THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : MSTII - Mathématiques, Sciences et technologies de l'information, Informatique
Spécialité : Informatique
Unité de recherche : Laboratoire de conception et d'intégration des systèmes

De l'organisation d'un système multi-agent de cyberdéfense

On the organisation of a multi-agent cyber defence system

Présentée par :

Julien SOULE

Direction de thèse :

Jean-Paul JAMONT PROFESSEUR DES UNIVERSITES, Université Grenoble Alpes	Directeur de thèse
Michel OCCELLO PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES	Co-directeur de thèse
Louis-Marie TRAONOUEZ DOCTEUR EN SCIENCES, THALES LAND AND AIR SYSTEMS, BU IAS	Co-encadrant de thèse

Rapporteurs :

LAURENT VERCOUTER
PROFESSEUR DES UNIVERSITES, INSA ROUEN
GAUTHIER PICARD
DIRECTEUR DE RECHERCHE, CENTRE ONERA TOULOUSE

Thèse soutenue publiquement le **17 novembre 2025**, devant le jury composé de :

AURELIE BEYNIER, PROFESSEURE DES UNIVERSITES, SORBONNE UNIVERSITE	Présidente
JEAN-PAUL JAMONT, PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES	Directeur de thèse
MICHEL OCCELLO, PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES	Co-directeur de thèse
LAURENT VERCOUTER, PROFESSEUR DES UNIVERSITES, INSA ROUEN	Rapporteur
GAUTHIER PICARD, DIRECTEUR DE RECHERCHE, CENTRE ONERA TOULOUSE	Rapporteur
LAETICIA MATIGNON, MAITRESSE DE CONFERENCES, UNIVERSITE LYON 1 - CLAUDE BERNARD	Examinatrice
OUM-EL-KHEIR AKTOUF, PROFESSEURE DES UNIVERSITES, GRENOBLE INP - UGA	Examinatrice
FLAVIEN BALBO, PROFESSEUR, ECOLE NAT SUP MINES ST ETIENNE	Examinateur

Invités :

PAUL THERON
DOCTEUR EN SCIENCES, AICA IWG
LOUIS-MARIE TRAONOUEZ
DOCTEUR EN SCIENCES, THALES LAND AND AIR SYSTEMS, BL IAS

DE L'ORGANISATION D'UN SYSTÈME MULTI-AGENT DE CYBERDÉFENSE

JULIEN SOULÉ

Docteur de Philosophie
Université Grenoble Alpes
Grenoble INP
Laboratoire de Conception et d'Intégration des Systèmes

17 Septembre 2025 – v0.1

Julien Soulé : *De l'Organisation d'un Système Multi-Agent de Cyberdéfense*, Docteur de Philosophie, © 17 Septembre 2025

ACKNOWLEDGMENTS

Pursuing a doctorate may seem like a solitary path, but I certainly didn't walk it alone.

First and foremost, I would like to thank my thesis supervisors, Jean-Paul and Michel, as well as Louis-Marie, who supported me at Thales, and Paul, who guided me throughout these three years. They guided me throughout this process, allowing me to discover what research is all about (including the mistakes), and also to intervene when necessary. They were always there for me, even during their holidays, taking the time to proofread my thesis manuscript.

A big thank you also to the members of the thesis committee. I would like to thank my thesis reviewers, Laurent Vercouter and Gauthier Picard, for their proofreading of my manuscript. I would also like to thank the examiners : Aurélie Beynier for serving as jury president, Flavien Balbo for his support within the CSI program, and Laeticia Matignon and Oum-el-Kheir Aktouf for agreeing to attend and examine my thesis defense.

I met many incredible students during my time at the lab. I would especially like to mention Arthur, Vincent, Thinh, and Karem for welcoming me. I would also like to thank Minh Tuan, Sébastien, and Maximilian for their kindness and for all the conversations we had.

I would also like to thank François Suro and Simon Gay, who regularly provided feedback and helped me stay on track when I was motivated but still lacked confidence in my work.

