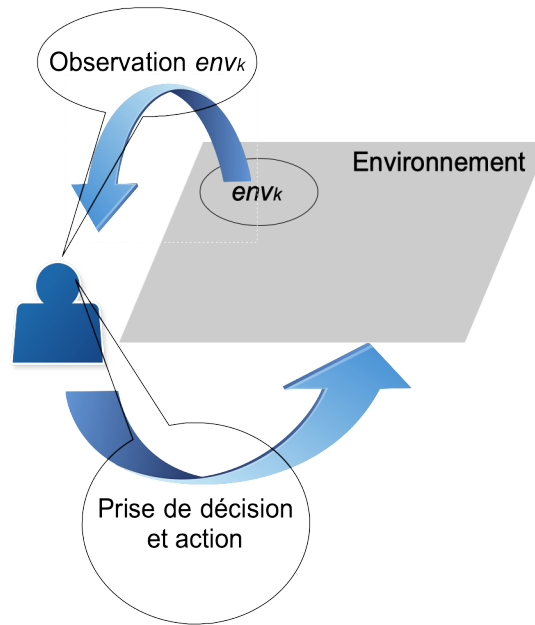


FIGURE 4.3 – Définition d’un agent

d’un agent a_i qui lui permet d’interagir sur un agent a_j . Ainsi, l’état individuel E_j d’un agent a_j peut s’exprimer par la fonction : $I \times E \rightarrow \mathbb{R}$. L’état E_j de l’agent a_j est donc dépendant des interactions prédéfinies $I = \cup_{0 \leq k \leq n} (I_k)$ avec les autres agents et de l’état global E du système. Ce type d’interactions est implémenté dans un contexte non-connecté (données issues de simulations cf 4.4). La détermination d’interactions “non embarquées” résulte quant à elle d’une approche déductive. A partir de l’émergence telle que nous l’avons définie dans la Section 3.3.3 du Chapitre 3, nous déterminons des causalités/similarités d’un état global E . La fonction :

$$E \times \prod_{1 \leq k \leq n} C_k \rightarrow \prod_{1 \leq k \leq n} I_k$$

détermine un ensemble d’interactions I_k entre un agent a_k et d’autres agents a_i ($i \neq k$). Ces interactions dépendent de l’état du système et d’un ensemble de causalités/similarités $\cup(C_k)$ avec $1 \leq k \leq n$. La construction de l’ensemble $\cup(C_k)$ est l’objet du Chapitre 5. Nous y présentons l’une de nos contributions dont le principe consiste à classifier (clusteriser) les données comme réponse possible à la détermination d’interactions I_k . Les interactions “non embarquées” prennent tout leur sens dans un mode connecté (données issues de capteurs cf Section 4.4).

Le formalisme présenté dans ce paragraphe pointe le regard sur les interactions entre agents. Nous pensons qu'elles sont des facteurs d'influence majeurs sur l'évolution de l'état du système et sur la notion d'émergence (cf Section 3.3.3 du Chapitre 3). Celles-ci sont conceptuellement intégrées à notre modélisation que nous décrivons dans la section qui suit, selon le protocole ODD (Overview, Design concepts, Details) [37].

4.4.2 Description du Modèle SMA pour le bloc opératoire : protocole ODD

ODD est le protocole que nous avons choisi pour décrire notre modèle. Celui-ci offre en effet l'avantage d'une approche plus formelle que d'autres outils descriptifs tels que l'UML ou l'AML proposé par [17]. Ces derniers sont cependant très utiles pour les vues globales sous forme de diagramme. Dans la suite de cette section, nous décrivons notre SMA selon : une vue générale du modèle (Overview), les éléments de conception du modèle (Design concepts) et les détails de fonctionnement du modèle (Details). Nous retrouvons à travers cette présentation ODD les organisations "Congrégations", "Sociétés" et "Hiérarchies" présentées dans la Section 3.3.3 du Chapitre 3.

4.4.2.1 Vue globale : Overview

La figure 4.4 nous donne une vue de la structure globale du modèle. Nous avons choisi un diagramme de type AML plutôt qu'UML, en raison de son orientation SMA. En effet, ce langage étend l'UML grâce à l'intégration de concept tel que AGR (Agent Groupe Rôle) [28]. Nous proposons à présent de décrire cette vision générale en la déclinant selon les trois focales proposées par Overview : Objectifs, Entités et variables d'état, Processus et ordonnancement.

Objectifs

La modélisation du bloc opératoire a pour objectif la détection d'EIAS en fonction des activités de chirurgie en salle d'opération. Les activités englobent : les actes humains, l'activité contaminante et l'activité matérielle. Il s'agit de produire un grand nombre de situations dont certaines peuvent conduire à des EIAS.

Entités et variables d'état

La salle d'opération est conçue comme un système multi-agents où chaque entité est représentée par un agent. Chacun d'eux appartient à un groupe que l'on nomme espèce ou species (par exemple, l'agent *surgeon* (chirurgien) appartient à l'espèce *Personnel*) selon les organisations "Congrégations" et "Sociétés". Notons qu'une espèce peut n'être composée que d'un seul agent (agent *alert* de l'espèce *Supervisor*). Ce type de modèle permet de créer différents scénarios et de simuler les évolutions possibles de l'état du système. Nous rappelons que les agents agissent en fonction de leurs propres objectifs et comportements, et interagissent avec les autres agents

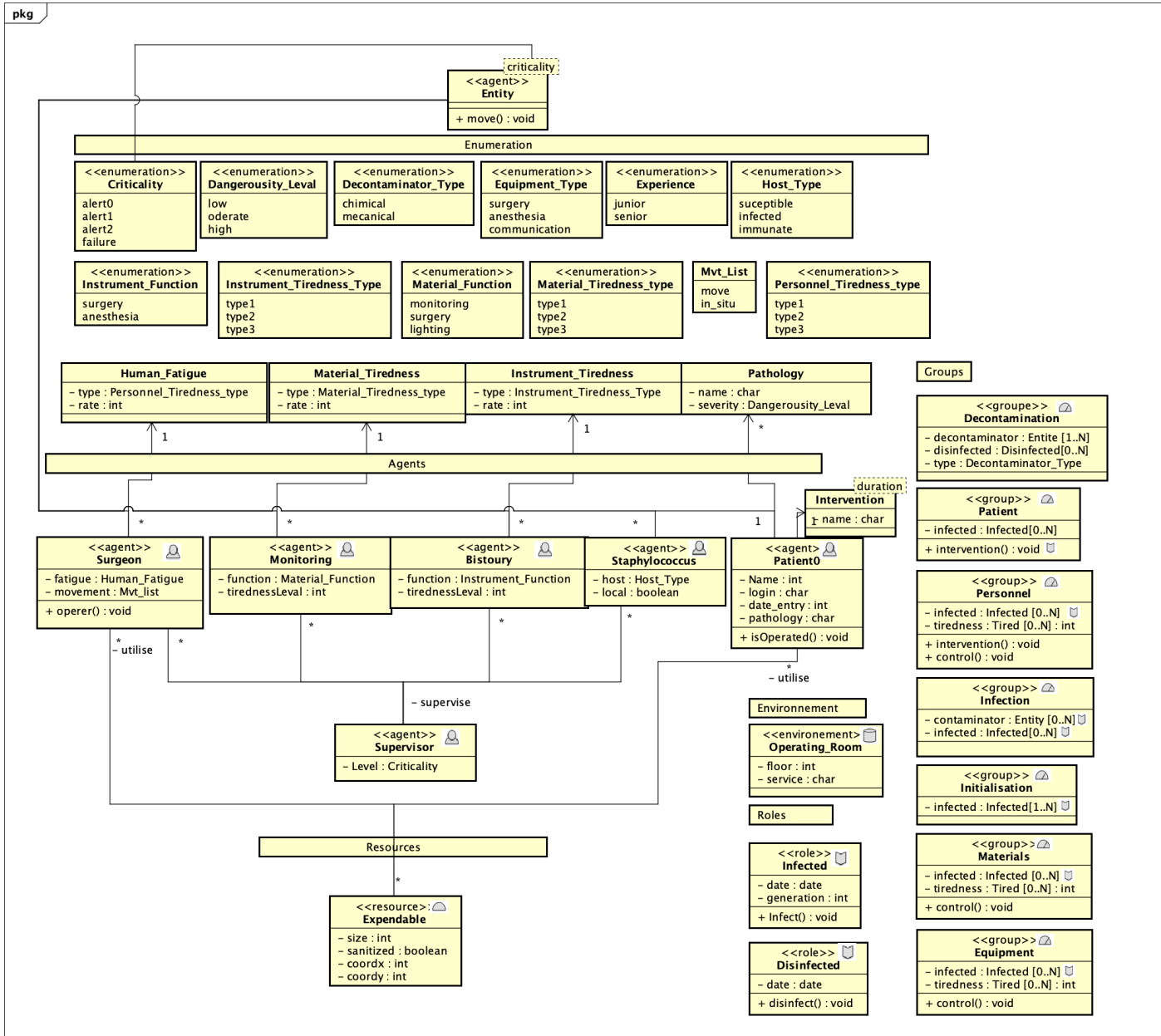


FIGURE 4.4 – Vision globale de la structure du modèle

ainsi qu’avec leur environnement. Notre modèle utilise des agents de type BDI (cf Section 3.3.2 du Chapitre 3 [85]). Ces agents appartiennent à cinq espèces : Personnel, Material, Infection, Patient, Alert décrites dans la Section 4.4.2.4.

Processus et ordonnancement

Dans chaque intervalle de temps (tous les 5 cycles de 30 s chacun), les agents appartenant aux différentes espèces (*Personnel*, *Material*, *Infectious*, *Patient* et *Supervisor*) agissent

selon leurs croyances, intentions et désirs. Les agents *Personnel* soignent un agent *patient_i* ($0 \leq i \leq n$), les agents *Material* surveillent l'intervention chirurgicale, les agents *Infectious* contaminent l'environnement et l'agent *Supervisor* détecte les dépassements de seuils. Comme le montre la Figure 4.5, les dépassements de seuils produisent des alertes mais n'interrompent pas l'activité en salle d'opération.

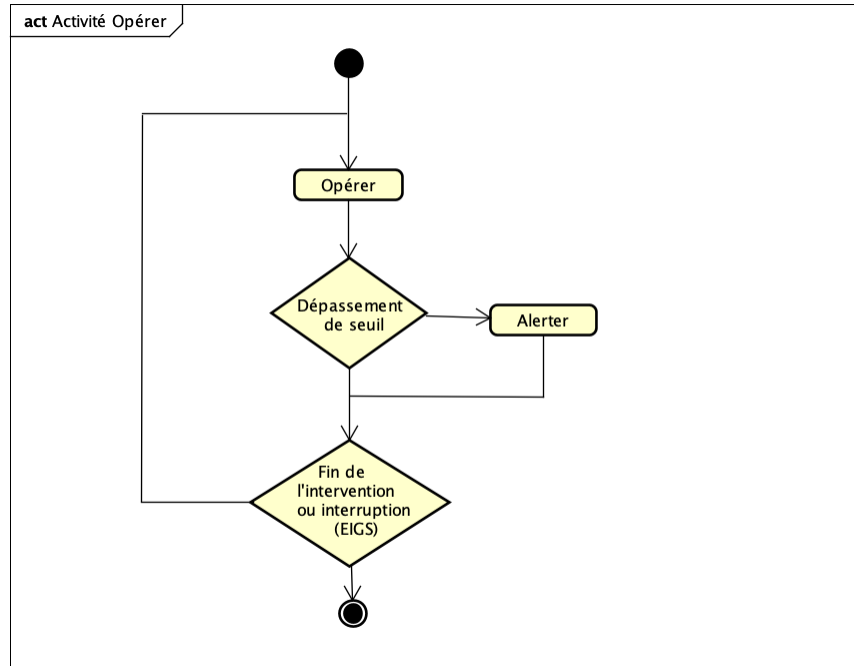


FIGURE 4.5 – Diagramme simplifiée de l'activité "Opérer"

4.4.2.2 Éléments de conception : Design concepts

Les éléments de conception informent sur les concepts de la simulation multi-agents. Parmi l'ensemble des concepts proposés par [37], nous explicitons : les principes, la stochasticité, l'émergence, les interactions entre agents et enfin l'observation.

Principe et stochasticité

Dans notre contexte non connecté, les variables d'état évoluent de manière aléatoire hormis la fatigue dont la croissance est continue (cf 4.4.2.4). Les déplacements des agents mobiles (*Personnel* et *Infectious*) sont aussi aléatoires mais sont calibrés en terme de célérité et bornés dans l'espace. Ainsi par exemple, un anesthésiste (espèce *Personnel*) a moins de mobilité qu'un infirmier de bloc opératoire (espèce *Personnel*).

Émergence

Les flux de contamination sont superposables aux déplacements des agents *Personnel* et cer-

taines alertes sont précédées par l'apparition de trajectoires imputables à des déplacements identiques répétés.

Interactions

Dans un contexte non-connecté (mode de simulation actuel) toutes les interactions sont paramétrables. L'attribut *noise* de l'agent *phone* (espèce *Material*) peut par exemple interagir ou non avec l'attribut *fatigue* d'un agent *personnel*. Lorsque le mode est connecté, les interactions ne sont pas paramétrées mais résultent de la réalité.

Affichage

L'affichage du simulateur est divisé en 5 fenêtres : une fenêtre de paramétrage, une vue grossie des agents *Infectious*, une vue de la salle d'opération et des praticiens (*Personnel*), les graphiques de chaque attribut surveillé et enfin l'affichage des alertes (agent *Supervisor*). Lors des simulations, l'utilisateur voit évoluer les agents dans leur vue respective ainsi que les variations de leurs attributs. Les alertes sont spécifiques aux attributs observés. Ces vues sont présentées dans la Section 4.8.1.

4.4.2.3 Détails

Nous abordons à présent les détails techniques. Ils sont scindés en trois parties dans notre contexte : initialisation, données d'entrées et exécution. Nous les présentons dans les trois paragraphes qui suivent.

Initialisation

Chacune des vues (salle d'opération, agents *infectious*, graphiques, agent *Supervisor*) est initialisée en même temps et indépendamment. Les praticiens (agent *Personnel*) sont disposés aléatoirement autour de la salle d'opération où se trouve l'agent *Patient*. Les particules contaminées et non contaminées (agents *Infectious*) sont réparties aléatoirement dans leur volume 3D. Les graphiques sont initialisés à partir des valeurs paramétrées, et enfin, l'agent *Supervisor* est en état "normal".

Données d'entrée

Les données d'entrée intègrent : le type et le nombre d'intervenants au bloc opératoire, la criticité de l'intervention, le type de fatigue humaine (*anxious, no anxious*), le type d'évolution de chaque attribut (*ascending, descending, random*) et enfin le type et le nombre de particules.

Exécution

Chaque simulation est exécutée dans des intervalles de temps réguliers allant de t_i à t_{i+1} $i \in$

$\llbracket 1; n \rrbracket$ appelés cycles et pouvant être paramétrés (par exemple 60s). Les variables d'état nommées attributs définissent le comportement de chaque agent indispensable à l'atteinte des objectifs de simulation. Leurs valeurs sont modifiées à chaque cycle de la simulation. Elles nous donnent l'état du système qui correspond ainsi à la valeur du ou des attributs de chaque agent (la température du patient par exemple) à un moment donné (à un cycle donné). Toutes ces valeurs confèrent au système un état (état global du système) : *normal*, *alert* ou *failure*. L'état *normal* précède l'état *alert_i*, qui est succédé par l'état *failure*. Il existe trois niveaux d'alerte croissants (*i* allant de 1 à 3) dont l'intensité est inversement proportionnelle au temps qui sépare une alerte d'un incident.

4.4.2.4 Description des agents BDI

Le Tableau 4.1 donne une description des variables d'état de nos agents BDI. La modélisation des espèces d'agents et de leurs variables d'état qui figurent dans ce tableau est l'objet des cinq paragraphes qui suivent. Une attention particulière est portée sur l'agent *alert* traité dans le cinquième paragraphe. En effet, son rôle est majeur dans l'organisation fonctionnelle du système.

Modélisation de l'espèce *Personnel*

Les agents de l'espèce *Personnel* ont pour principal objectif (*intention*) le soin du patient dans un contexte aussi sécuritaire que possible. Ils utilisent des ressources (*desire*) et prennent des décisions à partir de mesures disponibles (*belief*). Ils peuvent se déplacer (*movement* de type *move*) ou avoir une gestuelle sans déplacement (*movement* de type *in_situ*). Dans les deux cas, une quantité est attribuée au mouvement. Leur expérience (*junior* ou *senior*) est prise en compte lors des simulations.

La fatigue (*tiredness*) peut être modélisée à l'aide de plusieurs méthodes d'acquisition de données. Il peut s'agir de capteurs dans un mode connecté ou de fonctions dans un mode non connecté. Dans notre cas (mode non connecté), nous avons choisi de définir la croissance de la fatigue par une fonction exponentielle (configurable par l'utilisateur) car il s'agit d'un phénomène continu mais non linéaire. Elle est donnée par la relation suivante : $f(t) = \delta a e^{(k \times t)}$ où : *a* est la valeur initiale, δ la quantité de mouvement cumulée (initialisée à 1), *k* la constante de croissance et *t* le temps. Il convient de noter que chaque personne peut avoir son propre seuil de fatigue limite. Enfin ils peuvent être contaminés (*infected*) par un agent de l'espèce *Infectious*.

Modélisation de l'espèce *Material*

Chaque matériel possède une fonctionnalité (*function*) et risque une éventuelle contamination (*infected*) par les agents de l'espèce *Infectious*. Deux types d'évolution d'efficacité d'un matériel sont à considérer. La première suit une courbe de décroissance en général non linéaire (appareils alimentés par des batteries) et la seconde est caractérisée par une rupture brutale (par

Espèce	Attributs	Commentaires
<i>Personnel</i>	<i>intention</i> <i>desire</i> <i>belief</i> <i>fatigue</i> <i>movement</i> <i>qty_mvt</i> <i>infected</i> <i>experience</i>	opérer un patient dans des conditions de sécurité optimales utiliser les ressources humaines et matérielles (personal, material) mesures utiles à la prise de décision (monitoring, seuils d’alerte) taux de fatigue (échelle allant de 1 légèrement fatigué à 5 épuisé) type de mouvement (déplacement : <i>move</i> ou sur place <i>in_situ</i>) quantité de mouvement (mesure des distances parcourues, ou de la gestuelle (quantité de gestes)) booléen junior, senior
<i>Material</i>	<i>function</i> <i>mat_tiredness</i> <i>noise_mat</i> <i>infected</i>	fonctionnalité du matériel efficacité matérielle (échelle allant de 1 efficace à 5 inefficace) niveau sonore émis par un matériel booléen
<i>Infectious</i>	<i>type</i> <i>local</i> <i>intention</i> <i>desire</i> <i>belief</i> <i>cont_dist</i>	type d’agent infectieux (contaminant) a un impact sur une zone, sur la salle d’opération ou sur les deux contaminer utiliser les vecteurs de contamination (air, eau, surface ..) température, pression de l’air distance minimum de contamination
<i>Patient</i>	<i>state</i> <i>surgery_type</i> <i>temperature</i> <i>FeCO2</i> <i>SpO2</i> <i>FC</i> <i>PNI</i> <i>infected</i>	état de santé urgent, non urgent, complexe, non complexe température corporelle du patient mesure de la capnie (CO ₂ expiré) mesure de la saturation en O ₂ (quantité d’hémoglobine oxygénée dans le sang) mesure de la fréquence cardiaque mesure de la pression artérielle booléen
<i>Supervisor</i>	<i>intention</i> <i>desire</i> <i>belief</i> <i>level</i>	éviter les <i>EIAS</i> alerte préventive (avant l’apparition d’ <i>EIAS</i>) écouter et suivre l’évolution des données qui influencent l’intervention chirurgicale seuils d’alerte

Tableau 4.1 – Variables d’état des agents simulés

exemple une lampe d’éclairage). Cela implique donc une distinction des types de matériels que l’on peut classer en sous-espèces si nécessaire. De manière générale, l’ensemble des appareils possède ses propres outils de diagnostic et d’alerte assimilables à des boîtes noires. N’ayant donc pas accès à ces données, nous avons opté dans le cadre de notre étude pour une évolution aléatoire et décroissante, ou une rupture aléatoire brutale de l’efficacité. L’intégration de capteurs dans le bloc opératoire est une solution pour pallier la boîte noire. On retrouve aussi

parmi nos choix d'attributs *noise_mat* car les perturbations liées aux émissions sonores sont des facteurs de risques non négligeables.

Modélisation de l'espèce *Infectious*

Les agents infectieux sont classés selon leur type (*type*) et selon leur localisation (*local*). Ils contaminent (*intention*) l'environnement (agents et ressources) avec l'appui de différents vecteurs (*desire*), en considérant certaines caractéristiques ambiantes (*belief*). La force de la contamination repose sur la distance minimum (*cont_dist*) entre l'agent contaminant et l'hôte. La croissance du nombre de particules infectieuses est liée à la présence d'une source de contamination (par exemple le patient) qui augmente ou à l'activité humaine (mouvements dans la salle d'opération). Leur diminution dépend principalement du traitement de l'air (régulation du débit d'air, surpression dans la salle d'opération). Comme il n'existe pas de fonctions monotones types décrivant l'évolution de leur nombre, elles peuvent être définies : de manière aléatoire, sur la base de données statistiques ou à partir de capteurs. Cela dit, les agents infectieux n'entrent pas dans le champ des résultats présentés dans ce document car leur impact (selon l'équipe médicale) est trop marginal.

Modélisation de l'espèce *Patient*

Chaque patient est initialement caractérisé par un état de santé (*state*) ainsi que par un score de gravité de l'intervention chirurgicale (*surgery_type*). Au même titre que la fatigue, les constantes du patient utiles à l'évaluation de son état de santé peuvent être extraites de données statistiques ou être le résultat d'une fonction. Cependant, il n'existe pas de fonction monotone adaptée à ce type de variation, et l'ensemble des données disponibles sur les patients sont trop limitées. Pour cette raison, les données évoluent aléatoirement selon un pas défini par les experts pour chacun des attributs (par exemple + ou - 0,2 degré pour la température à chaque cycle). Nous rappelons que toutes ces données peuvent aussi être issues de capteurs dans un mode connecté (non mis en œuvre pour le moment). Enfin, comme tout autre agent il peut aussi être contaminé (*infected*).

Modélisation de l'espèce *Supervisor* : agent *alert*

Chaque agent évolue selon le paradigme BDI que nous avons préalablement défini et qui détermine les règles et les comportements de chacun. Parmi eux se trouve un agent déterminant en ce sens qu'il joue à la fois le rôle de superviseur dans le SMA et le rôle d'interface avec le RàPC. Ce choix d'agent "central" est motivé par notre volonté de simplifier les procédures de décisions et la liaison avec le RàPC. Il supervise le système, l'adapte et propose des solutions afin d'éviter les EIAS (*intention*) selon une organisation "Hiérarchique".

Il supervise le système car il gère les niveaux d'alerte en recherchant des analogies avec des expériences passées extraites d'une base de cas (*desire*). Cela revient à écouter l'évolution des valeurs de chaque attribut (*belief*) sensible aux alertes (par exemple la fatigue humaine), pour les comparer à des seuils (*level*). Il mesure aussi les interactions entre chaque agent (ou entre

un agent et une ressource) en fonction de l'analyse de ces dernières (cf Section 1 du Chapitre 5). Cela permet de pondérer en retour l'évolution des valeurs des attributs des agents concernés. Ainsi par exemple, une augmentation de l'intensité du bruit de la ressource *ventilation* entraîne une accélération de la fatigue d'un agent *chirurgien* étant donné la forte interaction entre les deux.

Il adapte le système en modifiant son statut (exemple : *normal* \rightarrow *alert₁*). Certains statuts comme par exemple l'*alert_i* ($i \in \llbracket 1, 3 \rrbracket$) entraînent une action d'information. Cela répond à l'un des buts majeurs de cet agent qui consiste à prévenir la survenue d'EIAS.

Il propose aussi des solutions à partir d'expériences similaires grâce au raisonnement à partir de cas. Ces préconisations sont toujours soumises à validation par les experts avant toute action corrective du système. Ainsi, par exemple, une alerte qui fait suite à la détection d'un niveau sonore de la ressource *ventilation* considéré comme trop élevé nécessite l'accord de l'expert pour toute modification de la valeur de son intensité préconisée par le système. Le changement de l'état du système en conséquence dépend donc de cet accord.

Nous avons notifié en introduction de ce chapitre l'importance de l'acquisition de connaissances dans notre système. Il s'agit plus précisément de choisir un paradigme qui intègre d'une part l'enrichissement de sa base de données à partir d'expériences, d'autre part la recherche de la solution d'un problème par analogie avec d'autres problèmes déjà résolus. L'implémentation du raisonnement à partir de cas adaptée à ce type d'apprentissage est présentée dans la section qui suit.

4.5 Modèle RàPC pour le bloc opératoire

Les principes du raisonnement à partir de cas ont été expliqués dans le Chapitre 3. Nous rappelons que ce paradigme consiste principalement à réutiliser une (des) solution(s) (solution source) d'un ou de plusieurs précédents problèmes (problème source) dans le but d'en résoudre un nouveau (problème cible) [75]. Par convention, un cas source est écrit (*source*, *Sol(source)*) et un cas cible est écrit (*cible*, *Sol(cible)*). Ainsi *source* et *cible* désignent un problème et *Sol(source)*, *Sol(cible)* leur(s) solution(s). Nous décrivons dans les cinq paragraphes qui suivent l'implémentation choisie pour les cinq phases du RàPC : Élaboration, Remémoration, Réutilisation, Révision, Mémoire.

4.5.1 Élaboration

Un cas est représenté par un couple qui a un identifiant unique et est décrit par deux composantes (*Problème*, *Solution*). Transposé dans notre système, il correspond au couple ordonné (U, R) où U caractérise une séquence de quadruplets (E, A, V, c) et R le couple (S, P).

U est composé de l'entité (agent) E , où chaque attribut A possède la valeur V au cycle c . La première composante de R donne le statut S de l'attribut A et la seconde une préconisation P .

Les composantes de U et de R sont nommées descripteurs dans la suite de ce manuscrit. Dans l'exemple :

$$((nurse, fatigue, 1.5, 1200), (normal, no_preco))$$

la valeur 1.5 est attribuée à l'attribut *fatigue* de l'agent *nurse* au cycle 1200. La solution correspondante indique un état de l'attribut *fatigue* égal à *normal* qui ne nécessite aucune préconisation (*no_preco*). On retrouve dans cette phase, la structure d'un agent :

- E : agent,
- A : variable d'état,
- V : valeur,
- c : cycle de son environnement temporel.