A big thank you to Clément Raïevsky, Laurent Lefèvre, Annabelle Mercier, and Louis Morge-Rollet for their support during my teaching experiences. Working at LCIS wouldn't have been so easy without Patricia and Carole, who helped me navigate the mountain of paperwork, and Arthur, Marie, and Nicolas, who provided me with the technical resources I needed.

I also want to thank my friends Dorian and Thomas, who are also pursuing their doctorates, and Joaquin, who has always been there to help me take a break from my studies.

And finally, I owe everything to my family, who have supported me wholeheartedly throughout this adventure. None of this would have been possible without them.

RÉSUMÉ

Face à la complexité croissante des menaces en Cybersécurité, les approches centralisées montrent leurs limites pour protéger efficacement des systèmes distribués et dynamiques. Cette thèse explore une approche distribuée fondée sur des Systèmes Multi-Agents (SMAs) capables de détecter, répondre et s'adapter collectivement à des attaques autonomes et évolutives. L'objectif central est de permettre la conception d'un SMA de cyberdéfense en trouvant un mécanisme d'organisation adapté aux contraintes des concepteurs et de l'environnement. La littérature sur le sujet met en lumière l'approche symbolique favorisant le contrôle et l'approche connexionniste favorisant la performance. Pour dépasser cette tension, la thèse propose une méthode hybride combinant un modèle organisationnel symbolique et *Multi-Agent Reinforcement Learning* (MARL).

La clé de cette méthode consiste à voir la conception d'un SMA au travers d'un *problème d'optimisation sous contraintes*, dans lequel la politique conjointe des agents est apprise tout en respectant des contraintes organisationnelles exprimant les exigences du concepteur. Cette approche requiert à la fois une modélisation fidèle de l'environnement et une capacité à analyser et contrôler les comportements obtenus. La méthode intègre les différents travaux selon quatre activités : (i) **modélisation** de l'environnement cible à l'aide de techniques manuelles ou de type *World Models*, pour obtenir une version simulée de l'environnement cible ; (ii) **entraînement** des agents via MARL, avec intégration de contraintes issues du modèle organisationnel $\mathcal{M}$OISE$^+$; (iii) **analyse** des politiques apprises, en extrayant rôles et objectifs implicites via des méthodes non supervisées sur les trajectoires ; (iv) **transfert** des résultats dans l'environnement réel, avec mise à jour continue des modèles et politiques.

Un outil logiciel a été développé pour mettre en œuvre cette méthode, et appliqué à trois cas d'usage : un essaim de drones, une infrastructure d'entreprise et une architecture de micro-services. Les résultats montrent une amélioration en termes de résilience, d'adaptabilité et d'autonomie par rapport aux approches centralisées. Enfin, cette thèse ouvre plusieurs perspectives de recherche : l'amélioration de la modélisation de l'environnement grâce à l'intégration de connaissances expertes, le renforcement de la robustesse de l'apprentissage dans des environnements dynamiques, ainsi que l'exploration des représentations latentes pour faciliter l'analyse organisationnelle.

MOTS-CLEFS : Système Multi-Agent • Cyberdéfense • Apprentissage par Reinforcement Multi-Agent • Autonomous Intelligent Cyberdefense Agent • Conception assistée

ABSTRACT

Faced with the growing complexity of cybersecurity threats, centralized approaches are showing their limitations in effectively protecting distributed and dynamic systems. This thesis explores a distributed approach based on *Multi-Agent System*s (MASs), capable of collectively detecting, responding to, and adapting to autonomous and evolving attacks. The central objective is to enable the design of a MAS for cyber defense by finding an organizational mechanism suited to the constraints of both the designers and the environment. The literature highlights the symbolic approach, which favors control, and the connectionist approach, which favors performance. To overcome this tension, the thesis proposes a hybrid method combining a symbolic organizational model with MARL.

The key idea of this method is to view the design of a MAS as a *constraint optimization problem*, in which the agents' joint policy is learned while respecting organizational constraints expressing the designer's requirements. This approach requires both a faithful modeling of the environment and the ability to analyze and control the resulting behaviors. The method integrates the various contributions into four activities : (i) **modeling** the target environment using manual techniques or *World Models*, to obtain a simulated version of the target environment ; (ii) **training** the agents via MARL, incorporating constraints derived from the organizational model $\mathcal{M}\text{OISE}^+$; (iii) **analyzing** the learned policies by extracting implicit roles and objectives through unsupervised methods applied to trajectories ; (iv) **transferring** the results to the real environment, with continuous updates of the models and policies.

A software tool was developed to implement this method and applied to three use cases : a drone swarm, an enterprise infrastructure, and a microservices architecture. The results show improvements in terms of resilience, adaptability, and autonomy compared to centralized approaches. Finally, the thesis opens several research directions : improving environment modeling through the integration of expert knowledge, strengthening the robustness of learning in dynamic environments, and exploring latent representations to facilitate organizational analysis.

KEYWORDS : Multi-Agent Systems • Cyberdefense • Multi-Agent • Autonomous Intelligent Cyberdefense Agent Reinforcement Learning • Assisted-Design

TABLE DES MATIÈRES

INTRODUCTION GENERALE 1

I CONTEXTE DE TRAVAIL 7

Introduction 9

1 REPENSER LA CYBERDÉFENSE POUR DE NOUVEAUX ENJEUX 12
- 1.1 Aperçu du domaine de la Cyberdéfense . . . 12
- 1.2 Des menaces de plus en plus autonomes et distribuées . . . 14
- 1.3 La piste d'une vision multi-agent . . . 17
- 1.4 De la conception d'un SMA de Cyberdéfense . . . 19
- 1.5 Bilan . . . 22

2 VERS DES SMAS DE CYBERDÉFENSE ET LEUR CONCEPTION 24
- 2.1 Les SMAs de Cyberdéfense dans la littérature . . . 24
  - 2.1.1 Mécanismes organisationnels dynamiques . . . 24
  - 2.1.2 Organisations des SMAs de Cyberdéfense . . . 25
  - 2.1.3 Revue critique . . . 26
- 2.2 Les cadres de conception de SMA de Cyberdéfense dans la littérature . . . 28
  - 2.2.1 État des lieux et diversité des approches . . . 28
  - 2.2.2 Couverture des critères de conception (C1 à C5) . . . 29
  - 2.2.3 Limites actuelles et besoins non adressés . . . 31
- 2.3 Une tension entre approche symbolique et connexionniste . . . 31
  - 2.3.1 Les approches symboliques . . . 32
  - 2.3.2 Les approches apprenantes . . . 32
  - 2.3.3 Une opposition structurelle révélée par les critères . . . 32
- 2.4 Bilan . . . 33

3 UN PROBLÈME D'OPTIMISATION POUR STRUCTURER UNE MÉTHODE 35
- 3.1 Formulation du problème global . . . 35
  - 3.1.1 Enjeux de conception . . . 35
  - 3.1.2 Problème d'optimisation sous contraintes . . . 36
  - 3.1.3 Spécification organisationnelle vs émergence . . . 38
- 3.2 Décomposition en sous-problèmes . . . 38
- 3.3 Hypothèses de recherche . . . 38
- 3.4 Vers une méthode de conception organisationnelle . . . 40
- 3.5 Bilan . . . 41

Conclusion 43

II ÉTAT DE L'ART 45

Introduction 47

4 LES VERROUS D'UNE MÉTHODE DE CONCEPTION 50
- 4.1 La modélisation d'un environnement en simulation (H-MOD) . . . 50
- 4.2 L'intégration de contraintes en MARL (H-TRN) . . . 55
- 4.3 L'extraction des spécifications organisationnelles émergentes (H-ANL) . . . 59
- 4.4 Le maintien de cohérence entre environnement simulé et réel (H-TRF) . . . 62
- 4.5 Bilan . . . 67