Cela s'explique par le rôle de la base d'expériences (base de cas) qui traduit "l'histoire" de nos modélisations selon le paradigme SMA. La Figure 4.6 donne une vue de la formalisation d'un cas que nous venons de décrire.

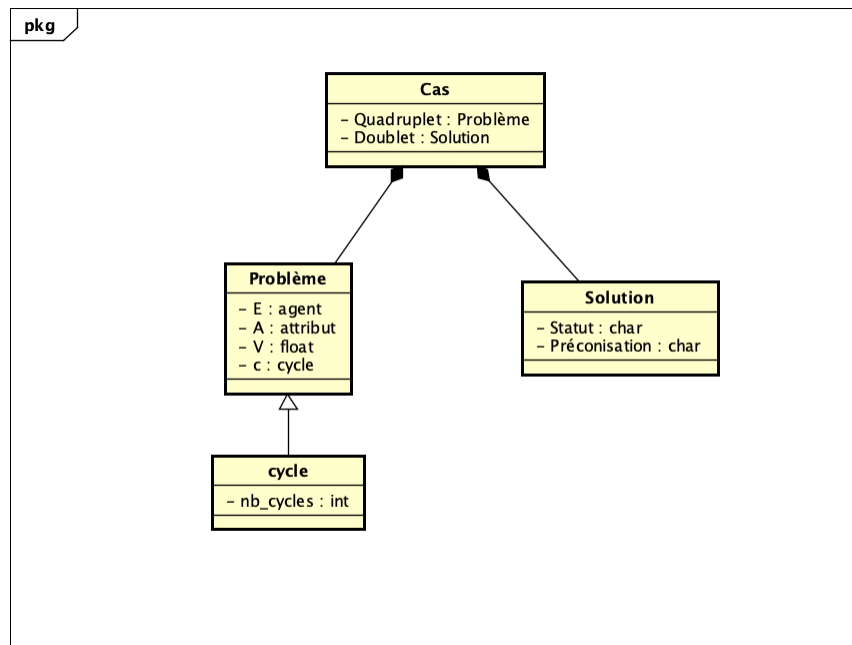


FIGURE 4.6 – Diagramme de classe UML : Élaboration d'un cas

4.5.2 Remémoration

Cette étape consiste à identifier les cas sources de manière à faire correspondre un problème cible (problème à résoudre) avec un problème source lorsqu'il existe une analogie entre eux.

Plus précisément, il s’agit de rechercher la plus forte similitude entre un problème cible et un problème source extrait de la base de cas. Dans notre implémentation, la mesure de similarité $sim(C, S)$ est définie comme suit :

$$\begin{cases} sim(C, S) = \min\left(\frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \sqrt{\sum_{k=1}^4 (1 - I)^2}\right) \\ sim(C, S) \leq \beta \end{cases} \quad (4.1)$$

$$\text{avec } I = \begin{pmatrix} x_{ki} \\ y_{kj} \end{pmatrix} \text{ si } x_{ki} \geq y_{kj}, \text{ sinon } (y_{kj} \geq x_{ki}) I = \begin{pmatrix} y_{kj} \\ x_{ki} \end{pmatrix}.$$

où x_{ki} et y_{kj} sont respectivement les valeurs associées aux éléments du quadruplet cible (C) et à ceux du quadruplet source (S) parmi les n quadruplets. Le fait de comparer les éléments en les quotientant nous exonère des problèmes d’échelle propres aux différents descripteurs. Un cas cible est donc considéré comme similaire d’une part lorsqu’il est le plus proche en termes de distance d’un cas source, d’autre part lorsqu’il est inférieur ou égal à un seuil d’acceptation β . Ce seuil est déterminé par les experts au fil des simulations.

Deux types de descripteurs sont à considérer parmi nos cas : les descripteurs quantitatifs (chiffrés : V, c) et les descripteurs qualitatifs (non chiffrés : E, A). Concernant ces derniers une valeur leur est attribuée conformément à leur classification (par exemple les agents d’une même espèce). Cette “numérisation” est nécessaire à nos calculs de distance. Dans notre étude, les valeurs entre les descripteurs qualitatifs dissemblables sont suffisamment grandes afin d’écarter la comparaison de cas qui n’appartiennent pas à la même classification (cf exemple du Tableau 4.2). Dans l’exemple du Tableau 4.2, l’agent *surgeon* et l’agent *nurse* sont affectés du même nombre 1 car ils appartiennent à la même espèce. En revanche, le quotient de la valeur des attributs *experience* et *fatigue* égal à 10 ($\frac{2}{0,2}$) augmente suffisamment le calcul de distance (4.1) pour écarter ce cas.

Cible/Source	Agent	Attribut	Valeur	Cycle
<i>cible</i>	<i>surgeon</i> (1)	<i>experience</i> (2)	4	15
<i>source</i>	<i>nurse</i> (1)	<i>fatigue</i> (0,2)	2,7	20

Tableau 4.2 – Exemple changement d’échelle entre deux attributs

4.5.3 Réutilisation

Dans cette phase, la solution du problème source la plus similaire au problème cible est appliquée. L’adaptation concerne uniquement le couple (S, P). Ainsi, le statut d’un attribut *normal* d’un cas cible peut correspondre à l’état *alert₁* du cas source le plus similaire et être modifié en ce sens. La préconisation du cas source est, elle aussi appliquée au cas cible.

4.5.4 Révision

En mode non connecté (mode actuel) les solutions sont évaluées par les experts ou par comparaison avec des données réelles. En mode connecté la validation dépend uniquement de l'expert. Dans les deux cas, un statut *draft* (brouillon) est affecté aux enregistrements en attentes de validation.

4.5.5 Mémorisation

Selon l'avis des experts ou du comparatif avec un jeu de données réelles, les enregistrements non encore validés (statut *draft*) sont soit supprimés de la base, considérés comme non conformes, soit enregistrés (statut *draft* → *validated*).

Nous avons porté notre attention dans ces deux premières sections sur la description de l'implémentation du système multi-agents et sur celle du raisonnement à partir de cas. Nous proposons à présent de décrire l'implémentation du couplage SMA/RàPC ainsi que nos spécificités afin de mettre en lumière l'intérêt de cette architecture dans notre objectif de prévention des EIAS.

4.6 Couplage SMA/RàPC

La capitalisation des connaissances dans un système complexe peut s'appliquer à des situations où toutes les solutions du problème sont préalablement stockées. Les notions de parcours de cas similaires et d'adaptation ne sont alors pas actives. En effet, chaque situation est corrélée à un cas pré-existant. La notion de similarité est en revanche au cœur des modèles dynamiques qui s'enrichissent par l'expérience. Notre modélisation d'entités non déterministes entre dans ce cadre et les valeurs imputables aux solutions ne sont pas forcément connues. Il devient donc indispensable de disposer d'une mémoire des précédents cas et d'une ontologie capable de la structurer. Le couplage SMA/RàPC est une réponse possible que nous proposons comme architecture pour notre modèle. La Section 4.6.1 explicite l'implémentation choisie en conséquence.

4.6.1 Vision globale du couplage SMA/RàPC

La Figure 4.7 apporte une vision globale du couplage SMA/RàPC. On y distingue les paradigmes SMA et RàPC. Le système multi-agents est composé d'agents qui appartiennent à différentes espèces telles que : *Personnel*, *Infectious* et *Patient*. Les variables d'état décrites dans la Section 4.4.2 caractérisent les comportements, les actions à mener et les objectifs de chaque agent. Le RàPC est quant à lui représenté selon les quatre cycles (Remémoration, Réutilisation, Révision, Mémorisation) décrits dans la Section 4.5. SMA et RàPC interagissent

sous la forme de coopérations. En effet, la détermination de l'état du système et sa mise à jour dépendent du raisonnement par analogie propre au RàPC. De même l'enrichissement de la base de cas découle de l'état du système exhibé à chaque cycle par le SMA. Au-delà d'une représentation classique, notre architecture intègre une organisation holonique (cf Organisation "Holarchies" : Section 3.3.3 du Chapitre 3) à notre SMA, et la possibilité pour notre raisonnement à partir de cas d'interagir avec plusieurs sources d'acquisition de données. Ces deux spécificités sont abordées dans les deux paragraphes qui suivent.

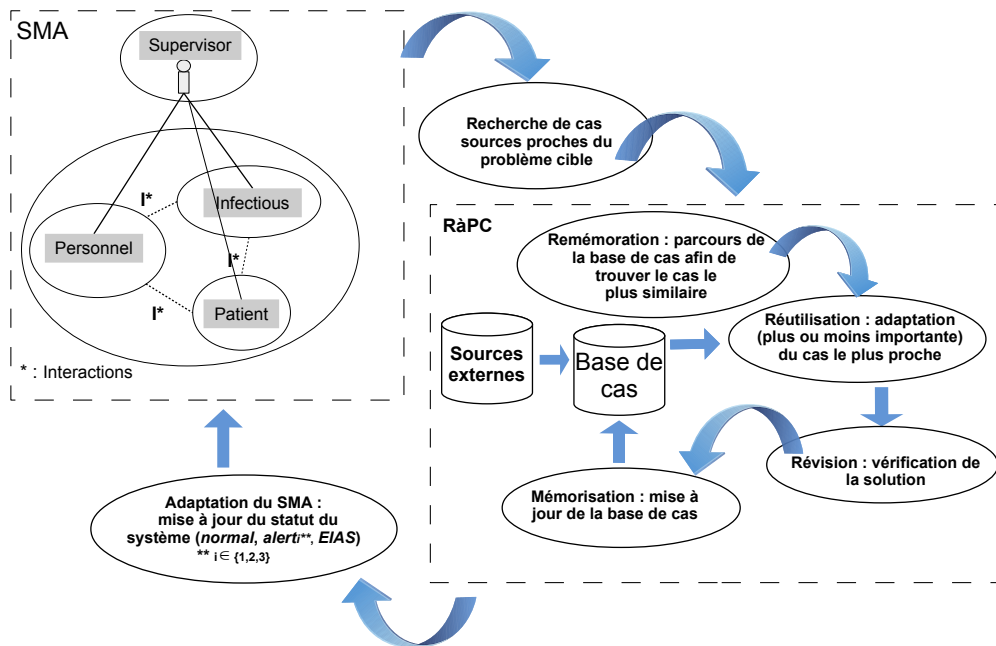


FIGURE 4.7 – Représentation globale couplage SMA/RàPC

4.6.2 SMA : une organisation holonique

D'après Koestler [48], une architecture holonique appartient à une organisation qui répond à trois conditions :

- stabilité : chaque holon est en capacité de réagir quelles que soient les perturbations du système ;
- autonomie : chaque holon peut s'autogérer ;
- coopération : chaque holon peut coopérer avec d'autres holons, voire d'autres groupes d'holons appartenant à différentes couches (niveaux) hiérarchisées.

Cette organisation où l'on retrouve les principes imputables aux agents BDI est transposable

aux systèmes multi-agents. A cela s'ajoute la possibilité de hiérarchiser les rôles de chaque holon, qui est un point clé de notre architecture. Ces éléments ont motivé l'implémentation de cette organisation. Notons qu'un holon peut n'inclure qu'un seul agent : agent p_0 et aa dans notre contexte. La Figure 4.8 décrit les différents types de relations mis en jeu dans la structuration holonique de notre système multi-agents. Dans ce zoom de la Figure 4.7, on remarque tout d'abord qu'il existe des interactions intra et inter espèces. Ainsi, par exemple, une déficience d'un agent m_i associé au bistouri électrique peut non seulement entraîner une modification de l'état du système ($normal \rightarrow alert_i$), mais peut aussi impacter l'état de santé du patient (espèce *Patient*) et mobiliser du personnel (espèce *personnel*) pour pallier cet incident. La figure met aussi en avant deux types de relations transversales matérialisées par l'agent *alert* (aa) puis par les agents de l'espèce *infectious* (i_k).

Les premières sont régies par une hiérarchisation des rôles qui confère à l'agent *alert* une ascendance sur les autres espèces. Ainsi, cet agent "maître" peut influencer l'organisation d'un ou de plusieurs agents d'une ou de plusieurs espèces. Cela se traduit essentiellement par la capacité à modifier l'environnement pour cet agent (modification de l'état du système). Il s'en suit une activation d'actions prédéterminées par l'agent "maître" sur les agents impactés, sans pour autant entraver leur l'autonomie. Les secondes qui excluent toute hiérarchisation peuvent concerner l'ensemble des espèces. Les agents de l'espèce *Infectious* peuvent par exemple interagir avec d'autres agents et produire en réaction une adaptation de leur comportement intrinsèque. Dès lors, un agent de l'espèce *Material* peut se neutraliser selon son niveau d'infection s'il le juge nécessaire.

Nous venons de montrer que l'organisation de type holonique définit les agents comme un ensemble d'holons qui obéissent à des règles. Le degré d'obéissance (choix des règles à appliquer) est d'ailleurs étroitement lié au niveau hiérarchique de l'agent. Cette organisation du modèle impose une structuration des connaissances exposé dans la section qui suit.

4.6.3 Organisation de la mémoire collective

La mémoire individuelle est rattachée à l'agent qui possède son propre espace de données matérialisé par une base locale. Chaque agent possède son propre champ de connaissances qu'il peut enrichir et partager. Dès lors, la mémoire globale est une somme de mémoires individuelles mutualisées. Ce système se suffit à lui même lorsque l'expérience vécue n'est pas une nécessité. Dans le cas contraire, le couplage SMA/RàPC montre l'intérêt conjoint de la dynamique du SMA associée à une base de cas incrémentable (base de cas). Nous avons présenté dans la Section 4.5 la structure d'un cas qui est quasiment similaire à celle d'un agent DBI. Cette similitude s'explique par la coopération entre les deux paradigmes (cf Section 4.6.1). Chaque nouveau cas de la base de connaissances résulte de cas vécus par le système et satisfait notre volonté d'apprentissage par l'expérience. Ce type d'implémentation est classique dans le cadre d'un couplage SMA/RàPC mais nous semble limité car il réduit le champ des sources d'acquisition de données telles que les bases externes ou bien encore les bases de traces. C'est la raison

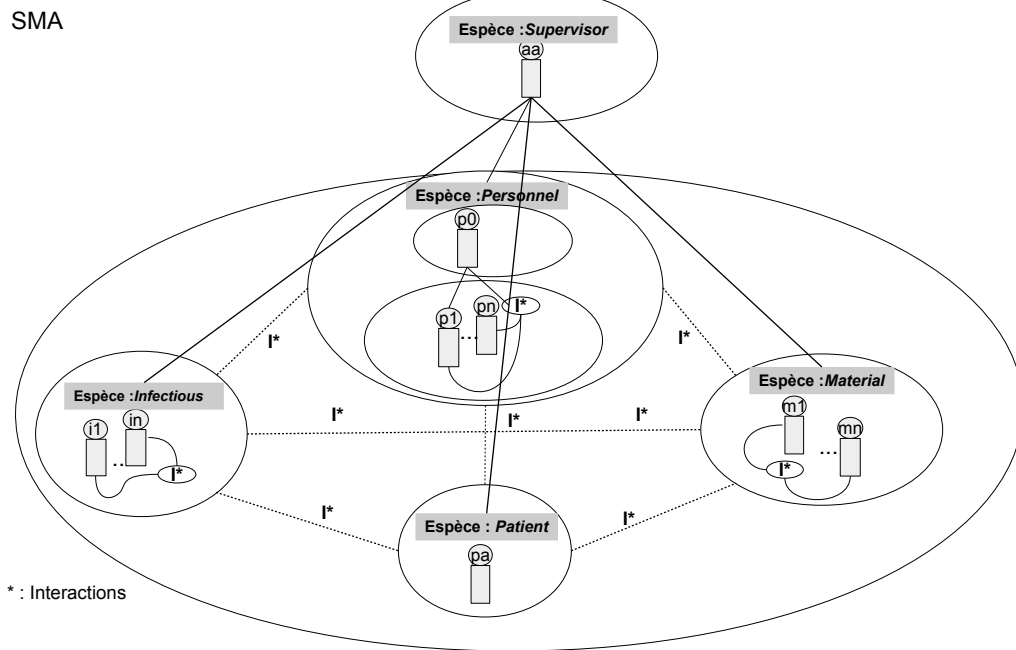


FIGURE 4.8 – Organisation holonique du SMA

pour laquelle nous avons intégré ces possibilités à notre architecture. Ces dernières sont l'objet des deux paragraphes qui suivent.

Les traces

Les bases de traces [60] dans le cadre de notre implémentation proviennent essentiellement de capteurs (capteur de fatigue ...) ou d'appareils de monitoring (fréquence cardiaque ...). Ces sources sont implémentées comme des agents dont les données expriment l'évolution des valeurs de leurs attributs dans un cycle donné. Leur exploitation prévue en mode connecté (désactivée dans notre mode actuel non connecté) nécessite cependant leur restructuration. Nous avons donc intégré à notre architecture un cycle d'intégration en amont de leur utilisation. Celui-ci, présenté dans la Figure 4.9, est régi par trois fonctionnalités :

- sélection : la donnée est soit traitée en flux (base de traces issue par exemple de capteurs), soit filtrée par des requêtes (base de connaissances externes) ;
- structuration : la donnée est structurée au format attendu par la base locale SMA ou par la base de cas ;
- mémorisation : la donnée est stockée dans une base locale avant son transfert.

Ces fonctionnalités formatent les données avant leur intégration dans la base locale SMA ou dans la base de cas (base de connaissances RàPC) lorsqu'il s'agit de bases externes .

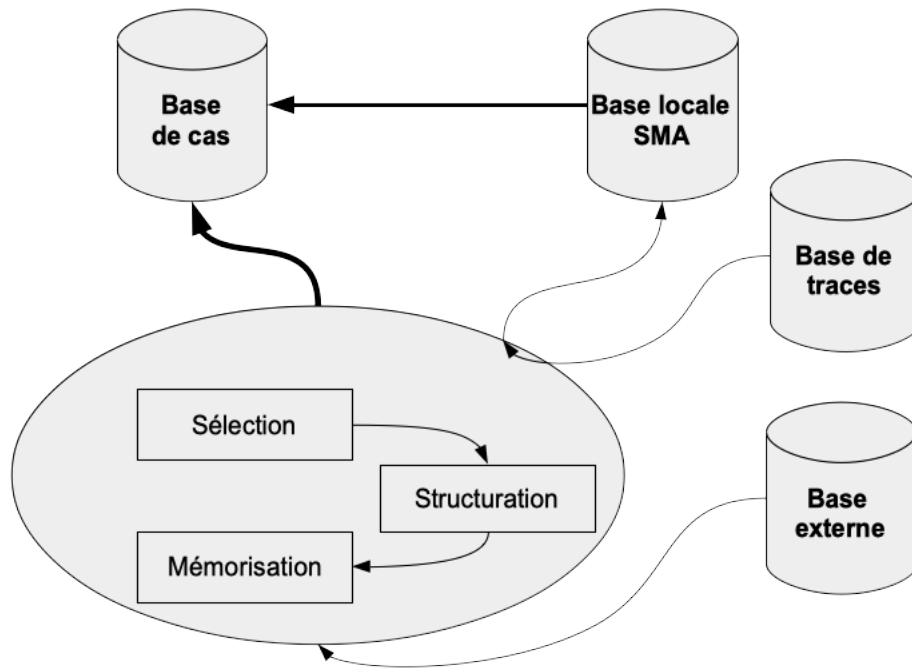


FIGURE 4.9 – Intégration de bases dans le RàPC

Les bases de connaissances externes

La base de traces est certes un élément complémentaire qui trouve tout son sens en mode connecté, mais reste une vision locale d'un univers (prévention des risques dans un bloc opératoire) que l'on peut étendre. L'acquisition de données dans des systèmes distribués pallie cette difficulté grâce à l'ouverture des requêtes à d'autres environnements. Pour les mêmes raisons que celles évoquées pour les bases de traces auxquelles s'ajoutent des restrictions sécuritaires liées au secret médical, ces données externes sont formatées selon le cycle : Sélection, Structuration et Mémorisation (cf Figure 4.9). Elles proviennent de bases de données médicales anonymisées (plateforme AMI : Assistance Médicale Informatisée [3]) utilisées comme jeux de données réelles que l'on souhaite comparer aux données simulées. Cette ouverture à d'autres sources d'acquisition de données offre aussi la possibilité d'accéder à des serveurs WEB. Cette option qui impose des contraintes encore plus draconiennes en termes de cyber-sécurité est écartée pour le moment.

4.7 Synthèse de la présentation de l'architecture

L'implémentation de notre architecture couplant un système multi-agents avec un raisonnement à partir de cas est une première contribution en réponse à notre objectif de prévention

d’EIAS. D’un point de vue plus théorique, il s’agit d’exploiter la capacité d’émergence d’un SMA (cf Section 3.3.3 du Chapitre 3) afin de mettre en exergue des situations inattendues potentiellement importantes. A cela s’ajoute le raisonnement par analogie qui permet, à partir d’une base d’expériences, d’apporter des solutions ou réponses à de nouveaux problèmes ou de nouvelles questions. La section suivante présente l’interface du simulateur, la forme utilisée pour chaque type de données par le RàPC et nos premières simulations. Leurs interprétations visent à mettre en lumière l’intérêt de ce couplage en termes d’émergence.

4.8 Simulateur pour MASSAI

Nous avons décrit dans les premières sections de ce chapitre, notre architecture construite selon le couplage d’un système multi-agents avec un raisonnement à partir de cas. Nos différents choix et contributions en termes de structure y ont été expliqués. La suite de ce chapitre est consacrée à l’aspect simulation. On distingue un aspect visuel à travers le SMA à la différence du RàPC plutôt assimilable à la face cachée du simulateur. Les deux sections suivantes présentent respectivement notre simulateur construit à partir de la plateforme Gama et la structuration des données du RàPC.

4.8.1 Plateforme de simulation CoupSR_PredRB : un outil de prédiction des risques au bloc

Le choix de la plateforme destinée au développement de notre simulateur s’est porté sur GAMA qui permet de construire les modèles dans un environnement de développement intégré (IDE) incorporant le langage GAML (GAMA Modeling Language). Cette spécificité garantit l’enrichissement du modèle et son évolutivité. La plateforme GAMA, riche de plusieurs composants, permet aussi de placer plusieurs modèles de visualisation dans une même fenêtre d’affichage. A tout cela s’ajoute aussi la possibilité de construire des modèles très complexes grâce aux outils de gestion de l’espace très performants dans différents environnements synchronisés à l’intérieur d’un espace continu de référence.

Comme le montre la Figure 4.10, l’interface utilisateur de notre simulateur *CoupSR_PredRB* (Couplage SMA RàPC pour la Prédiction des Risques au Bloc) est scindée en deux blocs. Le premier est réservé au paramétrage et le second à la visualisation des simulations en cours. Dans cet exemple, ce dernier est subdivisé en plusieurs fenêtres où évoluent, entre autres, les agents du bloc opératoire, les constantes patient, les agents infectieux et l’état du système (visualisation des alertes).

L’interface de paramétrage (Parameters) permet d’instancier les seuils d’alerte propres aux attributs surveillés (fatigue humaine, constantes patients, niveau du risque infectieux) et de déterminer le type d’évolution des valeurs (croissantes, décroissantes, aléatoires). A ces paramètres s’ajoutent aussi les possibilités de neutralisation d’un ou plusieurs attributs et de suppres-

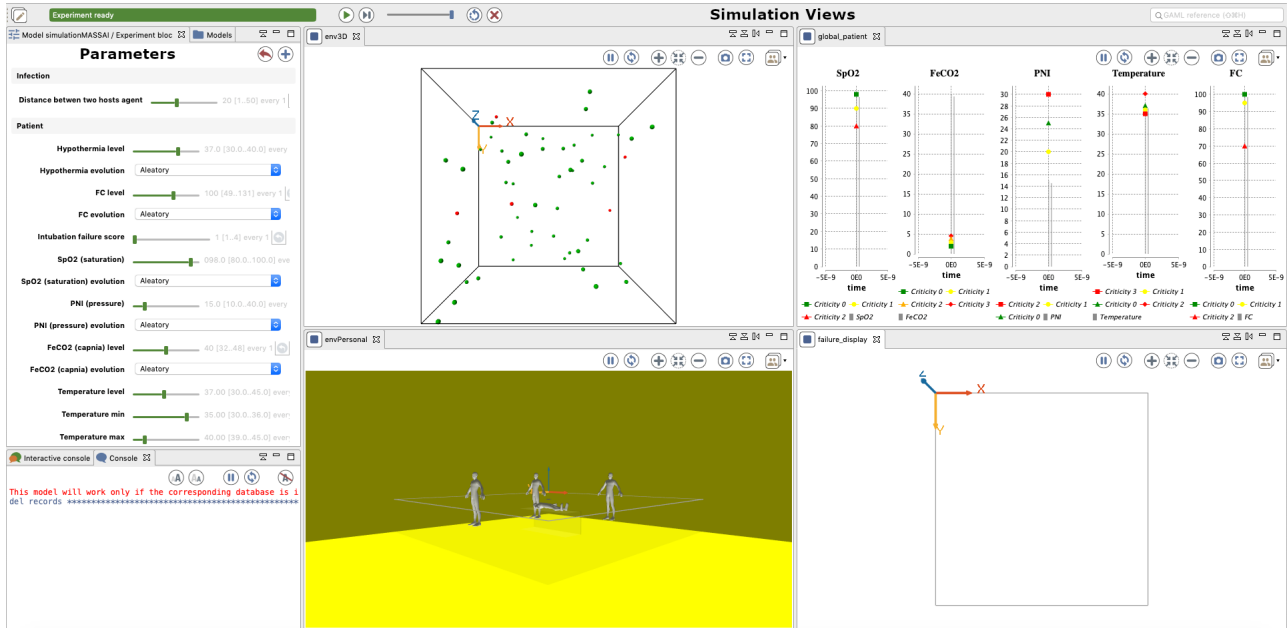


FIGURE 4.10 – Vision globale du simulateur

sion d'un ou plusieurs agents que l'on souhaite exclure de la simulation. La figure 4.11 illustre les possibilités offertes par l'outil "Actions" de la plateforme Gama. Chaque attribut d'un agent peut être désactivé (*alert_fatigue* dans notre exemple) et chaque agent peut être exclu de la simulation par une simple suppression (option "Kill"). Ces fonctionnalités sont utiles pour évaluer les interactions entre agents.