5 LES TRAVAUX ET CONCEPTS THÉORIQUES MOBILISÉS 70

5.1 Modélisation de l'environnement (MOD) . . . 70
5.1.1 Les modèles de représentation d'un environnement de Cyberdéfense 70
5.1.2 Positionnement vis-à-vis des Stackelberg Security Games . . . 73
5.1.3 Le modèle Dec-POMDP . . . 73
5.1.4 Les modèles de monde . . . 75
5.2 Apprentissage par renforcement sous contraintes (TRN) . . . 76
5.2.1 Apprentissage par renforcement multi-agent . . . 76
5.2.2 Constrained MDPs, Safe RL et guidages implicites . . . 78
5.2.3 Le modèle $\mathcal{M}$OISE$^+$ . . . 79
5.2.4 Pourquoi le choix de $\mathcal{M}$OISE$^+$? . . . 81
5.3 Explicabilité et extraction organisationnelle (ANL) . . . 81
5.3.1 Notion d'explicabilité . . . 82
5.3.2 Méthodes post-hoc . . . 82
5.3.3 Inférence organisationnelle . . . 83
5.4 Transfert simulation vers environnement réel et cohérence (TRF) . . . 83
5.4.1 Domain adaptation et Sim2Real . . . 84
5.4.2 Robust Reinforcement Learning . . . 84
5.4.3 Adaptation en ligne . . . 84
5.4.4 Synchronisation manuelle . . . 85
5.5 Bilan . . . 85
Conclusion 87

III LA MÉTHODE MAMAD 89
Introduction 91
6 PRÉSENTATION GLOBALE DE LA MÉTHODE 94
6.1 Application flexible de la méthode MAMAD . . . 98
6.2 Bilan . . . 100
7 MODÉLISER L'ENVIRONNEMENT EN SIMULATION 102
7.1 Travaux mobilisés et verrous identifiés . . . 103
7.2 Positionnement et contributions proposées . . . 103
7.2.1 Spécificité Cyberdéfense : éléments génériques vs spécialisés . . . 104
7.2.2 Les *World Models* Multi-Agents pour la génération automatique du modèle simulé . . . 104
7.2.3 Un modèle de simulation pour la génération manuelle du modèle simulé . . . 106
7.3 Description et mise en œuvre dans l'activité . . . 110
7.4 Bilan . . . 112
8 ENTRAÎNER DES POLITIQUES SOUS CONTRAINTES 114
8.1 Travaux mobilisés et verrous identifiés . . . 114
8.2 Positionnement et contributions proposées . . . 115
8.2.1 MOISE+MARL pour lier $\mathcal{M}$OISE$^+$ avec le MARL . . . 116
8.2.2 Détails algorithmiques et passage à l'échelle . . . 117
8.2.3 Portée des contraintes : multi-objectif, shielding et contraintes temporelles . . . 118
8.2.4 Faciliter l'implémentation des **Guides de Contrainte** . . . 119
8.2.5 Extension de MOISE+MARL aux *World Models* Multi-Agents . . . 120
8.3 Description et mise en œuvre dans l'activité . . . 122
8.4 Bilan . . . 123
9 ANALYSER LES COMPORTEMENTS ÉMERGENTS 126

9.1 Travaux mobilisés et verrous identifiés . . . 126
9.2 Positionnement et contributions proposées . . . 127
9.2.1 La méthode TEMM . . . 128
9.2.2 Auto-TEMM : la méthode TEMM étendue avec optimisation des hyperparamètres . . . 129
9.3 Description et mise en œuvre dans l'activité . . . 129
9.4 Bilan . . . 132
10 TRANSFÉRER ET SUPERVISER EN ENVIRONNEMENT RÉEL 135
10.1 Travaux mobilisés et verrous identifiés . . . 135
10.2 Positionnement et contributions proposées . . . 136
10.3 Description et mise en œuvre de l'activité . . . 136
10.4 Bilan . . . 138
Conclusion 140

IV VALIDATION EXPÉRIMENTALE DE LA MÉTHODE 142
Introduction 144
11 CYBMASDE : UN FRAMEWORK SUPPORTANT MAMAD 147
11.1 Fonctionnalités proposées par `CybMASDE` . . . 147
11.2 Cycle de vie de CybMASDE . . . 150
11.3 Cycle de vie implémenté sur Overcooked-AI . . . 153
11.3.1 Configuration initiale entre l'utilisateur, l'environnement réel et CybMASDE . . . 154
11.3.2 Processus *Transferring* . . . 154
11.3.3 Processus MTA : Modelling–Training–Analyzing . . . 156
11.4 Socle technologique (développement) . . . 158
11.5 Intégration des différentes contributions . . . 160
11.5.1 Implémentation du modèle Dec-POMDP pré-spécialisé pour la Cyberdéfense . . . 160
11.5.2 Implémentation du framework MOISE+MARL . . . 162
11.6 Bilan . . . 163
12 CADRE EXPÉRIMENTAL ET D'ÉVALUATION 165
12.1 Description des ensembles d'environnements et algorithmes considérés . . . 165
12.2 Conditions de reproductibilité . . . 166
12.2.1 Conditions expérimentales matérielles . . . 166
12.2.2 Gestion des hyperparamètres (par défaut et surcharges) . . . 167
12.3 Baselines expérimentales . . . 167
12.4 Grille d'évaluation . . . 167
12.4.1 Critères et métriques associées . . . 167
12.5 Protocole d'expérimentation et d'évaluation . . . 170
12.6 Bilan . . . 171
13 ÉTUDES DE CAS 173
13.1 Expérimentations sur les environnements non-orientés Cyberdéfense . . . 173
13.1.1 Description des environnements . . . 173
13.1.2 Description de l'instance commune du protocole d'expérimentation 176
13.2 Expérimentations sur l'environnement Company Infrastructure . . . 177
13.2.1 Description de l'instance du protocole d'expérimentation . . . 179
13.3 Expérimentations sur l'environnement Microservices Kubernetes . . . 181
13.3.1 Description de l'instance du protocole d'expérimentation . . . 182
13.4 Expérimentations sur l'environnement Drone Swarm . . . 184

13.4.1 Description de l'instance du protocole d'expérimentation 186
13.5 Bilan 187
14 RÉSULTATS EXPÉRIMENTAUX ET ANALYSE 189
14.1 Résultats et discussion des environnements non orientés Cyberdéfense . . 189
14.2 Résultats et discussion de l'environnement Company Infrastructure 193
14.3 Résultats et discussion de l'environnement Microservices Kubernetes . . . 195
14.4 Résultats et discussion de l'environnement Drone Swarm 197
14.5 Discussion comparée des résultats 200
14.5.1 Couverture des critères par la méthode 200
14.5.2 Analyse critique 201
14.5.3 Biais potentiels et limites 201
14.6 Bilan 202
Conclusion 204

CONCLUSION GÉNÉRALE 206

Annexes 220
A NOTATIONS DE LA MÉTHODE MAMAD 222
A.1 Notations générales 222
A.2 Notations pour l'activité de modélisation (MOD) 222
A.3 Notations pour l'activité d'entraînement (TRN) 223
A.4 Notations pour l'activité d'analyse (ANL) 223
A.5 Notations pour l'activité de transfert (TRF) 223
B DÉTAILS SUPPLÉMENTAIRES SUR CYBMASDE 225
B.1 Interface graphique de CybMASDE 225
B.2 API Environnementale de CybMASDE 228
B.3 Manuel en ligne de commande de CybMASDE 229