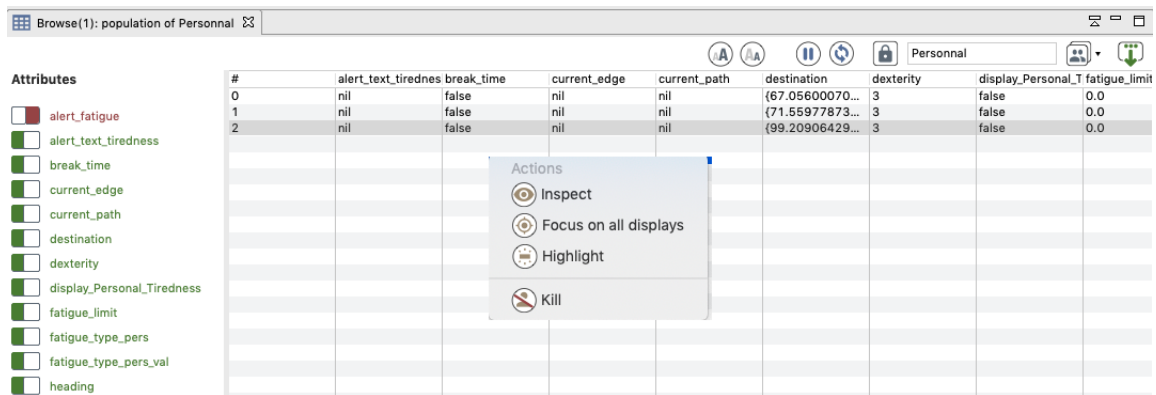


FIGURE 4.11 – Contrôle des attributs et des agents

L'interface de visualisation de la simulation en cours offre plusieurs vues de l'évolution des différents attributs que l'on surveille. Ces vues revêtent plusieurs formes selon le type d'agent modélisé. Ainsi, les praticiens (agents de l'espèce *Personnel*) sont représentés par des objets (personnages) évoluant dans le bloc opératoire. L'évolution des valeurs de leur attribut *fatigue* est visible individuellement lors du passage de la souris. Les constantes patients sont quant à

elles représentées par des graphiques (cf vue "global_patient") et les agents de l'espèce "Infectious" par une vue 3D. Les alertes qui apparaissent dans la fenêtre "failure display" indiquent leur proximité par rapport à la survenue d'un EIAS (*far* (éloigné), *near* (proche), *imminent*). Cette approche visuelle est légitime dans un contexte de simulation où l'on ambitionne de mettre en évidence des alertes.

Nous avons intégré le raisonnement à partir de cas et motivé nos choix dans la section précédente. La section qui suit présente le mode opérationnel du RàPC et en particulier les formes et méthodes utilisées pour l'exploitation de la base de cas.

4.8.2 Exploitation des données : une approche par analogie

L'implémentation d'un raisonnement à partir de cas et la qualité de son exploitation sont étroitement liées au soin apporté lors de la phase d'élaboration. En effet, la détermination d'un cas et les index utilisés conditionnent l'efficacité opérationnelle du RàPC. Les interactions entre les deux paradigmes (cf Section 4.6.1) imposent soit la création d'une passerelle soit une même structuration des données. Cette seconde option que nous avons choisie, nous a conduit à intégrer les variables d'état d'un agent comme des descripteurs d'un cas tout en conservant la même structure de données. Ainsi la variable d'état (attribut) *fatigue* et l'agent *surgeon* sont repris sous cette même forme et correspondent aux descripteurs *A* (attribut) et *E* (entité) d'un cas. Actuellement, la base de cas est enrichie à partir du système multi-agents. L'exemple de la Figure 4.12 représente un ensemble de cas qui illustre les états possibles pour chacun d'eux (*normal*, *failure*). Les attributs (*temperature*, *FeCO2*, *SpO2*, *PNI*, *FC*) de l'agent *patient0* choisis pour cette simulation possèdent un statut égal à *normal* pour les cycles allant de 15 à 25. Il est égal à *failure* dans le cycle 65. Il s'agit d'une vision détaillée de l'ensemble des agents et attributs du système sur plusieurs cycles. L'état du système qui, nous le rappelons, est piloté par l'agent *alert* est une synthèse de l'ensemble des états de tous les attributs observés à un cycle donné. Tout événement indésirable, résultant de l'évolution d'un ou plusieurs attributs, modifie alors l'état du système dont la valeur devient *failure* pour le cycle en cours. Les précédents états parcourus dans l'ordre antéchronologique prennent respectivement les valeurs : *alert3*, *alert2* et *alert1*.

La Figure 4.13 montre un état *failure* au cycle 65, puis respectivement *alert3*, *alert2* et *alert1* pour les cycles précédents. On observe que l'état *failure* du système résulte dans notre exemple de la défaillance d'un seul attribut (cf dernière ligne de la Figure 4.12 : statut *failure* de l'attribut *fatigue*). Les niveaux d'alertes qui précèdent cet état sont ceux qui seront proposés par le système lors de la recherche d'analogie entre le cas cible et le cas source.

Le couplage SMA/RàPC est une architecture destinée à observer la dynamique virtuelle d'une représentation possible du réel mais permet aussi l'émergence de phénomènes imprévisibles. Ce dernier aspect est essentiel dans un contexte où il est primordial d'étendre l'univers des "possibles". La section suivante lui est consacrée.

date	cycle	grp_agent	agent	grp_attribut	attribut	value	state	preconization	case_ind_number
Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre	Filtre
2020-10-14 14:34...	35	Patient	Patient0	Constante	temperature	36.7	normal	no_preco	1
2020-10-14 14:34...	35	Patient	Patient0	Constante	FeCO2	40	normal	no_preco	2
2020-10-14 14:34...	35	Patient	Patient0	Constante	SpO2	98.1	normal	no_preco	3
2020-10-14 14:34...	35	Patient	Patient0	Constante	PNI	14.6	normal	no_preco	4
2020-10-14 14:34...	35	Patient	Patient0	Constante	FC	87	normal	no_preco	5
2020-10-14 14:34...	35	Material	Material0	Sound_env	noise_mat	30	normal	no_preco	6
2020-10-14 14:34...	35	Personnal	Personnal0	Health	fatigue	1,4	normal	no_preco	7
2020-10-14 14:34...	35	Personnal	Personnal1	Health	fatigue	1,4	normal	no_preco	8
2020-10-14 14:34...	35	Personnal	Personnal2	Health	fatigue	1,4	normal	no_preco	9
2020-10-14 14:34...	45	Patient	Patient0	Constante	temperature	36.5	normal	no_preco	10
2020-10-14 14:34...	45	Patient	Patient0	Constante	FeCO2	43	normal	no_preco	11
2020-10-14 14:34...	45	Patient	Patient0	Constante	SpO2	98	normal	no_preco	12
2020-10-14 14:34...	45	Patient	Patient0	Constante	PNI	14.7	normal	no_preco	13
2020-10-14 14:34...	45	Patient	Patient0	Constante	FC	85	normal	no_preco	14
2020-10-14 14:34...	45	Material	Material0	Sound_env	noise_mat	35	normal	no_preco	15
2020-10-14 14:34...	45	Personnal	Personnal0	Health	fatigue	1.9	normal	no_preco	16
2020-10-14 14:34...	45	Personnal	Personnal1	Health	fatigue	1.7	normal	no_preco	17
2020-10-14 14:34...	45	Personnal	Personnal2	Health	fatigue	2.1	normal	no_preco	18
2020-10-14 14:34...	55	Patient	Patient0	Constante	temperature	36.3	normal	no_preco	19
2020-10-14 14:34...	55	Patient	Patient0	Constante	FeCO2	46	normal	no_preco	20
2020-10-14 14:34...	55	Patient	Patient0	Constante	SpO2	98.6	normal	no_preco	21
2020-10-14 14:34...	55	Patient	Patient0	Constante	PNI	15	normal	no_preco	22
2020-10-14 14:34...	55	Patient	Patient0	Constante	FC	88	normal	no_preco	23
2020-10-14 14:34...	55	Material	Material0	Sound_env	noise_mat	35	normal	no_preco	24
2020-10-14 14:34...	55	Personnal	Personnal0	Health	fatigue	2.4	normal	no_preco	25
2020-10-14 14:34...	55	Personnal	Personnal1	Health	fatigue	1.9	normal	no_preco	26
2020-10-14 14:34...	55	Personnal	Personnal2	Health	fatigue	2.4	normal	no_preco	27
2020-10-14 14:34...	65	Patient	Patient0	Constante	temperature	36.3	normal	no_preco	28
2020-10-14 14:34...	65	Patient	Patient0	Constante	FeCO2	49	normal	no_preco	29
2020-10-14 14:34...	65	Patient	Patient0	Constante	SpO2	98.3	normal	no_preco	30
2020-10-14 14:34...	65	Patient	Patient0	Constante	PNI	14.6	normal	no_preco	31
2020-10-14 14:34...	65	Patient	Patient0	Constante	FC	87	normal	no_preco	32
2020-10-14 14:34...	65	Material	Material0	Sound_env	noise_mat	34.9	normal	no_preco	33
2020-10-14 14:34...	65	Personnal	Personnal0	Health	fatigue	2.9	normal	no_preco	34
2020-10-14 14:34...	65	Personnal	Personnal1	Health	fatigue	2.7	normal	no_preco	35
2020-10-14 14:34...	65	Personnal	Personnal2	Health	fatigue	3	failure	pause	36

FIGURE 4.12 – Détail de l'état du système

simul_number ^	date	cycle	state
Filtre	Filtre	Filtre	Filtre
1	2020-10-14 14:34...	15	normal
1	2020-10-14 14:34...	25	normal
1	2020-10-14 14:34...	35	alert1
1	2020-10-14 14:34...	45	alert2
1	2020-10-14 14:34...	55	alert3
1	2020-10-14 14:34...	65	failure

FIGURE 4.13 – Focus sur l'état du système

4.8.3 Simulateur et émergence

Nous avons exprimé dans la Section 3.3.3 les différentes définitions d'un phénomène émergent en nous appuyant sur la littérature [83]. La question qui se pose concerne son évaluation à partir

du modèle et donc sa caractérisation lors des simulations. Dans cet objectif, nous avons considéré un phénomène émergent comme la survenue d'un événement imprévisible et soudain inhérent au collectif d'agents. Autrement dit, il s'agit de phénomènes qui ne seraient pas apparus sans interactions particulières. L'exemple du Tableau 4.3 est une illustration d'un phénomène émergent. Il a été construit "artificiellement" en activant puis en désactivant l'agent *staphylo* à partir de l'interface de paramétrage. Ainsi, nous observons qu'une interaction (caractérisée par une distance) entre l'agent *staphylo* et *patient0* entraîne des mouvements de type *move* pour l'agent *nurse*. Ce mouvement est la conséquence d'une dualité caractérisée par la volonté de soigner tout en préservant la sécurité du patient. En revanche, le mouvement *in_situ* observable lorsque l'agent *staphylo* est désactivé (*simul_umber* = 8) traduit un retour à un comportement considéré comme classique (déjà identifié). Ces comportements répondent à l'architecture BDI et notamment aux attributs *intention* et *movement* de l'agent *nurse* (cf Section 4.4.2). L'émergence de ce mouvement atypique de la part de l'agent *nurse* a été piloté par un jeu d'interactions ciblées (agent *staphylo* avec l'agent *patient0*). Cet exemple confirme la relation très forte entre le phénomène d'émergence et les interactions entre agents. La détermination de ces dernières et leur analyse est l'objet du chapitre 5.

simul_number	agent	attribut	state
7	<i>staphylo</i>	<i>desire</i> (contaminer)	<i>normal</i>
7	<i>patient0</i>	<i>infected</i>	<i>failure</i>
7	<i>nurse</i>	<i>movement</i>	<i>move</i>
8	<i>patient0</i>	<i>infected</i>	<i>normal</i>
8	<i>nurse</i>	<i>movement</i>	<i>in_situ</i>

Tableau 4.3 – Exemple de phénomène émergent

La section suivante décrit la singularité de notre plateforme en comparaison des plateformes : "Tour de contrôle" [68] et "3D Virtual Operating Room" [23] considérées comme assez proches.

4.8.4 Comparaison *CoupSR_PredRB* avec avec des plateformes simulant la salle d'opération

Le parcours de la littérature nous a permis de recenser deux plateformes qui semblent se rapprocher conceptuellement de notre projet. La première plateforme [68] est un système d'apprentissage profond dédié à la reconnaissance de flux pendant le travail chirurgical. Dans un but de simplification, nous la nommerons "Tour de contrôle" dans la suite de cette section. La seconde plateforme [23] est un jeu de rôles destiné aux étudiants en médecine et en soins infirmiers. Nous proposons dans la suite de cette section de donner les principes généraux de ces deux projets, pour ensuite les comparer entre elles.

Machine et apprentissage profond pour la reconnaissance du flux de travail pendant la chirurgie

L'objectif principal de cette "*Tour de contrôle*" est d'observer le flux du travail chirurgical afin que le système prenne conscience du contexte chirurgical pour améliorer la qualité des soins et la sécurité des patients. Les techniques d'apprentissage profond sont les outils utilisés pour analyser les activités en cours de chirurgie à partir de caméras endoscopiques ou de caméras fixées au plafond. L'étude conduite actuellement porte sur deux chirurgies : l'ablation de la vésicule biliaire (cholécystectomie) et le bypass gastrique.

Jeu de rôles "*3D Virtual Operating Room*"

La plateforme "*Virtual Operating Room*" [23] est un jeu de rôle destiné aux étudiants en médecine et en soins infirmiers. Il consiste pour chaque joueur à prendre le rôle de praticien qu'il a l'intention de jouer. Chacun d'entre eux se retrouve alors dans une salle d'opération virtuelle avec un patient prêt à être opéré. Cet outil permet au joueur d'être confronté à des situations professionnelles résultant de cas réels d'événements indésirables. Un bilan établi pour chaque joueur permet d'analyser ses comportements.

Le tableau 4.4 récapitule des éléments de comparaison entre notre plateforme *CoupSR_PredRB* et les plateformes "*Tour de contrôle*" et "*3D Virtual Operating Room*". Nous comparons les "Objectifs", le "Personnel ciblé" et les "Données d'entrée".

Éléments comparés	CoupSR_PredRB	3D Virtual Operating Room	Tour de contrôle
Objectifs	Générer des alertes et identifier les nouveaux événements indésirables graves et prédire leur survenue	Former les étudiants à la prévention des risques dans un bloc opératoire	Améliorer les soins et alerter
Personnel ciblé	Praticiens de bloc opératoire	Étudiants en médecine et en soins infirmiers	Praticiens de bloc opératoire
Mode opérationnel	Un simulateur produit un grand nombre de scénarios pour l'émergence et l'analyse des événements indésirables qui ne peuvent être prévus a priori	Jeu de rôle où les joueurs réagissent à de graves événements	Outil d'IA (apprentissage profond) pour analyser les activités du bloc opératoire
Données d'entrée	Environnement de simulation	Joueurs (étudiants)	Données issues de caméras endoscopiques ou de caméras fixées au plafond

Tableau 4.4 – Comparaison SMA/RàPC avec "*3D Virtual Operating Room*"

Comparaison *CoupSR_PredRB* avec "*Tour de contrôle*"

Nous constatons une similitude forte entre les deux outils par rapport à l'objectif de sécurisation des soins (générateur d'alertes dans les deux cas). Il en est de même pour le personnel ciblé qui

est identique pour les deux projets comparés. Ceci dit, au moins trois distinctions sont observables. La première concerne nos objectifs qui vont au-delà de l’alerte grâce à l’intégration d’un outil d’aide à la décision : prédictivité de l’évolution de l’état du système. La seconde concerne le mode opérationnel. En effet, ce sont des outils d’IA (apprentissage profond) qui ont été choisis pour la “*Tour de contrôle* ” alors que notre architecture couple un système multi-agents avec un raisonnement à partir de cas. Cette nuance positionne la “*Tour de contrôle* ” comme une plateforme d’apprentissage connectée avec le réel, alors que notre contexte de simulation est plutôt destiné à imaginer le réel (prédiction d’évènements a priori non détectables). La troisième distinction est liée à la seconde : les données de notre architecture résultent de nos simulations alors que celles de la “*Tour de contrôle* ” sont issues du bloc opératoire. La notion de simulation multi-agents est donc l’élément majeur qui distingue les deux outils. Ce paradigme porté par la capacité des SMA à produire des émergences est la réponse que nous avons choisie pour générer des situations a priori non prévisibles. En conclusion, la “*Tour de contrôle* ” se rapproche plus de la version connectée de notre simulateur (données issues de : capteurs de fatigue, de capteurs de mouvement, de sonomètre ...) que nous envisageons en perspective.

Comparaison *CoupSR_PredRB* avec “3D Virtual Operating Room”

A la lecture du Tableau 4.4, nous pouvons voir des différences pour chacun des éléments comparés. En termes d’objectifs, la plateforme “3D Virtual Operating Room” vise à former les étudiants tandis que notre architecture est conçue pour simuler le monde de la salle d’opération afin de produire des événements indésirables a priori imprévisibles. Les données d’entrée sont produites dans le cas du simulateur SMA, par l’environnement de simulation (agents, capteurs), tandis que dans le cas de la plateforme “3D Virtual Operating Room” elles sont produites par les joueurs. Ensuite, ce sont les étudiants qui sont ciblés dans “3D Virtual Operating Room”, alors que dans notre simulateur, ce sont les professionnels de santé déjà formés ou en formation. Enfin, le mode de fonctionnement diffère entre les deux outils, sachant que “3D Virtual Operating Room” implique les joueurs sous la forme d’un jeu de rôle, alors que notre simulateur fait évoluer des agents virtuels dans l’environnement de la salle d’opération. En conclusion, notre architecture, qui permet d’alerter et de prévenir de nouveaux risques, répond à un besoin opérationnel et éventuellement éducatif, alors que la plateforme “3D Virtual Operating Room” est exclusivement destinée à la formation.

4.9 Bilan du chapitre

Ce chapitre a pour vocation de justifier le couplage SMA/RàPC à travers la description de son implémentation. Nous avons présenté notre première contribution caractérisée par une architecture, qui permet de générer des émergences, mais aussi de capitaliser et d’exploiter les expériences simulées. Dans cet objectif, nous avons tout d’abord détaillé l’implémentation du modèle SMA. Les différentes variables d’états propres à chaque espèce d’agents du modèle

SMA ont été explicitées ainsi que la modélisation (rôles et buts) choisie pour chacun d’eux. Nous avons ensuite décrit les cinq phases du cycle RàPC dans le contexte de notre modélisation. Cela nous a conduit à mettre en avant l’intérêt de ce couplage caractérisé par la question cruciale de l’acquisition de données. Ce terme a d’ailleurs été étendu à la notion d’acquisition de connaissances dans un contexte où l’apprentissage est un axe fort de notre projet d’étude. Dans la Section 4.8.3, l’extrait de simulation choisi montre un exemple de phénomène émergent. Il révèle aussi le rôle majeur des interactions dans sa survenue. L’identification de ces interactions et les analyses qui en découlent sont l’objet des Chapitres 5 et 6. Enfin, la Section 4.8.4 a mis en exergue la spécificité de notre simulateur (émergence et prédictivité des risques au sein d’un bloc opératoire) en comparaison d’une plateforme éducative (jeu de rôles destiné aux étudiants en médecine) et d’une “*Tour de contrôle*” (Machine et apprentissage profond pour la reconnaissance du flux de travail pendant la chirurgie).

Dans le Chapitre 5, nous proposons un comparatif entre différentes méthodes de clusterisation destinées à identifier les interactions entre agents.

Chapitre 5

Clusterisation : comparatif entre plusieurs méthodes

Sommaire

5.1	Introduction	91
5.2	Clusterisation : une méthode pour identifier des comportements similaires	92
5.2.1	Un regroupement non supervisé	92
5.2.2	<i>similVar</i> et <i>similVar+</i> : des outils d'analyse d'interactions	94
5.2.3	<i>similVar</i> : Clusterisation des données pour évaluer les interactions entre agents	94
5.2.4	<i>similVar+</i> : Clusterisation des données pour évaluer les interactions entre agents en décalage dans le temps	97
5.3	Résultats	99
5.3.1	Méthode	101
5.3.2	Présentation des résultats	103
5.3.3	Discussion	104
5.4	Bilan du chapitre	108

5.1 Introduction

Nous avons, dans le chapitre précédent, exposé notre première contribution en réponse à la problématique de prédiction d'événements indésirables dans un contexte non déterministe. Cette contribution est matérialisée par une architecture basée sur un couplage SMA/RàPC. L'intérêt de cette architecture réside dans la capacité des SMA à produire des phénomènes émergents, et dans la capacité du RàPC à rechercher des solutions par analogie avec des expériences passées. Ce modèle reproduit l'univers du bloc opératoire dont les interactions entre agents sont la source d'émergences ou d'événements indésirables associés aux soins. La question sous-jacente concerne les relations de causalité qui peuvent exister entre ces interactions et la survenue d'un EIAS ou d'un phénomène émergent. Ces interactions sont de natures variables (neutres, limitatives ou amplifiantes) et cela joue un rôle majeur dans l'évolution de l'état du système. L'identification des interactions et l'analyse de leur nature est donc au cœur de notre réflexion dont l'objectif est centré sur la recherche de causalité(s) d'un EIAS souvent corrélée(s) à un phénomène émergent.

Le présent chapitre, consacré à l'identification des interactions, expose notre seconde contribution qui se traduit par la mise en œuvre de méthodes de clusterisation. Les clusters formés regroupent les agents dont les valeurs des attributs varient de manière similaire en fonction de critères prédéfinis (exemple : regroupement en fonction de variations similaires entre deux cycles). Nous pensons, en effet, qu'un ensemble d'agents classés selon les mêmes critères de variation peut traduire un jeu d'influences entre eux conduisant à la survenue d'un EIAS ou d'un phénomène émergent. Ce positionnement se justifie par la caractérisation d'une influence : une influence exprime une relation de dépendance entre un agent A et un agent B qui induit des interactions entre eux. Il convient aussi de noter que toute influence implique une interaction mais pas forcément l'inverse. L'exemple fictif de la Figure 5.1 montre une variation assez proche entre l'attribut *anxious* (A) de l'agent *surgeon* et l'attribut *decibel* (D) de la ressource *phone*. Le cluster regroupant D et A (représenté par leur liaison) n'inclut pas l'attribut *tiredness* (T) qui n'est pas impacté par le bruit (*decibel*, D). Tout l'enjeu à travers nos outils de regroupement consiste donc à identifier les influences effectives entre les agents. Les différentes méthodes de classification que nous avons explorées et exposées dans le Chapitre 3 sont certes adaptées à cette identification, mais présentent des faiblesses que nous avons contournées avec deux algorithmes : *similVar* et *similVar+*. Ces deux outils d'aide à la décision sont décrits dans la suite de ce chapitre. Leur fonctionnement et leur implémentation y sont présentés. La performance de *similVar* est ensuite évaluée puis discutée avant de conclure ce chapitre.

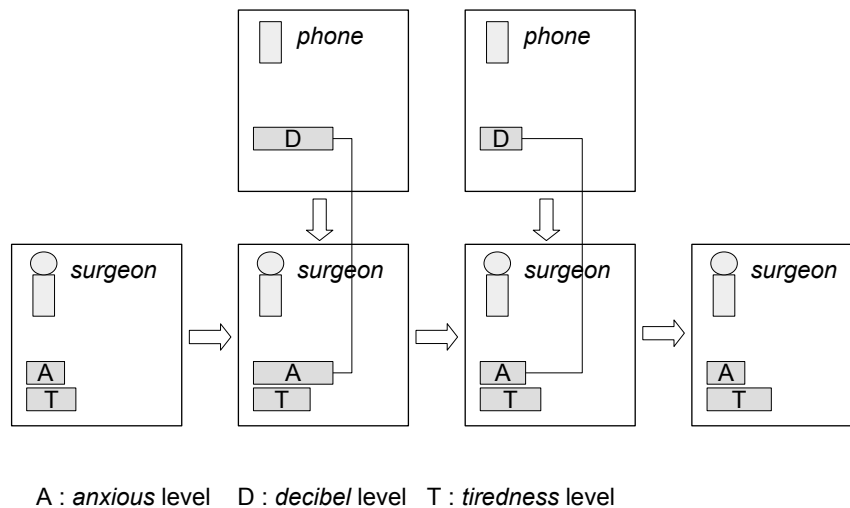


FIGURE 5.1 – Exemple d’influences entre un agent et une ressource

5.2 Clusterisation : une méthode pour identifier des comportements similaires

Rappelons que la classification non supervisée, ou clustering, regroupe des individus sachant :

- que l’on ne connaît pas le nombre de groupes ;
- que l’on ignore le groupe d’appartenance d’un individu ;
- que l’on souhaite classer les individus de manière homogène en fonction de différentes variables.

Il s’agit donc d’engendrer des groupes où l’évolution des attributs d’un agent d’un même groupe sont plus similaires que celles appartenant à d’autres groupes. On construit ainsi des groupes d’agents qui présentent des évolutions de valeurs d’attributs similaires. Le paragraphe qui suit décrit la notion de regroupement non supervisé.

5.2.1 Un regroupement non supervisé

En préambule, ce paragraphe définit nos clusters et les éléments qui le composent. En considérant l’analyse des interactions comme notre objectif à atteindre, nous proposons de regrouper

les variations de la valeur des attributs des différents agents entre deux cycles ($cycle_i$ et $cycle_j$, $(i, j) \in \llbracket 1, n \rrbracket^2$). Ces variations sont le résultat de calculs dans une période donnée et ne sont donc pas définies par avance. Dans ce cadre, l'intérêt d'un regroupement non supervisé repose sur la possibilité d'analyser des données sans connaissance préalable des groupes formés. Cette forme d'apprentissage prend tout son sens lorsqu'il n'existe pas de tutorat ou de réponse en cours de clusterisation. Les algorithmes sont autonomes dans leur travail de regroupement de données. Nous présentons dans la Figure 5.2 un exemple de clusterisation supervisée et non supervisée. Dans le premier cas, un apprentissage à partir de données déjà étiquetées a permis de déterminer la fonction de mappage $y = f(x)$: correspondance entre une donnée d'entrée x et une donnée de sortie y . Cette fonction définit la répartition au sein des différentes classes. Ainsi un nouvel élément est ajouté en à $G3$ selon f . Dans notre exemple, f répartit au sein d'une même classe les objets de mêmes formes, et le nombre de classes est dépendant du nombre de formes. En revanche, les groupes formés dans le cadre d'une clusterisation non supervisée rassemblent les éléments de même type (losange, rectangle, cercle) uniquement par similitude : "Clusterisation non supervisée" présentée dans notre Figure 5.2. Il s'agit d'optimiser un "critère d'homogénéité" afin que chaque classe soit la plus homogène possible, tout en distinguant le plus possible chacune des classes. L'indice de ressemblance noté IR est un exemple de critère d'homogénéité largement utilisé dans la clusterisation de données. C'est une mesure de proximité qui correspond dans les propriétés qui suivent, à l'application r définie de $\Delta \times \Delta$ (ensemble des objets : $\Delta = \{x_1, x_2, \dots, x_n\}$) dans \mathbb{R}_+ vérifiant :

$$\begin{aligned} r(x_i, x_j) &= r(x_j, x_i), \forall (x_i, x_j) \in \Delta \times \Delta \\ r(x_i, x_i) &= R : \text{un objet ressemble à lui-même} \\ r(x_i, x_j) &\leq R, \forall (x_i, x_j) \in \Delta \times \Delta : R \text{ majore la ressemblance.} \end{aligned}$$

L'indice de ressemblance IR est défini selon la relation :

$$\begin{aligned} IR(x_i, x_j) &= \frac{1}{R} r(x_i, x_j), \forall (x_i, x_j) \in \Delta \times \Delta \\ IR : \Delta \times \Delta &\mapsto [0, 1] \end{aligned}$$

Remarque : $r(x_i, x_j)$ est souvent associé à un calcul de distance.