BIBLIOGRAPHIE 233

PUBLICATIONS 249

INDEX 251

TABLE DES FIGURES

Figure 1 Schéma de la logique sous-jacente de notre raisonnement sous-tendant l'organisation du manuscrit . . . 4
Figure 2 Schéma de l'organisation du manuscrit . . . 5
Figure 3 Structure de la Partie I – Contexe de travail . . . 10
Figure 4 Le modèle P3R3 pour la cyber-résilience (tirée de [42]) . . . 14
Figure 5 Description de l'architecture modulaire MASCARA (tirée de [42]) . 16
Figure 6 Illustration schématique d'un SMA de Cyberdéfense dans une infrastructure d'entreprise jouet . . . 17
Figure 7 Exemple schématique d'un SMA . . . 18
Figure 8 Vue synthétique de l'organisation (tirée de [157]) . . . 20
Figure 9 Tension entre approches symboliques et apprenantes : couverture respective des critères C1 à C5 . . . 33
Figure 10 Spécification de la question sous l'angle d'un problème d'optimisation sous contraintes . . . 36
Figure 11 Structure de la Partie II : État de l'art . . . 48
Figure 12 Illustration d'un graphe d'attaque décrivant un scénario de compromission d'un coffre-fort. . . . 71
Figure 13 Illustration d'ADTree d'un scénario d'attaque sur un compte bancaire (tirée de [151]) . . . 72
Figure 14 Illustration de l'architecture d'un *World Model* comprenant l'Auto-encodeur et l'OPM . . . 76
Figure 15 Vue synthétique du modèle $\mathcal{M}$OISE$^+$ . . . 79
Figure 16 Structure de la Partie III : La méthode MAMAD . . . 92
Figure 17 Cycle de vie d'un SMA conçu avec MAMAD . . . 94
Figure 18 Schéma de l'architecture d'un JOPM incluant le RLDM et l'Auto-encodeur . . . 105
Figure 19 Vue illustrative du modèle de simulation . . . 106
Figure 20 Vue minimale du framework MOISE+MARL . . . 116
Figure 21 Temps de convergence normalisé en fonction de la représentativité minimale . . . 130
Figure 22 Structure de la Partie IV : Cadre expérimental et analyse des résultats 145
Figure 23 Diagramme de séquence pour une utilisation de CybMASDE . . . 151
Figure 24 Cycle d'utilisation de CybMASDE . . . 155
Figure 25 Diagramme de composants C4 illustrant l'architecture logicielle de CybMASDE . . . 160
Figure 26 Capture d'écran de l'environnement Overcooked-AI . . . 174
Figure 27 Capture d'écran de l'environnement Predator-Prey . . . 175
Figure 28 Capture d'écran de l'environnement de Warehouse Management . 177
Figure 29 Topologie réseau synthétique : EXT, DMZ, ACC, MAR, SRV. Les sous-réseaux sont interconnectés via routeurs/pare-feu implicites. 179
Figure 30 Aperçu de l'arbre Attaque–Défense (AD) structurant les chemins d'attaque (tactiques/techniques MITRE) et les contre-mesures associées. . . . 180

Figure 31 Cluster *réel* “Services en chaîne” (4 services) et leviers d’action exposés à CybMASDE/MAMAD via l’API Kubernetes. 183
Figure 32 Une vue abstraite du scénario de cluster Kubernetes 184
Figure 33 Schéma d’ensemble de *Kubernetes Autoscaling with Resilient Multi-Agent system* (KARMA) 185
Figure 34 Capture d’écran de l’environnement *Cyber Operations Research Gym* (CybORG) 186
Figure 35 Courbes d’apprentissage (récompenses normalisées, moyenne ± écart-type sur 5 runs). 190
Figure 36 Dendrogramme des trajectoires de transition dans Overcooked-AI 192
Figure 37 PCA des trajectoires de transition dans Overcooked-AI 192
Figure 38 Courbes d’apprentissage (récompense moyenne par épisode) pour l’environnement Company Infrastructure, moyenne ± écart-type sur 5 runs. 193
Figure 39 Courbes d’apprentissage (récompense QoS normalisée) pourMicroservices Kubernetes 195
Figure 40 Courbes d’apprentissage (récompense normalisée) pour Drone Swarm198
Figure 41 Capture d’écran de l’onglet modélisation de l’interface graphique de CybMASDE 225
Figure 42 Capture d’écran de l’onglet entraînement de l’interface graphique de CybMASDE 226
Figure 43 Capture d’écran de l’onglet analyse de l’interface graphique de CybMASDE 226
Figure 44 Capture d’écran de l’onglet raffinement et transfert de l’interface graphique de CybMASDE 227

LISTE DES TABLEAUX

Table 1 Grille des critères d'évaluation d'un SMA de Cyberdéfense 21
Table 2 Un aperçu de quelques organisations et des environnements hôtes utilisés dans les SMAs de Cyberdéfense étudiés 25
Table 3 Un aperçu des fonctions de Cyberdéfense prises en charge par les SMA de Cyberdéfense étudiés 26
Table 4 Lecture synthétique des organisation des SMAs de Cyberdéfense selon les critères C1–C5 27
Table 5 Exemples de cadres de conception pour agents de Cyberdéfense . 29
Table 6 Couverture des cadres de conception de SMA de Cyberdéfense au regard des critères C1–C5 30
Table 7 Synthèse des relations entre critères, activités, sous-problèmes et hypothèses ainsi que futurs verrous et contributions 41
Table 8 Couverture des critères spécifiques par les principales familles de travaux de modélisation d'environnements de Cyberdéfense 52
Table 9 Couverture des critères par les principales familles de travaux sur l'intégration de contraintes/guidages organisationnels dans le MARL 57
Table 10 Couverture des critères par les principales familles de travaux sur l'extraction organisationnelle émergente 60
Table 11 Couverture des critères par les principales familles de travaux sur le maintien de cohérence simulation/réel 64
Table 12 Synthèse des travaux retenus, verrous, limites et besoins méthodologiques par hypothèse 67
Table 13 Taxonomie de la méthode MAMAD avec activités, sous-activités et acronymes 98
Table 14 Exemple de guides appliqués au TP "remplir un pot avec un oignon". 120
Table 15 Résumé du fichier "project_configuration.json" avec exemples sur Overcooked-AI 148
Table 16 Correspondance entre critères globaux et métriques 168
Table 17 Caractérisation générique de la "baseline avancée" 170
Table 18 Baselines synthétiques pour les environnements non-orientés Cyberdéfense. 178
Table 19 Baselines synthétiques pour Company Infrastructure. 181
Table 20 Baselines synthétiques Microservices Kubernetes. 185
Table 21 Baselines synthétiques pour Drone Swarm. 187
Table 22 Récompenses cumulées et convergence (moyenne ± écart-type, 5 runs). 190
Table 23 Synthèse des résultats (moyenne sur 5 runs, ± écart-type) pour Company Infrastructure. 194
Table 24 Régime nominal (moyenne ± écart-type sur 5 runs, fenêtres de 2 h). 196
Table 25 Robustesse par scénario (moyenne sur 5 runs). 196
Table 26 Synthèse multi-métriques (moyenne ± écart-type sur 5 runs). . . . 197
Table 27 Résultats nominaux pour Drone Swarm (moyenne ± écart-type, 5 runs). 198

Table 28 Robustesse selon le scénario de compromission (moyenne ± écart-type, 5 runs). 199
Table 29 Résumé multi-métriques pour Drone Swarm (moyenne ± écart-type, 5 runs). 199
Table 30 Synthèse multi-environnements : couverture des critères C1–C5 par MAMAD (moyenne des métriques associées, normalisées sur [0,1], calculée sur 5 runs indépendants). 200

LISTE DE LISTINGS

Listing 1 Exécution complète de CybMASDE en mode full-auto 149
Listing 2 Extrait du fichier de configuration organisationnelle pour Overcooked-AI 156
Listing 3 Extrait de la sortie de log après entrainement et application de TEMM 157
Listing 4 Extrait de spécifications organisationnelles inférées à analyser puis raffiner manuellement 157
Listing 5 Extrait du fichier gabarit à utiliser pour implémenter l'API environnementale. 228