Les données à analyser, issues de nos simulations, ne sont pas identifiées par leur appartenance à un groupe. En effet, nous ne connaissons pas au préalable la valeur des différentes variations qui sont déterminées entre deux cycles au fil des simulations. Nous avons donc opté pour une clusterisation non supervisée qui nous le rappelons est adaptée aux situations : où l'on ne connaît pas le nombre de groupes des données à classer, où l'on ne connaît pas le groupe des données à classer et enfin où l'on souhaite obtenir des groupes les plus homogènes possible. Les méthodes basées sur la *Partition* et présentées dans le Chapitre 3 sont des outils appropriés dont nous nous sommes inspirés pour construire nos algorithmes *similVar* et *similVar+*. Ces der-

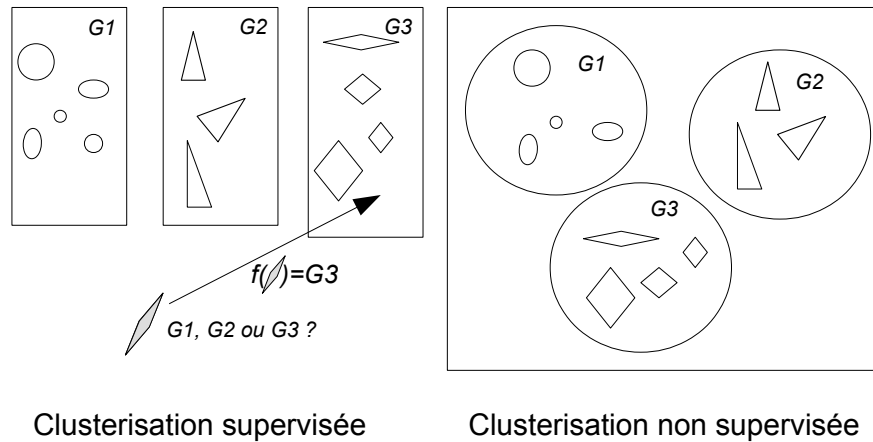


FIGURE 5.2 – Clusterisation supervisée et non supervisée

niers, explicités dans les trois paragraphes qui suivent, pallient certaines limites que nous avons relevées dans la Section 3.5 du Chapitre 3 lors de l’étude des algorithmes de clusterisation.

5.2.2 *similVar* et *similVar+* : des outils d’analyse d’interactions

Pour mémoire, l’objectif de cette thèse qui vise à évaluer l’état d’un système dans le temps est double. Il consiste en premier lieu à “alerter” lorsque ce dernier s’écarte d’une situation considérée comme stable et en second lieu à prédire son évolution. Dans ce chapitre, nous nous attachons en particulier à la notion de classification (clusterisation) qui permet d’analyser la survenue d’un événement indésirable consécutive aux interactions entre agents. Il en découle un possible ajustement des niveaux d’alerte et du comportement des agents. Dans cette optique, notre contribution est matérialisée par deux algorithmes *similVar* et *similVar+* décrits dans les deux paragraphes suivants. Nous avons souhaité aller au-delà d’autres méthodes de clusterisation (cf Section 3.5 du Chapitre 3), certes intéressantes mais restrictives dans l’analyse des interactions entre agents. La section qui suit décrit l’algorithme *similVar*.

5.2.3 *similVar* : Clusterisation des données pour évaluer les interactions entre agents

L’algorithme de classification appelé *similVar* est construit selon nos propres règles de classification. Il consiste à regrouper les expériences passées extraites de notre base de cas pour lesquelles les variations de la valeur des attributs sur une période donnée sont identiques. Nous rappelons que les attributs sont les variables d’état que nous avons affectées à chacun de nos agents dont le détail est présenté dans la Section 4.4.1 du Chapitre 4.

Lors de la clusterisation, l'algorithme classe tout d'abord les attributs appartenant à la classe C_t (attributs appartenant à une même période) en clusters V_k regroupant les attributs avec des variations similaires. L'algorithme détermine ensuite les occurrences des attributs en doublons au sein d'un même cluster V_k ainsi que la dispersion $Disp$ au sein du cluster. Celle-ci consiste à mesurer la dispersion absolue de la distribution des attributs clusterisés, sachant que les distributions ont la même valeur centrale et sont a priori de type gaussien. Elle est calculée selon la relation :

$$Disp = \sqrt{\frac{1}{n} \sum_{i=1}^n (O_i - \bar{O})^2}$$

O_i correspond aux occurrences de l'attribut A_i et $\bar{O} = \frac{1}{n} \sum_{i=1}^n O_i$ $i \in \llbracket 1, n \rrbracket$.

La détermination du nombre de doublons et de la dispersion au sein des clusters permet de différencier deux clusters a priori semblables (possédant les mêmes attributs). Cela augmente donc la précision de notre clusterisation.

Présentation de l'algorithme

L'algorithme 1 comprend les étapes suivantes :

- Itérations jusqu'à la dernière période q (ligne 2 à ligne 31) :
 - Ligne 3 : Regrouper dans des clusters V_k les attributs par variations identiques observées dans la période t à partir de la classe $C_t = \{A_1, A_1, \dots, A_n\}$. $V_k = \{Va(A_i) = Va(A_j) = \dots = Va(A_p)\}$, $\{i, j, p\} \in \llbracket 1, n \rrbracket$, $Va(A_i)$: variation de A_i dans une période donnée.
 - Ligne 4 : Étiqueter chaque cluster en fonction du statut de chaque attribut :

$$\begin{cases} normal \text{ si } \sum_{i=1}^u S_i = "normal" > \sum_{j=1}^v S_j = "failure" \\ failure \text{ sinon} \end{cases}$$
 S_k = état de chaque attribut ($k \in \mathbb{N}$).
 - Ligne 5 : Déterminer les occurrences des attributs en doublons du cluster V_k : $Dup(V_k)$.
 - Ligne 6 : Déterminer la dispersion $Disp(V_k)$ du cluster V_k .
 - Ligne 7 à 30 : Sélectionner le meilleur cluster lorsque chacun d'eux inclut le même ensemble d'attributs. Le même candidat est celui qui maximise les occurrences d'attributs en doublons ou celui qui minimise $Disp$ en cas d'égalité.
 - ligne 11 à 16 : Sélection du cluster qui maximise le nombre de doublons. Le cluster qui n'est pas sélectionné est exclu : $V_i \vee V_i \leftarrow \emptyset$.
 - ligne 17 à 25 : Sélection du cluster qui minimise la dispersion $Disp$. Le cluster qui n'est pas sélectionné est exclu : $V_i \vee V_i \leftarrow \emptyset$.

La détermination du nombre de doublons et l'homogénéité au sein d'un même cluster nous permettent de classer les clusters par ordre de priorité. Ainsi, lorsque plusieurs clusters appar-

Algorithme 1 : *similVar* : Algorithme de classification des variations

```
1 Function similVar ( $\{\{A_1\}, \{A_2\}, \dots, \{A_n\}\}, q)$ 
2   for  $t = 1$  to  $q - 1$  do
3     Regrouper tous les attributs de la classe  $C_t$  avec la même variation dans les
      classes  $V_k$   $k \in \llbracket 1 ; p \rrbracket$ .
4     Étiqueter chaque cluster en fonction du statut de chaque attribut.
5     Déterminez les occurrences d'attributs en doublons dans  $V_k$  :  $Dup(V_k)$ .
6     Déterminez la dispersion de  $V_k$  :  $Disp(V_k)$ .
7     for  $i = 1$  to  $p$  do
8       for  $j = 1$  to  $p$  do
9         if  $(V_i \sim V_j)$  /*  $V_i$  similaire à  $V_j$  */
10        then
11          if  $Dup(V_i) > Dup(V_j)$  then
12             $V_j \leftarrow \emptyset$ 
13          end
14          if  $Dup(V_i) < Dup(V_j)$  then
15             $V_i \leftarrow \emptyset$ 
16          end
17          if  $Dup(V_i) = Dup(V_j)$  /* Comparaison de la
            dispersion lorsqu'il a le même nombre de
            doublons */
18          then
19            if  $Disp(V_i) < Disp(V_j)$  then
20               $V_j \leftarrow \emptyset$ 
21            end
22            else
23               $V_i \leftarrow \emptyset$ 
24            end
25          end
26        end
27      end
28    end
29    retourner  $\{V_1, V_2, \dots, V_p\}$ .
30  end
31 end
```

tiennent à un même intervalle de variation (clusters similaires), le meilleur candidat est choisi en fonction du plus grand nombre de doublons et de la plus grande homogénéité.

Algorithme *similVar* : intérêt et motivation Notre algorithme *similVar* vient pallier les lacunes observées sur les méthodes de clusterisation présentées dans la Section 3.5 du Chapitre 3.

Les principales limites concernent : la détermination du nombre de clusters, la sensibilité des algorithmes au paramétrage et enfin la complexité temporelle. L'algorithme *similVar* qui produit un nombre de clusters en fonction de règles (variation de valeurs sur une période) ne nécessite aucun paramétrage et évite la contrainte liée à la détermination du nombre de clusters. La complexité temporelle égale à $O(n^2)$ est aussi en faveur de notre algorithme sachant que le nombre de clusters n est souvent faible en comparaison du nombre élevé d'itérations qu'exigent les algorithmes de clusterisation. La détermination des interactions entre attributs que nous venons d'explicitier concerne celles qui appartiennent à un même cycle. Au-delà de cette considération, la question sur les conséquences d'une interaction dans le temps reste posée. Autrement dit, est-il possible d'identifier l'impact de la variation d'un attribut sur un autre en décalage dans le temps. Le paragraphe qui suit expose nos propositions pour réaliser cette détection.

5.2.4 *similVar+* : Clusterisation des données pour évaluer les interactions entre agents en décalage dans le temps

La stratégie choisie pour déterminer les interactions entre les attributs avec un éventuel décalage dans le temps s'articule toujours selon le principe d'une classification sous forme de clusters. Cependant l'indexation de la base de cas (cycle puis variation des attributs) est inversée. Il s'agit d'une généralisation de notre recherche de possibles interactions entre les attributs des différents agents. Le fait de ne plus contraindre la recherche d'interactions par cycle, nous conduit à étendre l'hypothèse suivante : si dans un même cycle, une même variation apparaît, alors cela peut indiquer un comportement similaire. C'est pourquoi, nous avons enrichi notre algorithme 1 de deux hypothèses. La première nommée H détermine un sens dans la variation : positif lorsqu'elle augmente et négatif lorsqu'elle diminue. La seconde nommée H' introduit l'intensité de la variation. Ces deux hypothèses sont définies comme suit :

Soit la variation V de sens S et d'intensité I d'un attribut A . Soit la variation V' de sens S' et d'intensité I' d'un attribut A' .

- Si une variation V de sens S (positif ou négatif¹.) induit une variation V' de sens S' , alors une variation V de sens $-S$ peut induire une variation V' de sens $-S'$ (hypothèse H).
- De même, si une variation V d'intensité I induit une variation V' d'intensité I' , alors une variation V d'intensité kI ($k \in \mathbb{R}$) peut induire une variation V' d'intensité kI' (hypothèse H').

Ces deux hypothèses, classifient les attributs selon des types d'évolution caractérisés par un sens et une intensité sans contrainte de cycles. Cela permet d'élargir les possibilités de *similVar* conçu pour clusteriser les variations des attributs des agents par variation similaire. Ainsi, nous étendons la clusterisation de données à la détection des interactions : nous recherchons les cor-

1. Une variation est considérée comme positive lorsqu'elle croît et négative lorsqu'elle décroît

rélations entre les attributs de différents clusters selon l'intensité ou le sens de leurs variations. On peut par exemple détecter par ce procédé l'interaction entre l'attribut *noise* (bruit) de l'un des agents de l'espèce *Material* et l'attribut fatigue de l'agent *surgeon* (chirurgien); ces deux attributs variant avec une intensité similaire et de même sens. Cette approche permet non seulement de détecter des interactions mais aussi de déterminer les décalages dans le temps entre ces interactions. La systématique de ces décalages et leur récurrences sont deux éléments jugés utiles par les experts.

Présentation de l'algorithme

L'algorithme *similVar+* est une extension de l'algorithme *similVar* dans la mesure où il permet de détecter des interactions et de rapprocher des variations d'attributs en décalage dans le temps et pas uniquement sur une période donnée. Il débute par une phase d'initialisation de la classe S qui est constituée de l'ensemble des attributs A_i (de chaque agent) toutes périodes confondues (ligne 1). Cette étape est suivie d'une itération sur l'ensemble des attributs (ligne 2 à 5). Deux matrices de corrélation sont alors créées à partir de l'ensemble des clusters V_k et V_{k+1} : une matrice M selon l'hypothèse H et une matrice de M' selon l'hypothèse H' . L'algorithme retourne en ligne 6 les clusters obtenus sous la forme d'un ensemble de triplets (X_i, R_i, T_i) ainsi que les différentes matrices construites $(M_t, M'_t, t \in \llbracket 1, n \rrbracket)$. X_i représente l'ensemble des attributs résultants de règles $R_i (\{R\} \vee \{R'\} \vee \{R, R'\})$ dans un ensemble de périodes T_i .

Algorithme 2 : *similVar+* : Algorithme étendu de classification des variations

```

/* Initialisation : Clusterisation des données toutes
   périodes confondues. Tri par variation, par sens de
   variation (positif ou négatif) et par période.  $q = 1$  */
1 Initialisation :  $S \leftarrow \text{similVar}(\{A_1\}, \{A_2\}, \dots, \{A_n\}, q)$ .
2 for  $k = 1$  to  $p - 1$  do
   | /* Construction des matrices de corrélation */
3   | Construire la matrice de corrélation  $M$  selon l'hypothèse  $H$  à partir de  $V_k$  et  $V_{k+1}$ .
4   | Construire la matrice de corrélation  $M'$  selon l'hypothèse  $H'$  à partir de  $V_k$  et  $V_{k+1}$ .
5 end
6 retourner
    $E = \{(X_1, R_1, T_1), \dots, (X_n, R_n, T_n)\}, F = \{M_1, \dots, M_n\}, F' = \{M'_1, \dots, M'_n\}$ .

```

La construction des matrices de corrélation (explicitée dans la Section 6.2 du Chapitre 6) est le principal élément qui distingue l'algorithme *similVar+* de l'algorithme *similVar*. Dans le cas de l'hypothèse H , une première matrice est construite à partir des variations positives d'un jeu d'attributs $E = \{A_1, A_2, \dots, A_n\}$ et la seconde à partir des variations négatives de ce même jeu d'attributs. Concernant l'hypothèse H' , l'algorithme compare une matrice établie selon un jeu de variations d'attributs $E = \{A_1, A_2, \dots, A_n\}$ avec ce même jeu d'attributs dont les variations ont été pondérées par un coefficient k ($k \in \mathbb{R}$). La comparaison des matrices obtenues (selon les hypothèses H ou H') permet de déterminer les liaisons possibles entre les

attributs. Dans l'exemple de la Figure 5.3, nous comparons les corrélations entre les valeurs des variations positives des attributs $\{X_1, X_2, X_3, X_4\}$ (Figure 5.3a) et les valeurs des variations négatives (Figure 5.3b) de ces mêmes attributs. En admettant que 0,8 soit un seuil d'acceptation pour la corrélation entre attributs (cf Section 6.2 du Chapitre 6), seuls les attributs X_2 et X_4 évoluent de manière similaire lorsque le sens de la variation change. Il est donc possible d'émettre l'hypothèse d'une dépendance entre les deux attributs.

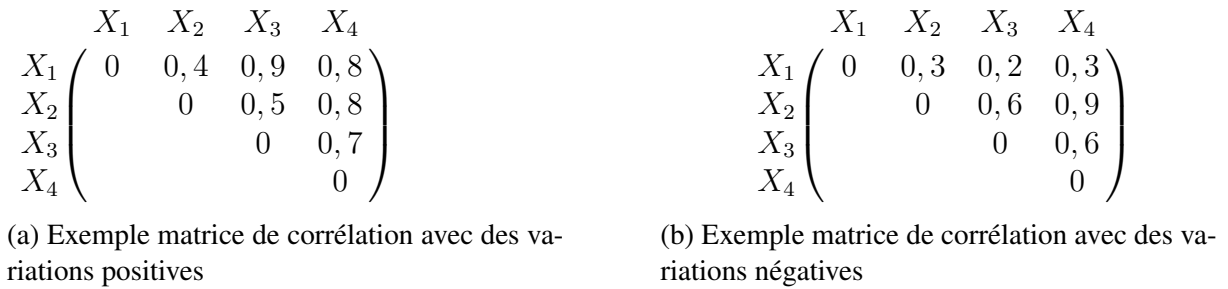


FIGURE 5.3 – Exemple matrice de corrélation avec des sens contraires de variation

Algorithme *similVar+* : intérêt et motivation

L'algorithme *similVar+* permet d'identifier des interactions possibles entre différents attributs en décalage dans le temps. L'intérêt de cet algorithme est double. Il permet d'une part de déterminer de possibles interactions entre les attributs des agents à partir des deux règles H et H' . D'autre part, il pallie la restrictivité de *similVar* où les interactions sont définies uniquement sur une même période. La Figure 5.4 est une représentation fictive de l'évolution des valeurs de deux attributs A_1 et A_2 dans l'intervalle $[0.1, 0.35]$. Celle-ci illustre l'intérêt de *similVar+* auquel a été intégré H et H' . En effet, au regard de l'évolution non monotone du nuage de points, nous pouvons conclure qu'il n'existe pas de corrélation entre A_1 et A_2 et par conséquent pas d'interactions entre eux. Autrement dit les attributs A_1 et A_2 n'évoluent pas de la même manière.

La partie suivante présente les résultats obtenus lorsque nous comparons les performances de notre algorithme *similVar* avec d'autres méthodes de clusterisation présentées dans la Section 3.5 du Chapitre 3. *similVar+* clusterise les données selon *similVar*. Les résultats de comparaisons avec les autres méthodes de clusterisation sont donc identiques à celles de *similVar* et ne sont pas traités dans ce chapitre.

5.3 Résultats

Nous proposons de comparer *similVar* avec d'autres méthodes de classification que nous avons présentées dans la Section 3.5 du Chapitre 3 : *Partition*, *Hierarchie*, *Théorie floue*, *Distribution*, *Modèles*, *Noyaux*, *Grappe spectral* et *Affinité de propagation*.

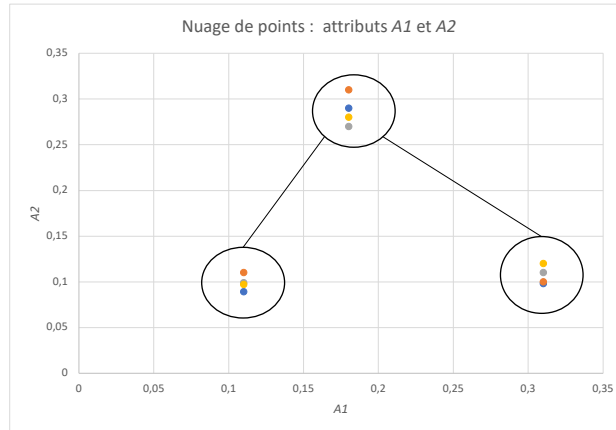


FIGURE 5.4 – Exemple nuages de points entre les attributs A_1 et A_2

Afin d'évaluer les différentes méthodes de clusterisation, nous avons effectué des simulations jusqu'à l'obtention de résultats stables : obtention d'un plateau dans les résultats. Dans le cadre de notre étude, ce nombre s'élève à 10 simulations de 150 cas. Ensuite, pour chaque cas extrait d'un ensemble de données réelles (composé de 55 cas), nous avons recherché le cluster le plus proche en nous basant sur les variations des valeurs des attributs (SpO_2 , $FeCO_2$, PNI , $fatigue$) observées entre deux cycles. Le jeu de données réelles résulte de l'extraction de données patients à partir du logiciel AMI [3] lors de quatre interventions chirurgicales. La fatigue des praticiens a, quant à elle, été déterminée avec la même fonction que celle utilisée dans nos simulations : $f(t) = \delta ae^{(k \times t)}$ (cf Section 4.4.2.4 du Chapitre 4). Enfin, les périodes d'enregistrement des données égales à 30 secondes correspondent à celles définies pour nos simulations.

Le fait de comparer des variations plutôt que des valeurs, évite les contraintes d'échelles et permet de déterminer la tendance de l'évolution (hausse ou baisse). Ce critère de comparaison est utilisé dans nos résultats présentés dans la Section 5.3.2. La Figure 5.5 est une vue des données où apparaissent les différentes variations issues d'un extrait d'une de nos simulations. Dans chacune des lignes de ce tableau, se trouve :

- “*case_attribute*” : attribut sélectionné dans la simulation ;
- “*cycle_begin*”, “*cycle_end*” : cycles d'une période ;
- “*value_begin*”, “*value_end*” : valeurs correspondant respectivement à “*cycle_begin*” et “*cycle_end*” ;
- “*interval_value*” : calcul de la différence entre “*value_end*” et “*value_begin*” ;
- “*interval_quotient*” : calcul du quotient de “*value_end*” par “*value_begin*”.

La classification en clusters est dépendante des données de notre base de cas enrichie lors de nos simulations. La Figure 5.5 décrite précédemment donne, à travers un extrait d'enregistrements, les différents éléments utilisés lors de la clusterisation. Parmi ces éléments, *interval_quotient* est le critère choisi pour le regroupement des données. Il regroupe les variations quotientées et réparties dans des intervalles qui varient selon la granularité souhaitée. Dans notre extrait

la granularité est la plus fine qui soit car elle correspond au quotient entre *case_value_end* et *case_value_begin*.

Cette étape de clusterisation est succédée par la comparaison de l'état des attributs des différents clusters (*normal, failure*) avec celui de nos données réelles. Les deux paragraphes suivants présentent la méthode utilisée et les résultats obtenus lors des comparaisons entre les différentes méthodes de clusterisation. Ces derniers sont ensuite discutés.

case_attribute	cycle_begin	variation_cases		simul_number : 1		interval_value	interval_quotient
		cycle_end	case_value_begin	case_value_end	interval_value		
temperature	5	15	37	37,4	0,400	1,011	
fatigue	5	15	1,2	1,2	0,000	1,000	
SpO2	5	15	98,2	98,3	0,100	1,001	
PNI	5	15	14,7	15	0,300	1,020	
FeCO2	5	15	45	45	0,000	1,000	
temperature	15	25	37,4	37	0,400	0,989	
fatigue	15	25	1,2	1	0,200	0,833	
SpO2	15	25	98,3	98,5	0,200	1,002	
PNI	15	25	15	14,8	0,200	0,987	
FeCO2	15	25	48	51	3,000	1,063	
temperature	25	35	37	37,4	0,400	1,011	
fatigue	25	35	1	0,9	0,100	0,900	
SpO2	25	35	54	54	0,000	1,000	
PNI	25	35	14,8	14,4	0,400	0,973	
FeCO2	25	35	97,9	97,8	0,100	0,999	
temperature	35	45	37,4	37,5	0,100	1,003	
fatigue	35	45	0,9	1,1	0,200	1,222	
SpO2	35	45	51	48	3,000	0,941	
PNI	35	45	14,4	14,7	0,300	1,021	
FeCO2	35	45	97,8	97,6	0,200	0,998	
temperature	45	55	37,5	37,6	0,100	1,003	
fatigue	45	55	1,1	1,2	0,100	1,091	
SpO2	45	55	48	45	3,000	0,938	
PNI	45	55	14,7	14,3	0,400	0,973	
FeCO2	45	55	97,6	97,1	0,500	0,995	

FIGURE 5.5 – Variations attributs entre deux cycles

5.3.1 Méthode

Quel que soit la méthode de clusterisation évaluée, nous avons opté pour un même protocole de comparaison lors de nos traitements. Les différents algorithmes de clusterisation que nous avons comparés à *similVar* ont été implémentés avec l'outil statistique "R". La Figure 5.6 donne une vision de classifications obtenues entre les cycles 15 et 25 avec la méthode de partitionnement de données *k-means*. Dans cet exemple, figurent en abscisse les intervalles des différentes variations et en ordonnées les occurrences obtenues pour chaque attribut des différents agents. Chaque cluster est étiqueté comme *normal* ou *failure* et désigne un événement indésirable associé aux soins selon la règle suivante :

$$cluster_state = \begin{cases} normal & si \frac{\sum_{i=0}^p n_i}{\sum_{j=0}^q f_j + \sum_{i=0}^p n_i} \geq \beta \\ failure & dans \text{ le cas contraire} \end{cases}$$

L'état du cluster (*cluster_state*) dépend de la valeur du quotient avec au numérateur les occurrences sans *failure* ($\sum_{i=0}^p n_i$) et au dénominateur l'ensemble des occurrences ($\sum_{j=0}^q f_j + \sum_{i=0}^p n_i$, f_j désigne une occurrence *failure*). La comparaison du quotient avec une constante β offre de nombreuses possibilités dans la détermination de l'état des clusters (*normal*, *failure*). Ainsi, par exemple, il est tout à fait possible d'étiqueter un cluster avec l'état *failure* lorsque seuls 30 % des cas possèdent cet état.

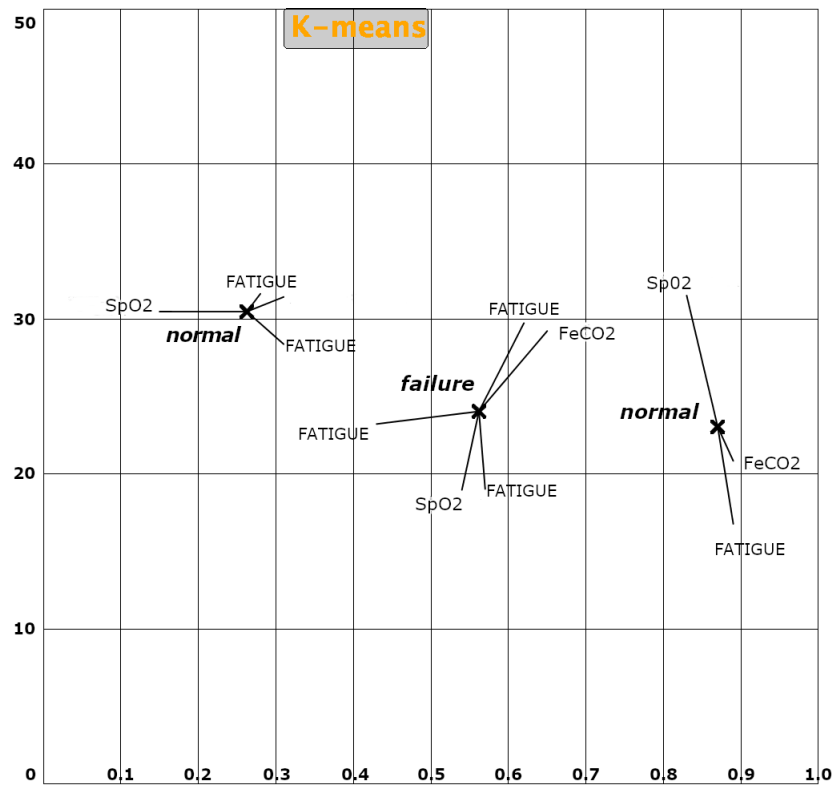


FIGURE 5.6 – Représentation de 3 clusters avec la méthode des *k-means*.

Après cette étape de détermination des clusters et de leur étiquetage, nous avons comparé chaque cas de notre jeu de données réelles avec le cluster le plus analogue. Nous rappelons qu'une approche par analogie propre à la phase de "remémoration" du cycle RàPC consiste à rechercher la plus forte similitude entre un cas cible et un cas source. Dans notre contexte

notre choix s’est porté sur un calcul de distance (définis selon nos règles : cf Section 4.5.2 du Chapitre 4) entre un cas réel et un ensemble de données de même nature (cluster). Les cas sources éligibles sont les cas les plus similaires, donc ceux minimisant cette distance. Les résultats obtenus sont présentés dans le paragraphe suivant.

5.3.2 Présentation des résultats

Les Figures 5.7 et 5.8 présentent, à titre d’exemples, respectivement nos résultats extraits d’une simulation puis de dix simulations avec la méthode *similVar*. Pour chaque *cycle*, l’état d’un cas réel (*state_case* d’un *case_number*) est comparé à celui d’un cluster (*state_cluster* d’un *cluster_number*). Lorsque l’égalité est vérifiée (*state_case = state_number*), la valeur 1 est assignée à *result_similarity*. La valeur 0 est affectée dans le cas contraire. Le résultat final (*Result*) donne la proportion du nombre de cas analogues (*state_case = state_cluster*) par rapport à l’ensemble des cas issus d’une seule simulation (cf Figure 5.7) ou de dix simulations (cf Figure 5.8). La méthode de calcul présentée pour comparer l’algorithme *similVar* avec un jeu de données réelles est aussi appliquée à l’ensemble des méthodes de clusterisation. Cela évite les biais imputables à la variabilité des méthodes.

Ce premier travail d’évaluation du calcul de similarité pour chacune des méthodes de clusterisation nous a permis ensuite d’établir un comparatif entre elles. Les résultats des différentes méthodes de clusterisation sont présentés dans le tableau 5.1 et dans les figures 5.9 et 5.10. Le tableau récapitule l’ensemble des résultats observés après : 2 simulations, 4 simulations, 6 simulations, 8 simulations et 10 simulations. Chaque valeur de la colonne “Résultat” indique en pourcentage le nombre de cas qui ont le même statut (*normal*, *EIAS*) que celui des cas réels comparés. Ce pourcentage est noté *S* dans la Figure 5.9. Les valeurs de la colonne “Disp” indiquent la dispersion relative de l’ensemble des valeurs prises lors des simulations observées : $Disp = \frac{\sigma}{\bar{x}}$ où σ représente l’écart type de la série et \bar{x} sa moyenne. Cela nous permet d’estimer l’impact des cycles sur nos résultats. Les graphiques utilisent les mêmes données que le Tableau 5.1. L’axe des *x* indique le nombre de simulations dans les deux graphiques. L’axe des *y* de la Figure 5.9 indique la valeur de *S* et l’axe des *y* de la Figure 5.10 indique la valeur de la dispersion relative observée lors des simulations. L’objectif de ces graphiques est de mettre en évidence des tendances par rapport au type de regroupement, que ce soient en termes de résultats ou de dispersion. Les courbes obtenues à partir de la Figure 5.9 montrent que malgré la disparité entre les différentes méthodes comparées, une classification en deux groupes apparaît. Si l’on considère la comparaison par rapport à *S*, les résultats des méthodes *Partition*, *Hiérarchie*, *Foue Distribution* et *Modèles* sont compris entre 30% et 40% alors que ceux des méthodes *Noyaux*, *Graphe spectral*, *Affinité propagation* et *similVar* sont compris entre 60% et 80%. On retrouve ces deux mêmes regroupements avec la dispersion : les méthodes *Partition*, *Hiérarchie*, *Théorie floue*, *Distribution* et *Modèles* se situent entre 0,1 et 0,3 alors que celles des méthodes *Noyaux*, *Graphe spectral*, *Affinité propagation* et *similVar* se situent entre 0,4 et 0,6. Les courbes de la Figure 5.9 en relation avec les résultats *S* révèlent des tendances similaires

Simul number : 1 cycle	Data set : 1 case_number	Méthode : <i>similVar</i>		state_cluster	result similarity
		cluster_number	state_case		
15	1	2	Normal	Normal	1
25	2	4	Failure	Normal	0
35	3	8	Failure	Failure	1
45	4	10	Failure	Failure	1
55	5	13	Failure	Failure	1
65	6	17	Failure	Failure	1
75	7	20	Normal	Failure	0
85	0	24	Failure	Normal	0
95	9	26	Failure	Failure	1
105	10	30	Normal	Failure	0
115	0	33	Normal	Failure	0
125	0	36	Normal	Failure	0
135	0	39	Normal	Failure	0
145	14	42	Normal	Failure	0
155	15	44	Normal	Normal	1
165	16	47	Normal	Normal	1
175	0	51	Normal	Failure	0
185	18	53	Normal	Normal	1
195	19	57	Normal	Normal	1
205	20	60	Normal	Failure	0
215	21	62	Normal	Failure	0
225	22	66	Normal	Normal	1
235	23	68	Normal	Normal	1
245	24	71	Normal	Normal	1
255	0	75	Normal	Failure	0
265	0	78	Normal	Normal	0
275	0	81	Normal	Normal	0
285	28	82	Normal	Normal	1
295	0	87	Normal	Normal	0
				Result	48,27586207

FIGURE 5.7 – Comparaison cas réel avec cluster sur une simulation

selon les méthodes. En effet, nous observons une croissance continue et faible pour les méthodes *Partition*, *Hiérarchie*, *Théorie floue*, *Distribution* et *Modèles*. La croissance plus rapide pour les méthodes *Noyaux*, *Graphe spectral*, *Affinité propagation* et *similVar* atteint un plateau à partir de 8 simulations. Ces tendances de croissances continues nous permettent de corréler l'augmentation de S avec l'augmentation du nombre de simulations.

5.3.3 Discussion

Notre Algorithme, *similVar*, montre une réelle efficacité par rapport aux méthodes classiques de clusterisation : *Partition*, *Hiérarchie*, *Théorie floue*, *Distribution* et *Modèle*. En effet, à partir de deux simulations, nous obtenons un pourcentage satisfaisant de comparaison entre les jeux

Simul number : 10	Data set : 1	Méthode : <i>similVar</i>			
cycle	case_number	cluster_number	state_case	state_cluster	result similarity
15	1	4	Normal	Normal	1
25	2	31	Failure	Normal	0
35	3	62	Failure	Failure	1
45	4	110	Failure	Normal	0
55	5	139	Failure	Normal	0
65	6	152	Failure	Failure	1
75	7	187	Normal	Normal	1
85	8	214	Failure	Failure	1
95	9	244	Failure	Normal	0
105	10	274	Normal	Normal	1
115	11	307	Normal	Normal	1
125	12	335	Normal	Failure	0
135	13	370	Normal	Normal	1
145	14	393	Normal	Failure	0
155	15	437	Normal	Normal	1
165	16	463	Normal	Normal	1
175	17	499	Normal	Normal	1
185	18	538	Normal	failure	0
195	19	543	Normal	Normal	1
205	20	573	Normal	Normal	1
215	21	602	Normal	Normal	1
225	22	633	Normal	Normal	1
235	23	677	Normal	Normal	1
245	24	711	Normal	Normal	1
255	25	729	Normal	Normal	1
265	26	769	Normal	Normal	1
275	27	796	Normal	Normal	1
285	28	827	Normal	Normal	1
295	29	844	Normal	Normal	1
				Result	75,86206897

FIGURE 5.8 – Comparaison cas réel avec cluster sur dix simulations

de données simulés et le jeu de données réel, supérieur à 65%. Ces valeurs augmentent ensuite et avoisinent 80% après 10 simulations. Comparativement, les meilleurs résultats des méthodes classiques ne dépassent pas 39%. Les méthodes récentes que nous avons comparées (*Noyaux*, *Graphe spectral*, *Affinité propagation*) sont en revanche plus efficaces que les méthodes classiques et se rapprochent des valeurs de notre algorithme *similVar* même si elles restent légèrement inférieures. Tous les résultats qui placent notre algorithme en première place, nous incitent d'une part à l'intégrer dans notre implémentation et d'autre part à engager des pistes d'amélioration pour augmenter les performances par rapport à des méthodes récentes très proches en termes de résultats. Les premiers résultats spécifiques à l'Algorithme *similVar* permettent également de mesurer, d'une part, l'impact de l'apprentissage en nombre de simulations, et d'autre part, l'influence de la position dans le temps (nombre de cycles) sur l'évolution de l'état du

Méthode	2 simul		4 simul		6 simul		8 simul		10 simul	
	Résultat	Disp	Résultat	Disp	Résultat	Disp	Résultat	Disp	Résultat	Dev
<i>similVar</i>	65.99	0.46	72.24	0.51	76.55	0.39	79.69	0.34	79.87	0.51
<i>Partition</i>	32.14	0.27	33.93	0.21	35.12	0.15	37.50	0.18	37.50	0.18
<i>Hiérarchie</i>	33.14	0.28	34.93	0.22	36.12	0.15	37.31	0.18	38.5	0.2
<i>Théorie floue</i>	32.24	0.27	34.03	0.21	35.22	0.15	36.41	0.18	37.6	0.19
<i>Distribution</i>	32.74	0.28	34.53	0.21	35.72	0.15	36.91	0.18	36.91	0.18
<i>Modèles</i>	33.34	0.28	35.13	0.22	36.32	0.16	37.51	0.19	38.7	0.2
<i>Noyaux</i>	64.19	0.45	71.44	0.50	74.75	0.38	78.07	0.50	78.07	0.50
<i>Graphe spectral</i>	64.79	0.45	72.04	0.50	75.35	0.38	78.49	0.33	78.67	0.50
<i>Affinité propagation</i>	63.49	0.44	70.74	0.49	74.05	0.38	77.37	0.49	77.37	0.49

Tableau 5.1 – Comparaison des méthodes de clusterisation

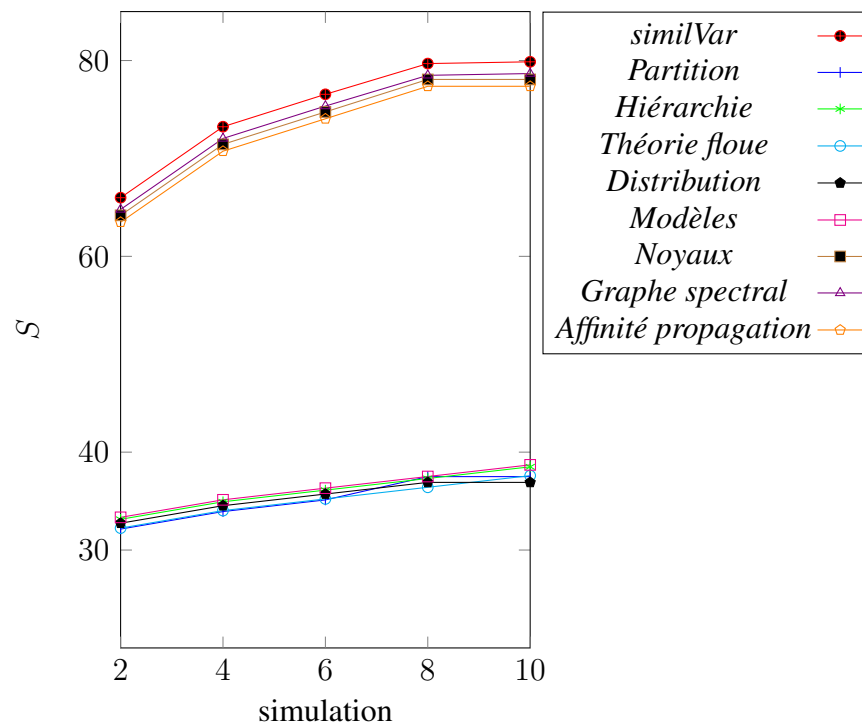


FIGURE 5.9 – Comparaison des méthodes de clusterisation

système. Nous pouvons observer une corrélation entre l'augmentation du pourcentage S et le nombre de simulations. La robustesse de l'apprentissage, caractérisée par le nombre de simulations, est donc un élément clé dans la recherche de similitude entre le statut d'un cas réel et celui issu d'un cluster. Cette fiabilité est essentielle pour l'analyse post-traitement des interactions entre les agents du cluster sélectionné. Concernant les valeurs de la dispersion relative, la non continuité des courbes au cours des simulations (cf Figure 5.10) nous montre que le nombre de cycles ne semble pas influencer les résultats.

L'ensemble de ces résultats, nous confortent donc dans l'implémentation de l'algorithme

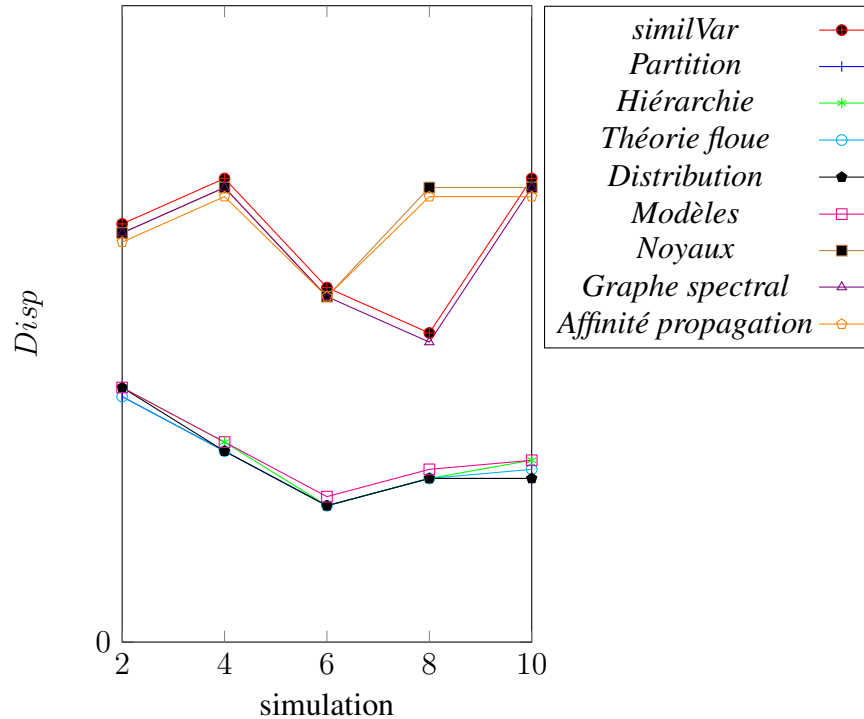


FIGURE 5.10 – Comparaison des dispersions

similVar et de *similVar+* qui intègre *similVar* sous la forme d'un appel de fonction dans *similVar+*. Concernant les temps d'exécution en mode simulation, ils sont de l'ordre de la seconde (6551 ms par simulation) et ne constituent donc pas un frein à notre étude. En revanche, c'est un paramètre qui peut-être influent en mode connecté et sur lequel nous portons toute notre attention. Outre l'intérêt de la clusterisation dans l'analyse des interactions (développé dans la Chapitre 6) nous pensons que c'est une approche intéressante pour la recherche de cas similaires propre à l'étape de "remémoration" du cycle RàPC. En effet une approche globale dans la recherche de l'état du système nous semble plus appropriée qu'une recherche de cas isolé. Cela est lié au fait qu'une tendance est souvent rattachée à un ensemble d'événements multifactoriels. Ainsi, une alerte peut résulter soit d'un dépassement de seuil (cas d'un dépassement de seuil individuel de l'attribut d'un agent) soit d'une analogie avec un cluster. Dans ce cas, un état du système considéré comme *normal* peut-être analogue (après l'étape de remémoration du cycle RàPC) à un cluster pourtant étiqueté comme *failure* et générer en conséquence une alerte. Ces comparaisons avec les différentes méthodes de clusterisation ont révélé l'intérêt de nos algorithmes en termes de résultats obtenus.

5.4 Bilan du chapitre

Nous avons développé à travers ce chapitre, l'aspect formel et opérationnel de notre seconde contribution. Ainsi nous avons explicité et motivé notre choix en termes d'outils d'aide à la décision basé sur la clusterisation des données. Ce choix, nous le rappelons, découle de l'hypothèse suivante : des phénomènes émergents peuvent résulter d'interactions communes (identifiables à partir de variations classées selon leurs similitudes). Dès lors un événement indésirable n'est pas forcément lié à un dépassement de seuil de l'attribut d'un agent, mais peut aussi être la résultante d'interactions entre agents (l'attribut *noise* de l'agent *Material* amplifie l'attribut *anxious* de l'agent *Personnal*). Il s'agit alors de facteurs d'influence que nous pensons classifiables en cluster. A partir de cette réflexion, nous avons comparé des méthodes de clusterisation avec notre algorithme *similVar* proposé en réponse à certaines limites observées dans l'état de l'art (Section 3.5 du Chapitre 3). La comparaison entre un jeu de plusieurs simulations et un jeu de quelques données réelles (55 cas) portait sur la phase de "remémoration" (recherche d'un cas source analogue à un cas cible). Les résultats en faveur de notre algorithme *similVar* lors de sa comparaison avec des méthodes de clusterisation ont motivé son implémentation. Au-delà de cette contribution qui permet de dépasser la notion de seuil individuel et donc de recherche de similitude avec un seul cas, nous avons souhaité analyser ces interactions et prédire l'évolution du système. Ces deux thèmes sont au cœur du chapitre suivant.

Chapitre 6

Analyse des interactions entre agents et prédictivité de l'évolution de l'état du système

Sommaire

6.1	Introduction	110
6.2	Évaluation de la plausibilité des interactions entre les attributs	110
6.2.1	Matrice de corrélation : matrice d'adjacence	111
6.2.2	Détermination des liaisons entre attributs à partir de la matrice d'adjacence	113
6.3	Prédictivité	114
6.4	Résultats	117
6.4.1	Plausibilité des influences résultant d'interactions entre agents	118
6.4.2	Prédictivité de l'évolution de l'état du système	125
6.5	Bilan du chapitre	127

6.1 Introduction

Nous avons explicité dans les Chapitres 4 et 5 de cette seconde partie deux de nos contributions. La première, matérialisée par une architecture SMA/RàPC, a permis de révéler des phénomènes émergents et des EIAS (à l'aide d'un SMA) tout en capitalisant des expériences issues de nos simulations (à l'aide d'un RàPC). La seconde, axée sur la clusterisation de données, est représentée par deux algorithmes de classification : *similVar* et *similVar+*. Ces deux algorithmes visent à déterminer les interactions entre agents selon l'hypothèse : des variations semblables en termes de sens ou d'intensité entre différents attributs peuvent indiquer de possibles interactions et donc d'éventuelles influences entre agents. De plus, le fait de regrouper les données nous permet d'avoir une approche plus globale du comportement des agents. C'est un avantage lors de la recherche d'analogies entre un cas cible et un cas source (phase de remémoration du cycle RàPC).

Au-delà de la notion de clusterisation, nous pensons que c'est aussi une piste intéressante en termes d'analyse des différentes interactions entre agents et des liens de causalité qui les relie à un événement indésirable. Cela nous interroge sur la possibilité de discriminer un vrai positif d'un faux positif. A titre d'exemple, l'augmentation simultanée du bruit (attribut *noise*) et de la saturation en oxygène (attribut *SpO2*) n'implique pas l'existence d'influences entre eux contrairement à l'augmentation simultanée du bruit et de l'anxiété (attribut *anxious*). Nous avons, en conséquence, révélé de possibles influences entre agents qui, nous le rappelons sont au cœur des interactions qu'elles induisent. En complément de la génération d'alerte et de l'analyse des interactions, nous présentons ici, intégré à notre système, un dernier outil d'aide à la décision permettant de prédire la survenue d'EIAS et de proposer si possible une manière de les éviter.

L'analyse des interactions et la prédictivité d'un EIAS, ainsi que les algorithmes qui en découlent sont explicités de manière formelle dans les sections 6.2 et 6.3. La section 6.4 montre leurs performances et les résultats obtenus par ces outils y sont discutés.

6.2 Évaluation de la plausibilité des interactions entre les attributs

L'hypothèse selon laquelle des variations similaires indiquent de possibles interactions entre différents attributs (variables d'état des agents) nous a conduit à clusteriser les données issues de nos simulations. Nous proposons, en complément de cet outil, une évaluation de ces interactions caractérisée par la mesure des influences entre les attributs d'un même cluster. Transposé au monde de la statistique, cela revient à déterminer les corrélations entre nos variables et donc à quantifier leurs liaisons. La relation entre deux variables peut être linéaire positive (Figure 6.1), linéaire négative (Figure 6.2), monotone non linéaire (Figure 6.3), non linéaire non monotone (Figure 6.4) ou bien encore nulle (Figure 6.5, Figure 6.6). Sachant que l'évaluation des

interactions ne se limite pas à deux variables mais à un ensemble de variables appartenant au même cluster, nous proposons une évaluation selon un cycle de deux phases qui sont explicitées dans les deux paragraphes qui suivent.

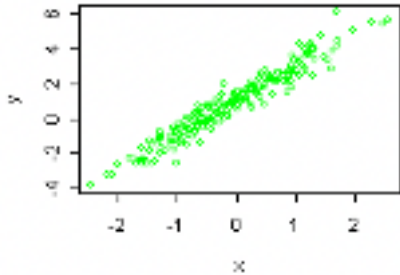


FIGURE 6.1 – Corrélation linéaire positive

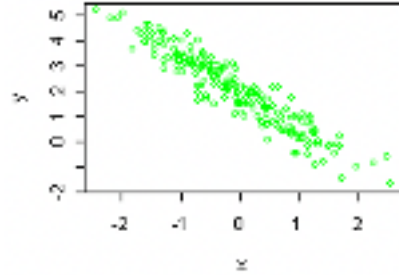


FIGURE 6.2 – Corrélation linéaire négative

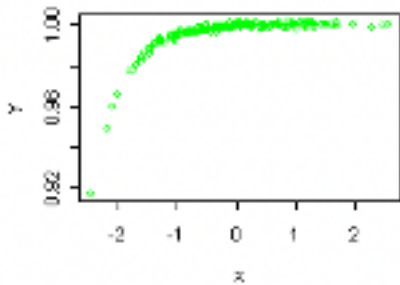


FIGURE 6.3 – Corrélation monotone non linéaire positive

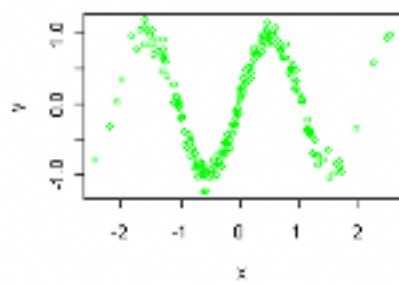


FIGURE 6.4 – Corrélation non monotone non linéaire

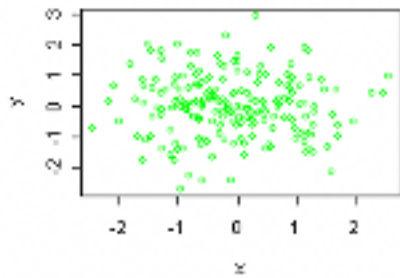


FIGURE 6.5 – Absence de corrélation : nuage de points éparses

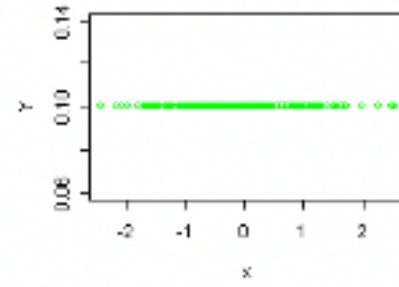


FIGURE 6.6 – Absence de corrélation : $f(x) = a$

FIGURE 6.7 – Types de corrélation entre deux variables

6.2.1 Matrice de corrélation : matrice d'adjacence

La régression linéaire entre plusieurs variables (lien linéaire entre plusieurs variables) est classiquement une généralisation de la régression linéaire à deux variables. Il s'agit de relier

une variable U à q variables V selon le modèle linéaire :

$$U = \alpha_0 + \alpha_1 V_1 + \alpha_2 V_2 + \dots + \alpha_q V_q + \epsilon$$

ϵ est la part de variabilité de U qui n'est pas expliquée, $\{\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_q\}$ sont les coefficients du modèle que l'on cherche à estimer. Celui-ci est adapté aux situations où l'on souhaite estimer une variable U en fonction de variables V_i . Ainsi, le modèle (fictif) :

$$conso = \alpha_1 prix + \alpha_2 cylind + \alpha_3 puiss + \alpha_4 poids + \epsilon$$

permet d'estimer la consommation d'un véhicule. Cette régression linéaire multiple montre cependant un intérêt limité dans notre contexte où nous cherchons à mesurer les liaisons entre plusieurs variables et non les liaisons linéaires d'une variable avec plusieurs variables. C'est la raison pour laquelle les corrélations par paires ont été choisies.

Les corrélations par paires ou *pairwise correlations* déterminent le coefficient de corrélation entre toutes les variables numériques deux à deux. On obtient alors une estimation de leurs dépendances qui s'avère utile lorsque l'on cherche à qualifier des évolutions (même sens, sens inverse, dépendantes, indépendantes). La matrice de corrélation est générée à partir des coefficients de corrélation de Spearman (cf l'équation 6.1).

$\rho(X, Y)$ détermine le coefficient de corrélation de Spearman basé sur la différence des rangs entre les observations X_i et Y_i appartenant aux distributions X et Y constituées de N observations :

$$\rho(X, Y) = 1 - \frac{6 \cdot \sum_{i=0}^N [r(X_i) - r(Y_i)]^2}{N^3 - N} \quad (6.1)$$

$r(X_i)$ est le rang de X_i , $r(Y_i)$ est le rang de Y_i et $i \in \llbracket 1, N \rrbracket$.

Cette méthode a été préférée à la corrélation de Pearson qui se limite à des relations de type affine contrairement à la corrélation de Spearman qui estime une relation entre deux variables selon une fonction monotone. L'exemple de Matrice 6.8 montre les coefficients de corrélation par paires de variables. Ainsi, on observe une relation monotone forte entre X_1 et X_3 puis entre X_2 et X_4 sachant que le coefficient r varie entre -1 et 1 : 0 indique l'absence de relation monotone entre deux variables, -1 une relation monotone négative forte entre deux variables et 1 une relation monotone positive forte entre deux variables.

La construction de la matrice de corrélation est la première étape exécutée lors de la mesure des liaisons entre variables. Cette matrice est une représentation des coefficients de corrélation calculés deux à deux pour un ensemble de variables $E = \{X_1, X_2, \dots, X_n\}$ avec $n \in \mathbb{N}$. L'étape suivante est la transformation de cette dernière en une matrice d'adjacence M d'un graphe non orienté $G(S, A)$ tel que S représente l'ensemble des sommets de G et A l'ensemble

$$\begin{array}{c}
X_1 \quad X_2 \quad X_3 \quad X_4 \\
\begin{array}{l}
X_1 \\
X_2 \\
X_3 \\
X_4
\end{array}
\begin{pmatrix}
0 & 0,2 & 0,8 & 0,3 \\
& 0 & 0,6 & 0,9 \\
& & 0 & 0,6 \\
& & & 0
\end{pmatrix}
\end{array}$$

FIGURE 6.8 – Exemple matrice de corrélation par paires de variables

des arêtes de G . Par définition, $M[i][j] = 1$ si $(a_i, a_j) \in A$ et $M[i][j] = 0$ dans le cas contraire. Sachant que les calculs des différents coefficients de corrélation ne font que tendre vers les valeurs significatives $\{-1, 0, 1\}$, à l'exclusion du cas $M[i][i] = 0$ (boucle), chacune des valeurs de M résulte de la règle :

$$M[i][j] = \begin{cases} 1 \text{ si } \begin{array}{l} \rho(X, Y) \geq \beta \wedge t < \rho(X, Y) \\ (a_i \in X, a_j \in Y) \end{array} & (t = \frac{\rho(X, Y)}{\sqrt{\frac{n-2}{1-\rho^2(X, Y)}}}) \\ 0 \text{ dans le cas contraire} \end{cases} \quad (6.2)$$

β correspond à la force de la corrélation minimale attendue et le calcul de t (qui suit la loi de Student) nous permet de tester la significativité de la corrélation avec ρ . Il s'agit plus précisément de la valeur critique égale à 5% dans la table de Student [35] pour un degré de liberté égal à $n - 2$. Cela correspond au rejet de l'hypothèse $H_0 = 0$ (ρ significativement égal à 0) pour un nombre d'observations égal à n . Nous rappelons que nous avons pris pour hypothèse qu'une liaison est assimilée à une interaction. Par conséquent la conjugaison entre deux variables d'une corrélation forte avec une significativité importante (faiblement liée au hasard) peut signifier de fortes influences entre elles. Dès lors, le contexte de notre étude suggère qu'il serait pertinent de généraliser ce modèle d'identification des corrélations. Il s'agit de rechercher les corrélations possibles sur un ensemble d'attributs et pas uniquement sur une paire d'attributs. Nous avons donc complété cette construction des matrices de corrélation par un algorithme permettant de déterminer tous les sous-graphes complets issus de la matrice d'adjacence extraite de la matrice de corrélation.

6.2.2 Détermination des liaisons entre attributs à partir de la matrice d'adjacence

Nous proposons, à partir de la matrice d'adjacence, de déterminer tous les ensembles d'attributs $E_i = \{a_p, \dots, a_q\}$ ($p, q \in \llbracket 0, n - 1 \rrbracket$) tels qu'ils soient tous liés entre eux. Dans la théorie des graphes, cela consiste à identifier tous les sous-graphes complets G_i d'un graphe G . L'algorithme 3 proposé extrait, à partir de la matrice d'adjacence, tous ces sous-ensembles

(sous-graphes complets). Il se décompose en trois étapes. L'algorithme parcourt en premier lieu (lignes 1 à 7) la matrice d'adjacence et répertorie tous les couples de sommets sans liaison ($(a_i, a_j)=0$). Il affecte ensuite \emptyset à tous les sous-ensembles qui incluent au moins un couple sans liaison (ligne 8 à 12). Enfin, il retourne (ligne 13) tous les sous-graphes complets qui correspondent aux sous-ensembles $E'_i \neq \emptyset$ ($i \in \llbracket 0, q \rrbracket$).

Algorithme 3 : liaisAttrib : Algorithme de détermination de liaisons entre attributs

Input : $E = \{a_1, \dots, a_n\}$, $F = \{E_0, \dots, E_{p-1}\}$: ensemble des sous-ensembles de E
($p = 2^n$), M matrice $n \times n$, f file *FIFO*

Output : $F' = \{E'_0, \dots, E'_q\}$: ensemble des sous-graphes complets

```

/* Recherche de couple(s) de sommet(s) sans liaison */
1 for  $i = 0$  to  $n - 1$  do
2   for  $j = 0$  to  $n - 1$  do
3     if  $M[i][j] = 0 \wedge i \neq j$  then
4       ajouter  $(a_i, a_j)$  à  $f$ 
5     end
6   end
7 end
/* Affectation  $\emptyset$  aux sous-ensembles avec couple(s) sans liaison */
8 for  $t = 0$  to  $p - 1$  do
9   if  $\exists (a_i, a_j) \in f / \{a_i, a_j\} \in E_t$  then
10     $E_t \leftarrow \emptyset$ 
11  end
12 end
/* Retour sous-graphes complets (sous-ensembles  $E'_i \neq \emptyset$ ,  $i \in \llbracket 0, q \rrbracket$ ) */
13 retourner  $F' = \{E'_0, E'_2, \dots, E'_q\}$ 

```

Cet algorithme est un outil d'aide à la décision pour la validation d'interactions entre attributs suggérées par nos outils de clusterisation (Algorithmes *similVar* et *similVar+*). Il n'en reste pas moins que la validation finale appartient toujours à l'expert. Au-delà de la détermination des ensembles d'attributs liés entre eux, nous proposons un outil d'aide à la décision destiné à prédire dans le temps la survenue d'un EIAS. Cet outil est présenté dans la section qui suit.

6.3 Prédictivité

Le principe de la prédictivité consiste à déterminer les différents états du système dans le futur en se basant sur les expériences passées et sur les relations qui peuvent exister entre elles. Nous essayons donc d'établir le lien le plus probable entre deux clusters appartenant à deux

cycles consécutifs, afin de prédire l'évolution du système la plus probable entre deux cycles. Ce lien est déterminé selon la règle R suivante :

$$R : cr_l = \underset{cr_j \in E}{argmax} | cr_j \cap cr_t |, E = \{cr_1, \dots, cr_p\} \quad (6.3)$$

Cette règle nous permet de déterminer le cluster cr_j qui possède le plus d'attributs communs avec le cluster initial cr_t . Celui-ci est alors affecté au cluster cr_l avec l qui désigne la période parcourue appartenant à $\llbracket t + 1, n \rrbracket$.

La construction de ces liens et le graphe qui en résulte dépend largement d'une structure qui permet de retracer les données. C'est pourquoi nous avons construit et indexé notre base de cas selon ce principe. Le graphe orienté et non connexe de la Figure 6.9 relie les centroïdes (centres de gravité) des différents clusters sur deux périodes (périodes c_0-c_1 à c_1-c_2 et c_1-c_2 à c_2-c_3). Chaque arête est sortante sur une période c_l-c_{l+1} ($l \in \llbracket 1 ; p \rrbracket$) et incidente sur la période suivante. Notons que le demi-degré intérieur (demi-degré entrant) d'un sommet peut être supérieur à 1. En effet, l'état prédit peut correspondre à deux états courants différents.

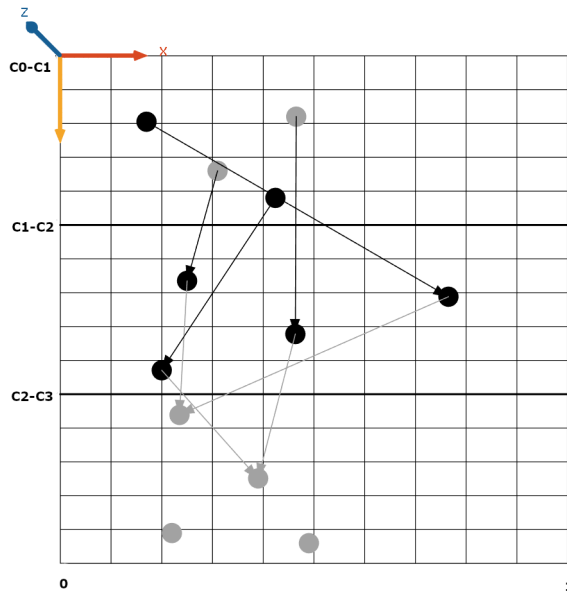


FIGURE 6.9 – Graphe des liaisons entre clusters

L'algorithme 4 que nous avons implémenté, se décompose en deux étapes :

- Initialisation à ligne 1 : Sélection du cluster initial cr_t (étape remémoration du cycle RàPC). t est la période initiale.
- Itération de la ligne 2 à la ligne 5 : Itérer de la période $t + 1$ à l'avant dernière période n . Rechercher le cluster cr selon la règle R (6.3). Affecter cr à cp_k ($k \in \llbracket t + 1 ; n \rrbracket$).

Le dernier outil d'aide à la décision présenté dans ce chapitre suggère un parcours limitant les risques de survenue d'un incident. Il s'agit de rechercher à travers notre base d'expériences,

Algorithme 4 : predict : Algorithme de prédictivité des variations

```
1 Initialisation :  $t$  (période initiale  $t \in \llbracket 0 ; n - 1 \rrbracket$ ),  $cr_t$  (cluster de la période initiale  $t$ )
2 for  $k = t + 1$  to  $n$  do
3   | Rechercher le cluster  $cr$  selon critère  $R$  (6.3)
4   |  $cp_k \leftarrow cr$  : /* Cluster prédit */
5 end
6 retourner  $\{cp_k, cp_{k+1}, \dots, cp_n\}$ .
```

une évolution de l'état de notre système ne conduisant pas à un EIAS. A cet éventuel parcours sans incident, nous ajoutons la notion de "plus court chemin". Cela est motivé par le souhait de minimiser les actions permettant d'éviter un EIAS, afin de modifier au minimum l'état du système pour le maintenir sans EIAS. L'algorithme de cet outil se fonde sur le calcul du plus court chemin conduisant à un état donné (sans défaillance dans le cas de notre étude). Le cluster est représenté par un centroïde choisi selon la règle R' se substituant à la règle R . Ainsi, il ne s'agit plus de rechercher le cluster suivant selon "l'évolution probable de l'état du système", mais selon la recherche d'un nouvel état C (état *normal*). Celle-ci est conditionnée par la règle R' :

$$R' : \arg \min dist(x_k, y_j). y_j \neq x_k, y_j \in \llbracket 0 ; m \rrbracket, \text{période}(y_j) = k, e_j = C. \quad (6.4)$$

On recherche donc parmi les y_j centroïdes appartenant à la période courante k ($k \in \llbracket t + 1 ; n \rrbracket$) celui qui est le plus proche du centroïde prédit x_k tout en respectant la condition : $e_j = C$ (état non défaillant dans notre contexte). Tout comme l'algorithme 4, l'algorithme 5 se décline en deux étapes :

- Initialisation à la ligne 1 : Sélection du cluster initial cr_t (étape remémoration du cycle RàPC) et calcul de son centroïde x_t . t est la période initiale et e_t est le statut du cluster initial.
- Itération de la ligne 2 à la ligne 10 : Itérer de la période initiale t à l'avant dernière période $n-1$. Rechercher le cluster cr selon la règle R' . Affecter cr à nc_k ($k \in \llbracket t+1 ; n \rrbracket$) si R' est respectée.

Dans la Figure 6.10, les clusters de couleurs claires indiquent les situations non défaillantes par opposition aux clusters de couleurs foncées. L'arc épais de couleur bleue qui apparaît en période c_1 - c_2 correspond au chemin le plus court menant à une situation non défaillante, à contrario de l'arc original moins épais et de couleur foncé. Sur cet exemple, le processus a permis de trouver la solution la moins coûteuse afin d'éviter un événement indésirable.

La première partie de ce chapitre a été consacrée à la description formelle de nos deux contributions en réponse à l'analyse des influences entre agents et à la prédictivité de la survenue d'un EIAS. La seconde partie présente nos résultats obtenus lors d'évaluations : des influences détectées et de la prédictivité entre un jeu de données réelles et nos simulations.

Algorithme 5 : predictPath : Algorithme de parcours selon critère C

```
1 Initialisation :  $t$  (période initiale  $t \in \llbracket 0 ; n - 1 \rrbracket$ ),  $cr_t$  (cluster initial),  $x_t$  (centroïde du  
cluster initial),  $e_t$  (statut du clusteur initial).  
2 for  $k = t + 1$  to  $n$  do  
3    $nc_k \leftarrow nil$  /* nouveau cluster proposé */  
4   if  $e_k = \neg C$  /* état du cluster de la période suivante prédit */ then  
5     Rechercher le cluster  $cr$  selon la règle  $R'$  6.4  
6     if trouvé then  
7        $nc_k \leftarrow cr$   
8     end  
9   end  
10 end  
11 retourner  $\{nc_k, nc_{k+1}, \dots, nc_n\}$ .
```

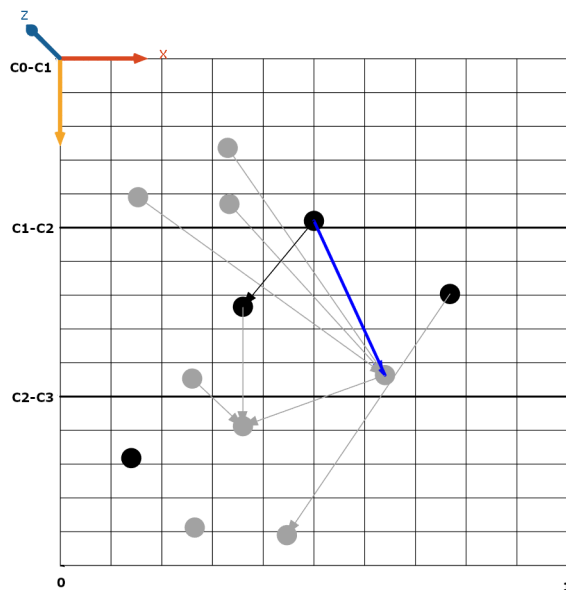


FIGURE 6.10 – Graphe du plus chemin sans incident

6.4 Résultats

Les résultats que nous proposons sont constitués de deux parties. La première évalue les possibles influences entre les attributs d'un cluster sous la forme de pourcentage. La seconde établit une comparaison de l'évolution de quelques cas réels (55 cas) avec celle prédite à partir de nos simulations. Ces deux ensembles de résultats sont présentés dans les deux sections qui suivent. Chacune d'elle est organisée de la manière suivante : présentation de la méthode utilisée, puis présentation des résultats et enfin discussion.

6.4.1 Plausibilité des influences résultant d’interactions entre agents

A partir des algorithmes *similVar*, *similVar+*, *liaisAttrib* et des données extraites de notre base de cas, nous avons construit des matrices de corrélation. Nous rappelons qu’une corrélation entre deux variables dépend des paramètres β et t (cf Section 6.2). Le choix de la valeur 0,8 pour β associée à la significativité de la corrélation ρ nous permet de ne considérer que les “fortes” corrélations. Le test de corrélation de Spearman et les valeurs de seuils que nous venons d’évoquer sont à considérer dans un contexte de simulation hors ligne où les données sont exclusivement issues de nos simulations. La modélisation en mode connecté qui permet une acquisition des données en temps réel est pour l’instant l’une de nos perspectives. Il est donc difficile d’élaborer des conjectures sans poser au préalable quelques règles. Celles-ci sont embarquées par l’agent et activées lors des interactions qui le régissent, que ce soit avec l’environnement ou avec les autres agents. Cette procédure introduit de manière partielle du déterminisme dans un contexte que nous souhaitons non déterministe, c’est pourquoi ce paramétrage n’est appliqué que dans le cadre du test de corrélation de Spearman. Nous proposons donc de générer artificiellement des interactions entre les différents attributs de l’ensemble des agents. Comme le montre la Figure 6.11 nous avons intégré sous forme de paramétrage la possibilité de créer une interaction artificielle entre eux. Il s’agit d’établir un rapport d’influence entre deux agents. Ainsi, l’attribut d’un agent peut amplifier, réduire ou faire évoluer de manière similaire l’attribut d’un autre agent lors d’une simulation. Dans notre exemple, chaque variation de l’attribut *noise* (agent source *phone*) augmente de 10% celle de l’attribut *fatigue* (agent cible *surgeon*).

Interaction	
First Agents	phone
Attribute of First Agents	noise
Interacton Type	variation
Interaction coefficient	1.1
Second Agents	surgeon
Attribute of Second Agents	fatigue

FIGURE 6.11 – Paramétrage d’une interaction

Dans la suite de ce chapitre, nous présentons les résultats obtenus à partir des attributs *SpO2*, *FeCO2* et *FC* (agent *patient*), puis de l’attribut *noise* (agent *phone*) et enfin de l’attribut *fatigue* (agent *surgeon*). Le cardinal du nombre d’interactions de ces 5 attributs est donné par la formule $C(n, p) = \frac{n!}{(n-p)!p!}$ qui dénombre le nombre de parties à p éléments parmi n éléments :

$$C(5, 2) = \frac{5!}{(5-2)!2!} = 10$$

Les résultats sont présentés sous forme de matrices de corrélation issues de 10 simulations de 305 cycles chacune : 305 cycles correspondent selon les experts à une durée moyenne d'intervention de 20 minutes, et les résultats à partir de 10 simulations sont considérés comme stables car ils ne varient plus ou très faiblement au-delà. Nous appliquons ensuite l'Algorithme *liaisAttrib* afin d'identifier tous les attributs des agents en interaction.

Présentation des résultats

Les 3 matrices qui suivent sont des exemples extraits de nos résultats qui illustrent les corrélations entre chaque attribut calculées selon la corrélation de Spearman avec $\beta = 0,8$ et avec une variation des valeurs égales à $2/100$. Nous rappelons que les variations résultent du quotient de la valeur des attributs comparée entre deux cycles, qui permet ainsi d'obtenir l'intensité de la variation et son sens (croissante si le quotient est ≥ 1 et décroissante dans le cas contraire). La première matrice (Figure 6.12) est une vision canonique des interactions entre les attributs des agents : les interactions ne sont pas paramétrées. La détermination des interactions est dans ce cas difficile à évaluer. Nous y avons adjoint la Figure 6.13 destinée à schématiser le fonctionnement de l'Algorithme *liaisAttrib*. Ainsi, les sous-graphes complets qui sont représentés, désignent deux sous-ensembles d'attributs en interaction parmi l'ensemble des sous-ensembles d'attributs en interaction : $\{\{FC, SpO2\}, \{FeCO2, noise, fatigue\}\}$.

$$M_0 = \begin{matrix} & FC & FeCO2 & SpO2 & fatigue & noise \\ \begin{matrix} FC \\ FeCO2 \\ SpO2 \\ fatigue \\ noise \end{matrix} & \begin{pmatrix} 0 & 0,414 & 1 & 0,414 & 0,612 \\ & 0 & 0,791 & 0,866 & 0,968 \\ & & 0 & 0,707 & 0,612 \\ & & & 0 & 0,866 \\ & & & & 0 \end{pmatrix} \end{matrix}$$

Interactions : $\{\{\emptyset\}\}$

FIGURE 6.12 – Matrice canonique sans paramétrage

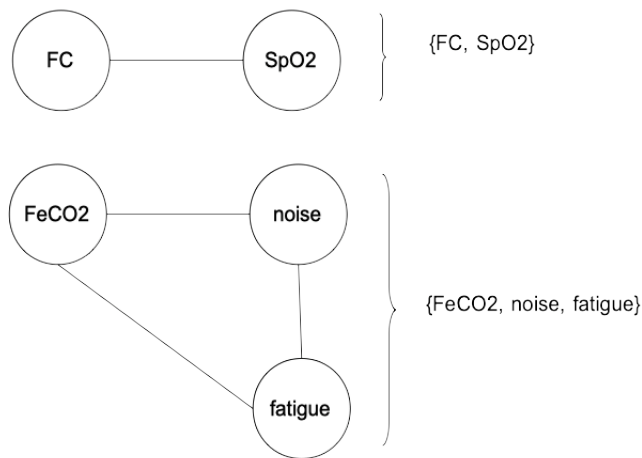


FIGURE 6.13 – Sous-graphes complets issus des interactions entre agents

Les Figures 6.14 et 6.15 correspondent aux matrices avec des interactions paramétrées : FC en interaction avec $FeCO2$ pour la matrice M_1 et FC en interaction avec $SpO2$ pour la matrice M_2 . Nous observons que dans les deux cas les interactions paramétrées sont incluses dans les sous-ensembles d'interactions construites à partir de notre Algorithme *liaisAttrib* et sont donc détectées par celui-ci.

Une vision synthétique des résultats générés par notre Algorithme *liaisAttrib* est donnée dans les Tableaux 6.1 et 6.2. Chacune des lignes indique respectivement : le nom de la matrice, les interactions paramétrées, les interactions proposées par l'algorithme ainsi que son évaluation (notée E) selon deux pas de variation (3/100 et 2/100). Cette évaluation est déterminée comme suit :

$$E = \begin{cases} 1 & \text{si } M[i][j] \geq \beta \wedge X \subset Y, (i, j) \in \llbracket 0, N - 1 \rrbracket^2 \\ 0 & \text{sinon} \end{cases}$$

$$M_1 = \begin{matrix} & FC & FeCO_2 & SpO_2 & fatigue & noise \\ \begin{matrix} FC \\ FeCO_2 \\ SpO_2 \\ fatigue \\ noise \end{matrix} & \begin{pmatrix} 0 & 0,866 & 0,612 & 0,612 & 0,612 \\ & 0 & 0,707 & 0,707 & 0,707 \\ & & 0 & 1,000 & 1,000 \\ & & & 0 & 1,000 \\ & & & & 0 \end{pmatrix} \end{matrix}$$

Interactions : $\{\{FC, FeCO_2\}, \{noise, SpO_2\}, \{SpO_2, fatigue\}\}$

FIGURE 6.14 – Interactions paramétrées $FC \leftrightarrow FeCO_2$

$$M_2 = \begin{matrix} & FC & FeCO_2 & SpO_2 & fatigue & noise \\ \begin{matrix} FC \\ FeCO_2 \\ SpO_2 \\ fatigue \\ noise \end{matrix} & \begin{pmatrix} 0 & 0,866 & 0,866 & 0,500 & 0,500 \\ & 0 & 1,000 & 0,866 & 0,866 \\ & & 0 & 0,866 & 0,866 \\ & & & 0 & 1,000 \\ & & & & 0 \end{pmatrix} \end{matrix}$$

Interactions :
 $\{\{FC, FeCO_2, SpO_2, noise\}, \{FeCO_2, SpO_2, fatigue, noise\}\}$

FIGURE 6.15 – Interactions paramétrées $FC \leftrightarrow SpO_2$

X est l'ensemble des attributs en interaction (paramétrés comme tel dans le simulateur), Y est l'ensemble des parties de l'ensemble des attributs en interaction détectés par notre Algorithme *liaisAttrib* et N correspond au cardinal de l'ensemble des attributs. L'évaluation de notre algorithme qui consiste à mesurer sa capacité à détecter des interactions, intègre deux valeurs. La première est égale à 1 lorsque les interactions paramétrées coïncident avec celle détectées par l'algorithme ; elle est égale à 0 dans le cas contraire. On obtient ainsi, d'une part une lecture simple d'une interaction entre deux attributs, d'autre part la possibilité de mesurer l'ensemble des interactions sur un intervalle donné.

Observations lorsque β est égal à 0,8

Le Tableau 6.1 montre que 100% des interactions proposées par l'Algorithme 3 coïncident avec celles qui ont été paramétrées lorsque le pas de la variation équivaut à 3/100. Ce pourcentage est en revanche très inférieur (40%) avec un pas de variation égal à 2/100. L'influence des paramétrages de β et du pas de variation mérite donc une attention particulière. Le paragraphe suivant présente une analyse de la modification de la valeur de β , désormais égale à 0,5.

Observations lorsque β est égal à 0,5

Nous observons en première lecture du Tableau 6.2 les mêmes résultats que ceux présentés dans le Tableau 6.1. Une seconde lecture nous montre cependant une augmentation significative

Matrice	Interactions Paramétrées	Interactions proposées	$E : 3/100$	$E : 2/100$
M_1	$FC - FeCO_2$	$Y_{3/100} = \{\{FC, FeCO_2\}, \{noise, SpO_2\}\}$ $\{SpO_2, fatigue\}$ $Y_{2/100} = \{\{FC, SpO_2, noise\}\}$	1	0
M_2	$FC - SpO_2$	$Y_{3/100} = \{\{FC, FeCO_2, SpO_2, noise\}\}$ $Y_{2/100} = \{\{FC, FeCO_2, SpO_2, noise\}\}$	1	1
M_3	$FC - fatigue$	$Y_{3/100} = \{\{FC, FeCO_2, fatigue\}, \{SpO_2, noise\}\}$ $Y_{2/100} = \{\{FC, FeCO_2, SpO_2, noise\}\}$	1	0
M_4	$FC - noise$	$Y_{3/100} = \{\{FC, FeCO_2, noise\}, \{FeCO_2, SpO_2, fatigue\}\}$ $Y_{2/100} = \{\{FeCO_2, SpO_2\}\}$	1	1
M_5	$FeCO_2 - SpO_2$	$Y_{3/100} = \{\{FC, fatigue, noise\}, \{FeCO_2, SpO_2\}\}$ $Y_{2/100} = \{\{FC, SpO_2, noise\}\}$	1	0
M_6	$FeCO_2 - fatigue$	$Y_{3/100} = \{\{FC, SpO_2, noise\}, \{FeCO_2, fatigue\}\}$ $Y_{2/100} = \{\{FC, SpO_2, noise\}, \{FeCO_2, fatigue\}\}$	1	1
M_7	$FeCO_2 - noise$	$Y_{3/100} = \{\{FC, SpO_2, fatigue\}, \{FeCO_2, noise\}\}$ $Y_{2/100} = \{\{FC, SpO_2, noise\}\}$	1	0
M_8	$SpO_2 - fatigue$	$Y_{3/100} = \{\{FeCO_2, SpO_2, fatigue\}, \{FC, noise\}\}$ $Y_{2/100} = \{\{FC, FeCO_2, SpO_2, noise\}\}$	1	0
M_9	$SpO_2 - noise$	$Y_{3/100} = \{\{FC, FeCO_2, fatigue, SpO_2\}, \{FeCO_2, noise, SpO_2\}\}$ $Y_{2/100} = \{\{FC, FeCO_2, fatigue, SpO_2\}, \{FC, FeCO_2, SpO_2, noise\}\}$	1	1
M_{10}	$fatigue - noise$	$Y_{3/100} = \{\{FeCO_2, fatigue, noise\}, \{FC, SpO_2\}\}$ $Y_{2/100} = \{\{FC, FeCO_2, SpO_2, noise\}\}$	1	0
Total			10	4

Tableau 6.1 – Tableau de synthèse de l'évaluation de l'Algorithme 3 : $\beta = 0,8$

Matrice	Interactions Paramétrées	Interactions proposées	$E : 3/100$	$E : 2/100$
M_1	$FC - FeCO_2$	$Y_{3/100} = \{\{FC, FeCO_2, SpO_2, fatigue, noise\}\}$ $Y_{2/100} = \{\{FC, SpO_2, fatigue, noise\}, \{FeCO_2, fatigue\}\}$	1	0
M_2	$FC - SpO_2$	$Y_{3/100} = \{\{FC, FeCO_2, SpO_2, fatigue, noise\}\}$ $Y_{2/100} = \{\{FC, FeCO_2, SpO_2, fatigue, noise\}\}$	1	1
M_3	$FC - fatigue$	$Y_{3/100} = \{\{FC, FeCO_2, SpO_2, fatigue, noise\}\}$ $Y_{2/100} = \{\{FC, SpO_2, FeCO_2, noise\}\}$	1	0
M_4	$FC - noise$	$Y_{3/100} = \{\{FC, FeCO_2, SpO_2, fatigue, noise\}\}$ $Y_{2/100} = \{\{FC, FeCO_2\}, \{FC, noise\}, \{FeCO_2, SpO_2\}\}$	1	1
M_5	$FeCO_2 - SpO_2$	$Y_{3/100} = \{\{FC, FeCO_2, SpO_2, fatigue, noise\}\}$ $Y_{2/100} = \{\{FC, SpO_2, fatigue, noise\}, \{FeCO_2, fatigue\}\}$	1	0
M_6	$FeCO_2 - fatigue$	$Y_{3/100} = \{\{FC, FeCO_2, SpO_2, fatigue, noise\}\}$ $Y_{2/100} = \{\{FC, SpO_2, noise\}, \{FeCO_2, fatigue\}\}$	1	1
M_7	$FeCO_2 - noise$	$Y_{3/100} = \{\{FC, FeCO_2, SpO_2, fatigue, noise\}\}$ $Y_{2/100} = \{\{FC, SpO_2, fatigue, noise\}, \{FeCO_2, fatigue\}\}$	1	0
M_8	$SpO_2 - fatigue$	$Y_{3/100} = \{\{FC, FeCO_2, SpO_2, fatigue, noise\}\}$ $Y_{2/100} = \{\{FC, FeCO_2, SpO_2, noise\}, \{FeCO_2, fatigue\}\}$	1	0
M_9	$SpO_2 - noise$	$Y_{3/100} = \{\{FC, FeCO_2, SpO_2, fatigue, noise\}\}$ $Y_{2/100} = \{\{FC, FeCO_2, SpO_2, fatigue, noise\}\}$	1	1
M_{10}	$fatigue - noise$	$Y_{3/100} = \{\{FC, FeCO_2, SpO_2, fatigue, noise\}\}$ $Y_{2/100} = \{\{FC, FeCO_2, SpO_2, noise\}\}$	1	1
Total			10	4

Tableau 6.2 – Tableau de synthèse de l'évaluation de l'Algorithme 3 : $\beta = 0,5$

du nombre d'éléments appartenant à Y (ensemble des sous-graphes complets). Cela révèle la cohérence des résultats en termes de corrélation : la baisse de la significativité de la corrélation (β) réduit l'exigence en termes de granularité entraînant l'augmentation du nombre d'éléments appartenant à Y . Ceci dit, le fait d'obtenir les mêmes résultats que le Tableau 6.1 nous conduit à considérer le pas de la variation comme le critère sélectif majeur. Cela s'explique en partie par la grandeur des valeurs manipulées. En effet, quotienter les variations est adapté à l'évaluation de tendance en nous exonérant des échelles, mais reste sensible à la grandeur d'un nombre. Ainsi par exemple, les quotients $1260/1325 \simeq 0,95$ et $1260/1320 \simeq 0,95$ sont plus proches que $1,5/1,7 \simeq 0,88$ et $1,5/1,6 \simeq 0,94$, alors que $1325 - 1260 \gg 1,7 - 1,5$ et $1320 - 1260 \gg 1,6 - 1,5$.

Cette sensibilité au pas de variation est discutée dans la section qui suit.

Discussion

L'outil d'aide à la décision qui découle des Algorithmes 2 (*similVar+*) et 3 (*liaisAttrib*) permet de lister de possibles interactions et donc de possibles influences entre agents. Des règles de décision peuvent alors en découler. Parmi elles, se trouvent principalement l'ajustement des seuils de risque et les préconisations résultant d'alertes. Ainsi, par exemple, l'identification d'une alerte par analogie avec un ensemble de cas (cluster) peut expliquer la survenue d'un événement indésirable consécutive à une combinaison d'attributs (*noise* et *fatigue* par exemple). Les experts peuvent dès lors soit modifier un seuil (exemple : baisse du seuil de risque de *fatigue*), soit agir sur la cause de la survenue d'EIAS (exemple : réduction des émissions sonores).

On constate à la lecture des résultats une sensibilité au pas des variations ($3/100, 2/100$). Ainsi, une réelle cohérence entre les interactions proposées par l'Algorithme *liaisAttrib* et celles que nous avons paramétrées en amont de nos simulations est observée, lorsque le pas des variations est paramétré à $3/100$ quel que soit β . En effet, les interactions sont alors détectées systématiquement par notre algorithme. Cette tendance s'amenuise quand le pas des variations et β sont respectivement paramétrées à $2/100$ et $0,8$. Le pas de variation est donc un élément majeur en termes d'analyse des résultats observés grâce au mode simulation. Qu'en est-il dans un contexte connecté (captation de données réelles)? Est-il possible d'écarter des interactions exclues en fonction de la sensibilité d'un pas? Nous pensons qu'en réponse à ces problématiques, un élargissement des critères affectés au traitement des données doit être envisagé. Cela se matérialise par la possibilité de filtrer les données selon leur sens, leurs intensités ou les deux. Le choix et le réglage de ces options découleront des résultats expérimentaux issus du contexte réel. On peut imaginer par exemple que le sens de la variation est un paramètre suffisant pour établir une corrélation entre le bruit (attribut *noise*) et la fatigue (attribut *fatigue*) sans exclure pour autant un affinement de la granularité (réduction du pas de la variation). Au vu de ces éléments, nous considérons notre algorithme de détection des interactions (Algorithme *liaisAttrib*) comme efficient mais sensible aux ajustements des variations. La section qui suit expose les résultats propres à la prédictivité de l'évolution de l'état du système.

6.4.2 Prédicativité de l'évolution de l'état du système

Afin d'évaluer notre Algorithme *predict* (prédicativité de l'évolution du système) utilisant une organisation en cluster, nous avons choisi de comparer les résultats obtenus avec ceux issus d'une prédiction basée sur un réseau de neurones artificiels. Il s'agit plus précisément d'un perceptron multi-couches qui utilise l'algorithme de rétropropagation du gradient paramétré avec : 4 couches cachées, 1000 itérations et un seuil d'erreur égal à 0,01. La couche d'entrée correspond aux valeurs de chaque attribut (*SpO2*, *FeCO2*, *PNI*, *fatigue*) des différents cas pour un cycle donné et la couche de sortie correspond à l'état prédit du système (*normal*, *failure*).

L'évaluation de la prédicativité de l'état d'un attribut (*normal*, *failure*) est donc le résultat de la comparaison de l'Algorithme *predict* avec notre RNA. Chaque jeu de simulations de 305 cas est comparé à un ensemble de 55 données réelles. Le nombre de simulations de chaque jeu de simulations varie de 2 à 20 par pas de deux. Le nombre maximum de simulations fixé à 20, correspond à un plateau et donc à des résultats considérés comme stables. Notre comparatif consiste tout d'abord à identifier les cas pour lesquels les états suggérés par l'une de nos méthodes (Algorithme *predict*, RNA) sont équivalents à ceux de l'ensemble du jeu de test (données réelles). Dans un deuxième temps, nous mesurons le nombre maximum d'occurrences de telle sorte que les comparaisons soient équivalentes sans discontinuité sur plusieurs cycles. Ce nombre est noté P . Ainsi, dans l'exemple du tableau 6.3, le résultat $P = 0,6$ correspond au maximum de 6 cas sur 10 dont les statuts de l'attribut (*normal*, *failure*) des cas comparés sont équivalents et consécutifs.

Method	Status	Status Test	Cycle
RNA	<i>normal</i>	<i>normal</i>	1
RNA	<i>normal</i>	<i>normal</i>	2
RNA	<i>normal</i>	<i>failure</i>	3
RNA	<i>normal</i>	<i>normal</i>	4
RNA	<i>normal</i>	<i>normal</i>	5
RNA	<i>failure</i>	<i>failure</i>	6
RNA	<i>failure</i>	<i>failure</i>	7
RNA	<i>normal</i>	<i>normal</i>	8
RNA	<i>normal</i>	<i>normal</i>	9
RNA	<i>normal</i>	<i>failure</i>	10

Tableau 6.3 – Exemple d'évaluation de la prédicativité

Présentation des résultats

La Figure 6.16 récapitule l'ensemble des résultats. Elle indique le nombre de simulations sur l'axe x et la valeur de P sur l'axe y . Nous pouvons voir deux évolutions différentes. La pre-

mière, celle de l'Algorithme *predict*, atteint rapidement un plateau (après 8 simulations) avec une valeur de P proche de 38 %. La seconde, celle du RNA, est progressive et semble culminer à environ 35 % après 20 simulations. Ces deux valeurs assez proches obtenues à des cycles différents sont jugées acceptables par les experts. La question du choix entre les deux méthodes doit donc être mise en balance avec la précocité des résultats considérés comme satisfaisants. Il est donc judicieux dans le cas présent d'opter pour une méthode de clusterisation plus précoce dans les résultats jugés acceptables. Cependant, la courbe RNA, moins performante au cours des 20 simulations progresse toujours au-delà de 20 simulations, même si c'est de manière faible. Le RNA est donc une alternative à notre algorithme lorsque l'apprentissage se construit sur un grand nombre de données.

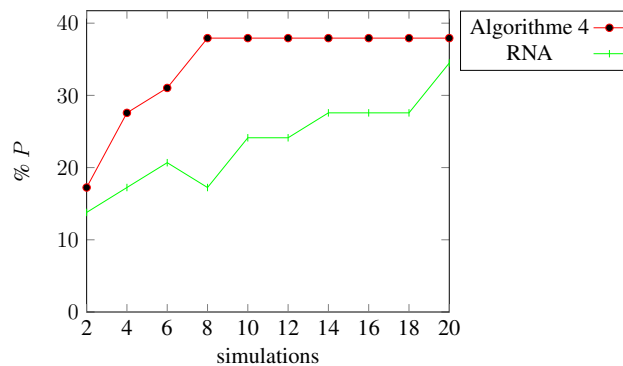


FIGURE 6.16 – Comparaison prédictivité : Algorithme *predict* et RNA

Concernant les temps d'exécution des deux méthodes, chaque jeu de simulations testé n'exécute pas 10 secondes. C'est acceptable en mode non connecté mais mérite une attention particulière en mode connecté où les données à traiter risquent d'augmenter de manière exponentielle.

Discussion

Le second outils d'aide à la décision (Algorithme *predict*) que nous avons évalué a pour objet la prédictivité de l'évolution de l'état de notre système. Nous avons expliqué dans la Section 6.3 que notre objectif consistait à construire le chemin le plus probable entre deux clusters appartenant à deux cycles consécutifs. Les résultats obtenus montrent que notre algorithme s'avère plus efficace que le RNA pour un nombre de simulations inférieur ou égal à 20. En conséquence, nous pensons que le choix d'une méthode de clustering (Algorithme *predict*) est plus adapté dans un contexte où le nombre de cas réels comparés à nos simulations est initialement faible. En effet, les ajustements de notre RNA sont alors peu significatifs, à contrario d'une base de cas. En revanche une exigence de seuil d'acceptation bien supérieure à 38% (seuil > 50%) est plutôt favorable au RNA lorsque son évolution est extrapolée. L'enrichissement de la base d'expériences et l'entraînement du réseau de neurones sont donc des facteurs déterminants dans le choix de l'une des deux méthodes.

Le dernier outil d'aide à la décision (en cours d'implémentation) est la recherche du plus

court chemin sans *failure* (Algorithme *predictPath*). Cet algorithme que nous avons décrit dans la Section 6.3 montre un intérêt comme appui décisionnel pour les praticiens, mais ouvre aussi des perspectives en termes d’auto-adaptation d’un SMA. Comme le montre la Figure 6.17 un état différent de *normal* induit une adaptation corrective du système applicable sur l’ensemble des n cycles d’une simulation. C’est une piste de réflexion en cours de formalisation.

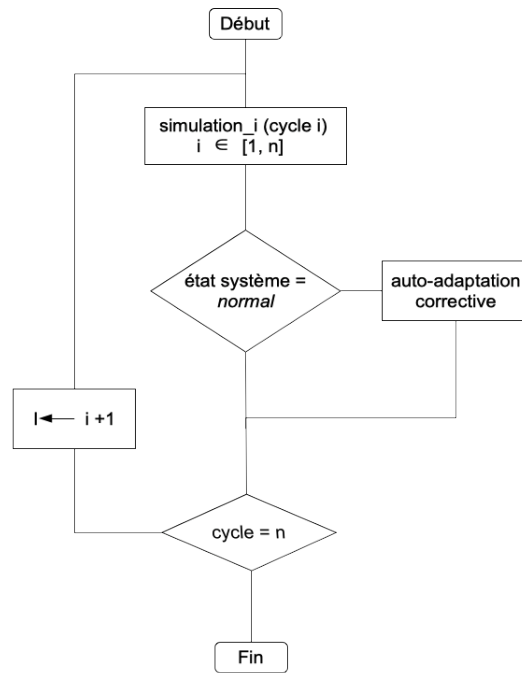


FIGURE 6.17 – Diagramme auto-adaptation SMA

6.5 Bilan du chapitre

Les deux outils d’aide à la décision que nous avons développés ont été l’objet de ce dernier chapitre. Le premier est caractérisé par la mise en évidence de facteurs d’influence à travers la reconnaissance d’attributs en interactions “fortes”. Ainsi, l’Algorithme *liaisAttrib* que nous avons implémenté regroupe en clusters les attributs dont les variations suivent les mêmes tendances en termes d’intensité et de variation (croissance, décroissance). Les résultats très satisfaisants que nous avons obtenus sont conformes à nos attentes mais dépendent du paramétrage du pas de variation et de celui de β . Il convient donc de prendre en compte la sensibilité à ces deux paramètres dans un contexte “non connecté” (données issues uniquement de simulations et non de capteurs). La piste de l’élargissement des paramètres (filtrage des données selon leur sens, leurs intensités ou les deux) est la piste de travail envisagée.

Le second outil d'aide à la décision est quant à lui destiné à proposer l'évolution prédictive de l'état du système. L'Algorithme *predict* a été comparé à un réseau de neurones artificiels. Les résultats révèlent que le nombre de données utilisées dans nos tests joue un rôle majeur dans le choix de la méthode : un faible jeu de données réelles à comparer est favorable à notre algorithme, à contrario d'un grand nombre de données. Le contexte actuel joue donc en faveur de notre algorithme.

Chapitre 7

Conclusion

7.1 Contexte

Parmi les préoccupations majeures qui accaparent les différents acteurs d'un bloc opératoire figurent l'optimisation de la sécurité du patient et par conséquent la réduction du risque de survenue d'un événement indésirable grave lors d'une intervention chirurgicale. De nombreux protocoles tel que la check-list ainsi que d'importants outils de contrôles (monitoring, capteurs ...) existent, mais s'avèrent insuffisants dans un environnement de plus en plus complexe. En réponse à cette problématique, nous avons proposé un travail de recherche orienté sur la prédiction des risques dans un contexte non déterministe; ce travail entrainé dans le cadre du projet MASSAI destiné à réduire la survenue d'événements indésirables lors d'une intervention chirurgicale. Le mot prédiction englobe trois thématiques :

- la prévention : interface entre l'utilisateur et le système, matérialisé par un générateur d'alertes ;
- l'analyse des interactions : outil d'analyse des interactions entre agents (clusters) à partir d'expériences passées, pour d'une part interpréter la survenue d'EIAS et d'autre part les limiter (ajustement des niveaux d'alertes).
- la prédictivité : outil d'aide à la décision ayant pour objet la prévision de l'évolution du système.

Nos contributions qui sont exposées dans le paragraphe suivant sont des réponses possibles à ces trois thématiques.

7.2 Contributions

Produire des situations typées comme EIAS et non prévisibles est le premier objectif que nous avons intégré dans notre travail de recherche. En conséquence, un couplage SMA/RàPC a été implémenté dans le but : de produire des émergences, de générer des alertes et de construire une base d'expériences. Cette première contribution basée sur une modélisation SMA enrichie par un raisonnement à partir de cas est une première réponse à la détermination de l'évolution de l'état d'un système, dont on ignore la causalité de certains événements (EIAS). Le paradigme agent caractérisé par l'autonomie et l'interactivité d'entités qui simulent (entre autres) des actions physiques, biologiques ou sociales, reproduit le contexte du bloc opératoire et en particulier certaines situations non prévisibles. Le RàPC est quant à lui plutôt destiné à approcher la résolution de problèmes par analogie. Cela répond à notre contexte où il n'existe pas de relation de cause à effet prédéfinies sur l'ensemble des événements produits par le système. C'est donc par une recherche de cas analogues (éventuellement adaptés) issus d'une base d'expérience, que sont proposées des solutions (situation d'alerte ou normale).

L'analyse de la survenue des EIAS est notre second axe de recherche qui nous a conduit à expérimenter des méthodes de clusterisation de données : *Partition, Hiérarchie, Théorie floue, Distribution, Modèles, Noyaux, Graphes spectraux, Affinité de propagation*. Il s'agissait de com-

parer des méthodes classifiant nos données en fonction de leurs évolutions, avec notre propre algorithme *similVar*. Face à certaines limites observées avec ces méthodes de clusterisation, les Algorithmes *similVar* et *similVar+* (qui appelle *similVar*) sont une réponse possible au regard de nos résultats rappelés dans la section suivante. Cette stratégie de classification était motivée par l'hypothèse suggérant que des évolutions appartenant à une même classe peuvent révéler des interactions et expliquer la causalité d'émergence ou de survenue d'EIAS (exemple : l'augmentation du niveau sonore entraîne l'augmentation de la fatigue humaine). L'outil d'aide à la décision destiné à analyser ces interactions était porté par l'Algorithme *liaisiAttrib*. Ce dernier évalue de possibles interactions déterminées par clusterisation.

La prédictivité de la survenue d'EIAS était le dernier objectif de ce travail de recherche. A la problématique de la prédiction de l'évolution du système nous avons suggéré comme réponse possible, un algorithme de parcours d'une base d'expériences organisée en clusters (Algorithme *predict*). Cette contribution a été complétée par un dernier outil d'aide à la décision : l'algorithme *predictPath*. Ce dernier (en cours d'implémentation) propose le plus court chemin (valeurs d'attributs les plus facilement atteignables) permettant l'évitement d'un EIAS. Le résumé des résultats obtenus est l'objet de la section suivante.

7.3 Résultats

Afin de mesurer l'efficacité de nos trois contributions que nous venons de rappeler (générateur d'alertes, analyse d'interactions et prédictivité) nous avons comparé les données issues de nos simulations avec un jeu test de données réelles de 55 cas. Les résultats présentés dans le Chapitre 5 montrent l'intérêt de nos algorithmes de clusterisation (Algorithmes *similVar* et *similVar+*) en comparaison d'autres outils standards. Les Algorithmes *similVar* et *similVar+* ont donc été implémentés dans notre modèle.

Le principe de clusterisation a de même été répliqué dans nos algorithmes d'identification d'interactions (Algorithme *liaisiAttrib*) et de prédictivité (Algorithme *predict*). Les résultats propres à l'identification des interactions (Chapitre 6) sont très satisfaisants et valident certes le concept de l'Algorithme *liaisiAttrib*, mais sont à considérer dans un contexte où les interactions ont été paramétrées pour la circonstance.

Les résultats relatifs à la prévision de l'évolution du système (Chapitre 6) sont en faveur de notre Algorithme *predict* en comparaison du réseau de neurones utilisé. Cependant, l'extrapolation de ce dernier (évolution continue croissante) laisse espérer une amélioration significative des résultats et donc in fine une préférence pour cette méthode.

On peut considérer à la lecture de l'ensemble de nos résultats que notre architecture qui agrège les paradigmes SMA et RàPC ainsi que nos outils d'aide à la décision sont une réponse prometteuse à notre problématique : prédiction des risques dans un contexte non déterministe.

7.4 Perspectives

En termes de perspectives, l'évolution à moyen terme du projet MASSAI porte sur l'intégration de capteurs au sein du bloc opératoire. Nous espérons ainsi capitaliser nos contributions dans un mode fonctionnel réel. La Figure 7.1 donne une vue générale de cette nouvelle architecture. Le bloc opératoire est désormais équipé de capteurs connectés à notre système. Parmi les processus "Ecoule", "Analyse", "Proposition" et "Autorégulation", la singularité du modèle est surtout portée par "Ecoule" et "Autorégulation". En effet, le raisonnement à partir de cas et les outils d'aides à la décision que nous avons implémentés permettent d'analyser l'évolution du système et de proposer des alternatives à la survenue d'événements indésirables. Les processus "Ecoule" et "Autorégulation" sont l'objet des deux paragraphes qui suivent.

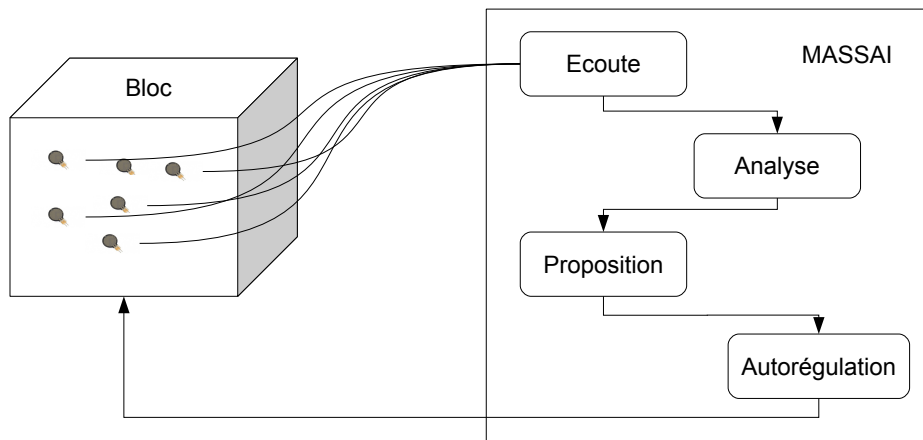


FIGURE 7.1 – Architecture de MASSAI : une version connectée de MASSAI

Le processus "Ecoule"

Le fait d'interfacer notre système avec des données réelles pourrait répondre à deux objectifs. Le premier concerne la mise en application de notre modèle en situation réelle dans le but de prévenir la survenue d'EIAS avec des outils de prédictivité et d'alertes. A cela s'ajoute la possibilité d'analyser ces EIAS. En effet, l'identification automatique des influences entre les différentes entités du bloc (agents, ressources), et les ajustements qui en découlent dans un contexte réel permettent de déterminer les causalités d'EIAS. C'est donc un environnement adapté à la mise en œuvre de notre modèle.

Le second objectif relève de l'ajustement du modèle et/ou de sa structuration. En effet, certains éléments de conception peuvent être ajustés au fil des simulations tels que les données d'entrées ou bien encore les principes de stochasticité. Au-delà de ces ajustements, le rôle et

l'influence d'un système multi-agents dans un contexte où l'on souhaite une analyse dynamique de l'évolution d'un système peut être posée. Est-ce que le paradigme SMA est à considérer comme un outil d'ajustement des seuils d'alerte grâce aux simulations produites à partir de données réelles ? Est-il plutôt un outil qui permet de prédire en simulant l'évolution de l'état du système à partir de données synchronisées avec le réel ?

Le processus "Analyse"

Le processus d'analyse actuel se compose d'outils qui déterminent entre autres la causalité de la survenue d'un événement indésirable. L'enrichissement de notre système avec des outils statistiques tels que l'inférence causale ou la propagation des incertitudes est une piste que nous envisageons. L'inférence causale qui détermine la causalité entre un événement et ses effets renforce notre recherche de causalité de survenue d'un EIAS ; la propagation des incertitudes destinée à calculer des incertitudes et leurs effets induits sur des résultats permet de mesurer l'impact d'une erreur sur notre système.

Le processus "Autorégulation"

L'autorégulation du système est une perspective clé en terme d'évolutivité du projet actuel. L'Algorithme *predictPath* qui est en cours d'implémentation propose certes l'évitement d'un événement indésirable mais pas l'autorégulation du système. Dès lors, l'autorégulation du système est une piste que nous envisageons. Il s'agit donc d'ajuster l'état du système pour éviter la survenue d'EIAS. Cela soulève cependant quelques interrogations en lien avec les principes de régulation d'un système. Est-il possible par exemple d'ignorer les problèmes de traçabilité rencontrés lors d'autorégulations ? De même, comment faut-il définir la notion de stabilité lorsque qu'une autorégulation peut être la cause d'instabilité ? Dans un tout autre registre, quelle organisation faut-il envisager : une approche micro comme actuellement qui impacte chaque agent ou plutôt macro avec une régulation globale du système ? La légitimité de l'autorégulation est très fortement liée à ces interrogations qui soulèvent chacune des pistes de recherche.

Les perspectives envisagées et les questions qu'elles soulèvent sont en cours de réflexion et à l'étude pour une suite du projet MASSAI, sachant que l'ensemble des outils d'aide à la décision qui en résultent sont des supports à la disposition des praticiens qui ne se substituent en aucun cas à leur décision.

Bibliographie

- [1] Agnar AAMODT and Enric PLAZA. Aicom-artificial intelligence communications. *IOS Press*, 7(1) :9–59, 1994.
- [2] Amjad ABOU ASSALI, Dominique LENNE, and Bruno DEBRAY. Ontologies pour l’analyse des risques industriels. *Modélisation des connaissances*, page 27, 2008.
- [3] AEGLÉ. *AEGLÉ informatique médicale*, 2017.
- [4] Rakesh AGRAWAL, Johannes GEHRKE, Dimitrios GUNOPULOS, and Prabhakar RAGHAVAN. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 94–105, 1998.
- [5] David W AHA, Leonard A BRESLOW, and Héctor MUÑOZ-AVILA. Conversational case-based reasoning. *Applied Intelligence*, 14(1) :9–32, 2001.
- [6] David W AHA and Héctor MUÑOZ-AVILA. Introduction : Interactive case-based reasoning. *Applied Intelligence*, 14(1) :7, 2001.
- [7] Negar ARMAGHAN. *Contribution à un système de retour d’expérience basé sur le raisonnement à partir de cas conversationnel : application à la gestion des pannes de machines industrielles*. PhD thesis, Université de Lorraine, 2009.
- [8] Claudio BACCIGALUPO and Enric PLAZA. Case-based sequential ordering of songs for playlist recommendation. In *European Conference on Case-Based Reasoning*, pages 286–300. Springer, 2006.
- [9] Daniel BARBARÁ and Ping CHEN. Using the fractal dimension to cluster datasets. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 260–264, 2000.
- [10] NA BASS. Emergence, hierarchies, and hyperstructures. *Artificial life*, pages 515–537, 1992.
- [11] Valentina BELLINI, Marco GUZZON, Barbara BIGLIARDI, Monica MORDONINI, Serena FILIPPELLI, and Elena BIGNAMI. Artificial intelligence : A new tool in operating room management. role of machine learning models in operating room optimization. *Journal of medical systems*, 44(1) :1–10, 2020.

- [12] Noura BENHAJJI. *Système multi-agents de pilotage réactif des parcours patients au sein des systèmes hospitaliers*. PhD thesis, Université de Lorraine, 2017.
- [13] DP Bertsekas. *Dynamic programming and optimal control, i and ii*, athena scientific, belmont, massachusetts. *New York-San Francisco-London*, 1995.
- [14] James C BEZDEK, Robert EHRLICH, and William FULL. FCM : The fuzzy c-means clustering algorithm. *Computers & geosciences*, 10(2-3) :191–203, 1984.
- [15] P BUISSON, F GUNEPIN, and M LEVADOUX. Organisation du bloc opératoire. *Support de cours Université Médicale Virtuelle Francophone*, 45pages, 2009.
- [16] OC CARRASCO. Gaussian mixture models explained : From intuition to implementation. *Retrieved*, 20 :2020, 2019.
- [17] Radovan CERVENKA and Ivan TRENCANSKY. *The Agent Modeling Language-AML : A Comprehensive Approach to Modeling Multi-Agent Systems*. Springer Science & Business Media, 2007.
- [18] Chun-Ling CHANG. Case-based reasoning support for liver disease diagnosis. *Artificial Intelligence in Medicine*, 53(1) :15–23, 2011.
- [19] Olivier CHATOR. *Système multi-agents centré sur les compétences*. PhD thesis, Université de Bordeaux, 2015.
- [20] Dorin COMANICIU and Peter MEER. Mean shift : A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5) :603–619, 2002.
- [21] Stephane COULONDRE, Olivier SIMONIN, and Jacques FERBER. Dynamo : a behavioural analysis model for multi-agent systems. In *Proceedings 1999 International Conference on Information Intelligence and Systems (Cat. No. PR00446)*, pages 614–621. IEEE, 1999.
- [22] Rajesh N DAVE and Kurra BHASWAN. Adaptive fuzzy c-shells clustering and detection of ellipses. *IEEE Transactions on Neural Networks*, 3(5) :643–662, 1992.
- [23] Gilles DEVREUX. *Le rôle des comportements informationnels dans la prise de conscience de la situation : usage dans le serious game 3D Virtual Operating Room*. PhD thesis, Université de Toulouse, 2015.
- [24] Gilles ENÉE. *Systèmes de classeurs et communication dans les systèmes multi-agents*. PhD thesis, Université Nice Sophia Antipolis, 2003.
- [25] Martin ESTER, Hans-Peter KRIEGEL, Jörg SANDER, Xiaowei XU, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.
- [26] J. FERBER. *SYSTEMES MULTI-AGENTS : Vers une intelligence collective*. InterEditions, 1995.

- [27] Jacques FERBER. Les systèmes multi-agents : un aperçu général. *Techniques et sciences informatiques*, 16(8), 1997.
- [28] Jacques FERBER, Olivier GUTKNECHT, and Fabien MICHEL. From agents to organizations : an organizational view of multi-agent systems. In *International workshop on agent-oriented software engineering*, pages 214–230. Springer, 2003.
- [29] Maurizio FILIPPONE, Francesco CAMASTRA, Francesco MASULLI, and Stefano RO-VETTA. A survey of kernel and spectral methods for clustering. *Pattern recognition*, 41(1) :176–190, 2008.
- [30] Brendan J FREY and Delbert DUECK. Clustering by passing messages between data points. *science*, 315(5814) :972–976, 2007.
- [31] Franck GECHTER and Olivier SIMONIN. Conception de sma réactifs pour la résolution de problèmes : une approche basée sur l’environnement. In *JFSMA*, pages 157–173, 2005.
- [32] Michael R GENESERETH and Nils J NILSSON. *Logical foundations of artificial intelligence*. Morgan Kaufmann, 2012.
- [33] Jean-Pierre GEORÉ, Marie-Pierre GLEIZES, and Pierre GLIZE. Conception de systèmes adaptatifs à fonctionnalité émergente : la théorie amas. *Revue d’intelligence artificielle*, 17(4) :591–626, 2003.
- [34] Nigel GILBERT and Klaus TROITZSCH. *Simulation for the social scientist*. McGraw-Hill Education (UK), 2005.
- [35] William Sealy GOSSET. Gosset, william sealy. *Encyclopedia of Mathematics*, 2016.
- [36] R GOURIVEAU, K MADJAER, E RAMASSO, and N Zerhouni. Phm-prognostics and health management-de la surveillance au pronostic de défaillances de systèmes complexes. *Techniques de l’Ingénieur*, 9, 2013.
- [37] Volker GRIMM, Uta BERGER, Finn BASTIANSEN, Sigrunn ELIASSEN, Vincent GINOT, Jarl GISKE, John GOSS-CUSTARD, Tamara GRAND, Simone K HEINZ, Geir HUSE, et al. A standard protocol for describing individual-based and agent-based models. *Ecological modelling*, 198(1-2) :115–126, 2006.
- [38] Sudipto GUHA, Rajeev RASTOGI, and Kyuseok SHIM. Cure : An efficient clustering algorithm for large databases. *ACM Sigmod record*, 27(2) :73–84, 1998.
- [39] Pari Delir HAGHIGHI, Frada BURSTEIN, Arkady ZASLAVSKY, and Paul ARBON. Development and evaluation of ontology for intelligent decision support in medical emergency management for mass gatherings. *Decision Support Systems*, 54(2) :1192–1204, 2013.
- [40] HAS. Retour d’expérience sur les événements indésirables graves associés à des soins (eigs). Technical report, HAS, 2019.
- [41] Myriam HECQUET-DEVIENNE. *Aristote. Métaphysique Gamma : Edition, traduction, études*. Éditions Peeters, Louvain-La-Neuve, 2008.

- [42] T HOET. Le bloc opératoire de demain. *Techniques hospitalières*, 54(637) :17–39, 1999.
- [43] Bryan HORLING and Victor LESSER. A survey of multi-agent organizational paradigms. *Knowledge Engineering Review*, 19(4) :281–316, 2004.
- [44] J. STUART, RUSSEL, P. NORVIG. *Artificial Intelligence : A Modern Approach*. Prentice-Hall, 1995.
- [45] Rosanne JANSSEN, Pieter SPRONCK, and Arnoud ARNTZ. Case-based reasoning for predicting the success of therapy. *Expert Systems*, 32(2) :165–177, 2015.
- [46] Nicholas R JENNINGS and Michael WOOLDRIDGE. Applying agent technology. *Applied Artificial Intelligence an International Journal*, 9(4) :357–369, 1995.
- [47] Michael JERMYN, Joannie DESROCHES, Jeanne MERCIER, Marie-Andrée TREMBLAY, Karl ST-ARNAUD, Marie-Christine GUIOT, Kevin PETRECCA, and Frederic LEBLOND. Neural networks improve brain cancer detection with raman spectroscopy in the presence of operating room light artifacts. *Journal of biomedical optics*, 21(9) :094002, 2016.
- [48] Arthur KOESTLER. Beyond atomism and holism—the concept of the holon 1. In *The Rules of the Game*, pages 233–248. Routledge, 2013.
- [49] Teuvo KOHONEN. The self-organizing map. *Proceedings of the IEEE*, 78(9) :1464–1480, 1990.
- [50] Janet L KOLODNER. Maintaining organization in a dynamic long-term memory. *Cognitive science*, 7(4) :243–280, 1983.
- [51] Marc-Olivier LABARRE. Apprentissage multi-agent. *Département d’informatique et de recherche opérationnelle, Université de Montréal, Montréal*, <http://www.emse.fr/~boissier/enseignement/sma05/exposes/marcolivier.pdf>, Québec, 2005.
- [52] Luc LAMONTAGNE and Guy LAPALME. Raisonnement à base de cas textuels—état de l’art et perspectives. *Revue de l’intelligence artificielle*, X, 2006.
- [53] J LANNELONGUE. L’asepsie au bloc opératoire. *Cahiers d’enseignement de la SOFCOT*, 73 :13–28, 2000.
- [54] Jean-Louis LE MOIGNE. *La théorie du système général : théorie de la modélisation*. FeniXX, 1994.
- [55] Jay LEE, Chao JIN, Zongchang LIU, and Hossein Davari ARDAKANI. Introduction to data-driven methodologies for prognostics and health management. In *Probabilistic prognostics and health management of energy systems*, pages 9–32. Springer, 2017.
- [56] Jean LIEBER. Contributions à la conception de systèmes de raisonnement à partir de cas. HDR, Université Henri Poincaré-Nancy I, 2008.
- [57] James MACQUEEN et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

- [58] Jitendra MALIK and Stuart RUSSELL. Measuring traffic parameters using video image processing. *Intellimotion*, 6(1) :6–13, 1997.
- [59] Florent MARIE. *COLISEUM-3D : Une plate-forme innovante pour la segmentation d'images médicales par Raisonnement à Partir de Cas (RàPC) et méthodes d'apprentissage de type Deep Learning*. PhD thesis, Université Bourgogne Franche-Comté, 2019.
- [60] Alain MILLE. Traces based reasoning (tbr) definition, illustration and echoes with story telling. *Rapport Technique RR-LIRIS-2006-002, LIRIS UMR, 5205*, 2006.
- [61] Alain MILLE. Tutorial : raisonner à partir de cas : principe, théorisation et ingénierie de la connaissance associée. *14e Atelier du Raisonnement à Partir de Cas*, 2006.
- [62] Alain MILLE, Beatrice FUCHS, and Olivier HERBEAUX. A unifying framework for adaptation in case-based reasoning. In *Workshop on Adaptation in Case-Based Reasoning, ECAI-96*, pages 22–28. Citeseer, 1996.
- [63] MF MOENS, R GERBRUERS, and C UYTTENDAELE. Salomon : Final report. Technical report, Katholieke Universiteit Leuven, 1996.
- [64] Alexandre MULLER, Adolfo CRESPO MARQUEZ, and Benoit IUNG. On the concept of e-maintenance : Review and current research. *Reliability Engineering & System Safety*, 93(8) :1165–1187, 2008.
- [65] S Issac NIWAS, R Shantha Selva KUMARI, and V SADISIVAM. Artificial neural network based automatic cardiac abnormalities classification. In *Sixth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'05)*, pages 41–46. IEEE, 2005.
- [66] O. GUTKNECHT, J. FERBER. *Un métal modèle organisationnel pour l'analyse, la conception et l'exécution de systèmes multi-agents*. Hermès, 1998.
- [67] Joseph A ORR and Dwayne R WESTENSKOW. A breathing circuit alarm system based on neural networks. *Journal of clinical monitoring*, 10(2) :101–109, 1994.
- [68] Nicolas PADOY. Machine and deep learning for workflow recognition during surgery. *Minimally Invasive Therapy & Allied Technologies*, 28(2) :82–90, 2019.
- [69] Hae-Sang PARK and Chi-Hyuck JUN. A simple and fast algorithm for k-medoids clustering. *Expert systems with applications*, 36(2) :3336–3341, 2009.
- [70] Michael G PECHT. A prognostics and health management roadmap for information and electronics-rich systems. *IEICE ESS Fundamentals Review*, 3(4) :4_25–4_32, 2010.
- [71] S PIRES, O SENECHAL, F DESCHAMPS, EFR LOURES, and MG PERRONI. Industrial maintenance for sustainable performance : A systematic literature review. In *23rd International Conference for Production Research (ICPR)*, 2015.
- [72] Albert PLA, Beatriz LÓPEZ, Pablo GAY, and Carles POUS. exit* cbr. v2 : Distributed case-based reasoning tool for medical prognosis. *Decision Support Systems*, 54(3) :1499–1510, 2013.

- [73] Carl Edward RASMUSSEN et al. The infinite gaussian mixture model. In *NIPS*, volume 12, pages 554–560, 1999.
- [74] Jean RENAUD, Eric BONJOUR, Brigitte CHEBEL-MORELLO, Béatrice FUCHS, and Nadia MATTA. *Retour et Capitalisation d'expérience. Outils et démarches*. AFNOR, 2008.
- [75] CK RIESBECK and RC SCHANK. Inside case-based reasoning lawrence erlbaum associates. *Hillsdale, NJ, US*, 1989.
- [76] Maria RIFQI. *Mesures de comparaison, typicalité et classification d'objets flous : théorie et pratique*. PhD thesis, Paris 6, 1996.
- [77] Worasak RUEANGSIRARAK, Nopasit CHAKPITAK, Komsak MEKSAMOOT, and Prapas POTHONGSUNUN. Case-based reasoning system for screening falling risk of thai elderly. In *The 4th Joint International Conference on Information and Communication Technology, Electronic and Electrical Engineering (JICTEE)*, pages 1–6. IEEE, 2014.
- [78] Rizwan Muhammad SALEEM, Aslam MUHAMMAD, and AM MARTINEZ-ENRIQUEZ. Remote patient monitoring and healthcare management using multi-agent based architecture. In *2010 Ninth Mexican International Conference on Artificial Intelligence*, pages 118–123. IEEE, 2010.
- [79] Narjès Bellamine-Ben SAOUD, Sandrine DARCY, Julie DUGDALE, Bernard PAVARD, and Mohamed Ben AHMED. Simulation multi-agents de situation de secours d'urgence. *Technologies de l'information et de la communication pour les pratiques médicales : Comptes rendus des dixièmes Journées Francophones d'Informatique Médicale, Tunis, 4 et 5 septembre 2003*, 2005.
- [80] Alexandre SARAZIN, Sébastien TRUPTIL, Aurelie MONTARNAL, and Jacques LAMOTHE. Système de systèmes dans les architectures phm. In *MOSIM'18-12ème Conférence internationale de Modélisation, Optimisation et SIMulation*, 2018.
- [81] Alsane SENE, Bernard KAMSU-FOGUEM, and Pierre RUMEAU. Telemedicine framework using case-based reasoning with evidences. *Computer methods and programs in biomedicine*, 121(1) :21–35, 2015.
- [82] Roded SHARAN and Ron SHAMIR. Click : a clustering algorithm with applications to gene expression analysis. In *Proc Int Conf Intell Syst Mol Biol*, volume 8, page 16, 2000.
- [83] T. WOLF, T. HOLVOET. *Emergence Versus Self-Organisation : Différents Concepts but Promising When Combined*. Springer, 2004.
- [84] Patrick TAILLANDIER. La modélisation du temps dans la simulation à base d'agents. *L'information géographique*, 79(2) :65–78, 2015.
- [85] Patrick TAILLANDIER, Olivier THEROND, Benoit GAUDOU, et al. Une architecture d'agent bdi basée sur la théorie des fonctions de croyance : application à la simulation du comportement des agriculteurs. *Journées Francophones sur les Systèmes Multi-Agents 2012*, pages 107–116, 2012.

- [86] Nikhil M VICHARE and Michael G PECHT. Prognostics and health management of electronics. *IEEE transactions on components and packaging technologies*, 29(1) :222–229, 2006.
- [87] Cedric Villani and Bernard Nordlinger. *Santé et intelligence artificielle*. Cnrs, 2018.
- [88] Ulrike VON LUXBURG. A tutorial on spectral clustering. *Statistics and computing*, 17(4) :395–416, 2007.
- [89] Wendell WALLACH, Colin ALLEN, and Iva SMIT. Machine morality : bottom-up and top-down approaches for modelling human moral faculties. *Ai & Society*, 22(4) :565–582, 2008.
- [90] Mei-Hui WANG, Chang-Shing LEE, Kuang-Liang HSIEH, Chin-Yuan HSU, Giovanni ACAMPORA, and Chong-Ching CHANG. Ontology-based multi-agents for intelligent healthcare applications. *Journal of Ambient Intelligence and Humanized Computing*, 1(2) :111–131, 2010.
- [91] Wei WANG, Jiong YANG, Richard MUNTZ, et al. Sting : A statistical information grid approach to spatial data mining. In *VLDB*, volume 97, pages 186–195, 1997.
- [92] Jr WARD and H JOE. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301) :236–244, 1963.
- [93] Jiuping XU, Yusheng WANG, and Lei XU. Phm-oriented sensor optimization selection based on multiobjective model for aircraft engines. *IEEE Sensors Journal*, 15(9) :4836–4844, 2015.
- [94] Jiuping XU and Lei XU. Health management based on fusion prognostics for avionics systems. *Journal of Systems Engineering and Electronics*, 22(3) :428–436, 2011.
- [95] Ren YU, Benoit IUNG, and Hervé PANETTO. A multi-agents based e-maintenance system with case-based reasoning decision support. *Engineering applications of artificial intelligence*, 16(4) :321–333, 2003.
- [96] Tian ZHANG, Raghu RAMAKRISHNAN, and Miron LIVNY. Birch : an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2) :103–114, 1996.
- [97] Abdelhamid ZOUHAIR. *Raisonnement à partir de cas dynamique multi-agents : application à un système de tuteur intelligent*. PhD thesis, Le Havre, 2014.

Table des figures

2.1	Check-list bloc opératoire [HAS 2018]	16
2.2	Circuit du bloc opératoire [15]	18
2.3	Anesthésie du patient	19
2.4	Préparation du patient	20
2.5	Intervention chirurgicale	22
3.1	Architecture PHM [80]	26
3.2	Appareil d'anesthésie	28
3.3	SMA selon Ferber [26]	33
3.4	Relation micro-macro dans les SMA [26]	36
3.5	Neurone artificiel	39
3.6	Réseau de deux neurones	40
3.7	Apprentissage d'un Perceptron	41
3.8	Carré d'analogie [62]	43
3.9	Cycle RàPC	44
3.10	Calcul de Ward	50
3.11	Distribution gaussienne [16]	51
3.12	Mappage de l'espace \mathbb{R}^2 vers l'espace caractéristique	52
3.13	Graph Clusterisation	53
4.1	Caractérisation du modèle	61
4.2	Structure du couplage SMA/RàPC	65
4.3	Définition d'un agent	66
4.4	Vision global de la structure du modèle	68
4.5	Diagramme simplifiée de l'activité "Opérer"	69
4.6	Diagramme de classe UML : Élaboration d'un cas	75
4.7	Représentation globale couplage SMA/RàPC	78
4.8	Organisation holonique du SMA	80
4.9	Intégration de bases dans le RàPC	81
4.10	Vision globale du simulateur	83
4.11	Contrôle des atributs et des agents	83

4.12	Détail de l'état du système	85
4.13	Focus sur l'état du système	85
5.1	Exemple d'influences entre un agent et une ressource	92
5.2	Clusterisation supervisée et non supervisée	94
5.3	Exemple matrice de corrélation avec des sens contraires de variation	99
5.4	Exemple nuages de points entre les attributs A_1 et A_2	100
5.5	Variations attributs entre deux cycles	101
5.6	Représentation de 3 clusters avec la méthode des <i>k-means</i>	102
5.7	Comparaison cas réel avec cluster sur une simulation	104
5.8	Comparaison cas réel avec cluster sur dix simulations	105
5.9	Comparaison des méthodes de clusterisation	106
5.10	Comparaison des dispersions	107
6.1	Corrélation linéaire positive	111
6.2	Corrélation linéaire négative	111
6.3	Corrélation monotone non linéaire positive	111
6.4	Corrélation non monotone non linéaire	111
6.5	Absence de corrélation : nuage de points épars	111
6.6	Absence de corrélation : $f(x) = a$	111
6.7	Types de corrélation entre deux variables	111
6.8	Exemple matrice de corrélation par paires de variables	113
6.9	Graphe des liaisons entre clusters	115
6.10	Graphe du plus chemin sans incident	117
6.11	Paramétrage d'une interaction	118
6.12	Matrice canonique sans paramétrage	120
6.13	Sous-graphes complets issus des interactions entre agents	120
6.14	Interactions paramétrées $FC \leftrightarrow FeCO_2$	121
6.15	Interactions paramétrées $FC \leftrightarrow SpO_2$	121
6.16	Comparaison prédictivité : Algorithme <i>predict</i> et RNA	126
6.17	Diagramme auto-adaptation SMA	127
7.1	Architecture de MASSAI : une version connectée de MASSAI	132

Liste des tableaux

3.1	Propriétés et capacités des agents	31
3.2	Organisations SMA	35
3.3	Summary of typical clustering methods	49
4.1	Variables d'état des agents simulés	72
4.2	Exemple changement d'échelle entre deux attributs	76
4.3	Exemple de phénomène émergent	86
4.4	Comparaison SMA/RàPC avec "3D Virtual Operating Room"	87
5.1	Comparaison des méthodes de clusterisation	106
6.1	Tableau de synthèse de l'évaluation de l'Algorithme 3 : $\beta = 0,8$	122
6.2	Tableau de synthèse de l'évaluation de l'Algorithme 3 : $\beta = 0,5$	123
6.3	Exemple d'évaluation de la prédictivité	125

Résumé : La gestion des risques liés aux actions et aux conditions des différents éléments qui composent un bloc opératoire est une préoccupation majeure lors d’une intervention chirurgicale. La prédiction d’événements indésirables associés aux soins (EIAS), et l’analyse de leur survenue sont des enjeux majeurs. Dans cet objectif, nous proposons dans le cadre du projet MASSAI (Modeling Aids for Safe Surgery using Artificial Intelligence) une architecture qui couple un système multi-agents (SMA) avec un raisonnement à partir de cas (RàPC). Le SMA capable de produire un grand nombre de situations où les agents interagissent entre eux créent des phénomènes émergents et donc un possible nouvel EIAS. Le couplage du système multi-agents avec un raisonnement à partir de cas, enrichit la base de connaissances du modèle, et répond à notre recherche d’approche de résolution de problèmes par analogie à partir d’expériences passées. Au-delà de la détermination d’EIAS nous avons intégré des outils d’aide à la décision. Les deux premiers algorithmes (*similVar* et *similVar+*) que nous avons conçus et comparés à d’autres méthodes analysent les données récupérées dans le bloc opératoire. Ils permettent d’établir des corrélations entre les comportements des agents. Deux autres algorithmes (*liaisAttrib* et *predictPath*) sont quant eux respectivement dédiés à la détermination de corrélations entre des données clusterisées et à la prédictivité de l’évolution du système. Les résultats encourageants montrent l’efficacité de notre approche dans la capacité à produire, à analyser et à prédire la survenue d’EIAS dans un contexte non déterministe où évoluent des entités hétérogènes.

Mot-clés : SMA, RàPC, événements indésirables associés aux soins, clusterisation, prédictivité.

Abstract : The management of risks related to the actions and conditions of the various elements that make up an operating room is a major concern during a surgical procedure. The prediction of adverse events associated with care (AEAC), and the analysis of their occurrence are major issues. In this objective, we propose in the MASSAI project (Modeling Aids for Safe Surgery using Artificial Intelligence) an architecture that couples a multi-agent system (MAS) with a case-based reasoning (CBR). The MAS is able to produce a large number of situations where agents interact with each other, thus creating emergent phenomena and a possible new AEAC. The coupling of the multi-agent system with case-based reasoning enriches the knowledge base of the model, and allows us to solve problems by analogy from past experiences. In addition to the determination of the AEAC, we have integrated decision support tools. The first two algorithms (*similVar* and *similVar+*) that we have designed and compared to other methods are oriented on data classification. They consist in clustering the data in order to establish correlations between the agents’ behaviors. Two other algorithms (*liaisAttrib* and *predictPath*) are respectively dedicated to the determination of correlations between clustered data and to the predictivity of the system evolution. The encouraging results show the efficiency of our approach in the capacity to produce, analyze and predict the occurrence of AEAC in a non-deterministic context where heterogeneous entities evolve.

Keywords : MAS, CBR, adverse events associated with care, clustering, predictivity.