LISTE D'ACRONYMES

ACD *Automated Cyber Defense*

ACO *Autonomous Cyber Operation*

AD *Attack-Defense*

AEC *Agent Environment Cycle*

AGR *Agents, Groups, Roles*

AHPA *Advanced Horizontal Pod Autoscaler*

AICA *Autonomous Intelligent Cyberdefense Agent*

AMD *Advanced Micro Devices*

ANL-AUT *Automated Analysis*

ANL-MAN *Manual Analysis*

ANSSI Agence Nationale de la Sécurité des Systèmes d'Information

AOSE *Agent Oriented Software Engineering*

API *Application Programming Interface*

ASAP *As Soon As Possible*

Auto-TEMM *Automatic Trajectory-based Evaluation in MOISE+MARL*

AutoPentest-DRL *Automated Penetration Testing Using Deep Reinforcement Learning*

AWARE *Adaptive Web-based Analysis for REsilience*

C2 *Command and Control*

CAGE *Cyber Automated Game-based Evaluation*

CANDLES *Cybersecurity ANomaly Detection via Learning and Evaluation System*

CAV *Concept Activation Vector*

CLAP *Contrastive Language-Audio Pretraining*

CLI *Command Line Interface*

CLS *Classification token*

CMDP *Constrained Markov Decision Process*

COMA *Counterfactual Multi-Agent Policy Gradients*

COPA *Combined Autoscaling for Kubernetes*

CPO *Constrained Policy Optimization*

CPU *Central Processing Unit*

CSIRT *Computer Security Incident Response Team*

CSLE *Cyber Security Learning Environment*

CTDE *Centralized Training Decentralized Execution*

CTF *Capture The Flag*

CUDA *Compute Unified Device Architecture*

CyberBattleSim *Cyber Battle Simulator*

CyberVAN Cyber Virtual Ad-hoc Networking

CybMASDE *Cyberdefense Multi-Agent System Development Environment*

CybORG *Cyber Operations Research Gym*

CYST *Cyber Security Simulation Testbed*

DB *Database*

DCOP *Distributed Constraint Optimization Problem*

DCQL *Deep Constrained Q-Learning*

DDS *Data Distribution Service*

Dec-POMDP *Decentralized Partially Observable Markov Decision Process*

DETERLab cyber DEfense Technology Experimental Research Laboratory

DQN *Deep Q-Network*

DRL *Deep Reinforcement Learning*

EmuLab *Emulation Laboratory*

ES *Email Server*

FN Faux Négatif

FOF *Functional Organizational Fit*

FP Faux Positif

FW *Firewall*

GPU *Graphics Processing Unit*

HPA *Horizontal Pod Autoscaler*

HPC *High Performance Computing*

HPO *Hyper-Parameter Optimization*

IA Intelligence Artificielle

ID *Intrusion Detection*

IDS *Intrusion Detection System*

IoC *Indicator of Compromission*

IOC *Indicator of Compromise*

IP *Internet Protocol*

IQL *Independent Q-Learning*

JAX *Just-in-time Accelerated computation (Google)*

JOPM *Joint-Observation Prediction Model*

JSON *JavaScript Object Notation*

KARMA *Kubernetes Autoscaling with Resilient Multi-Agent system*

KL *Kullback–Leibler divergence*

KNN *K-Nearest Neighbors*

KOSMOS *Kubernetes Vertical and Horizontal Resource Autoscaling*

LCS *Longest Common Sequence*

LGPL *GNU Lesser General Public License*

LIME *Local Interpretable Model-agnostic Explanations*

LLM *Large Language Model*

LM *Language Model*

LR *Learning Rate*

LRP *Layer-wise Relevance Propagation*

LSTM *Long-Short Term Memory*

MADDPG *Multi-Agent Deep Deterministic Policy Gradient*

MAE *Mean Absolute Error*

MAMAD *MOISE+MARL Assisted MAS Design*

MAPPO *Multi-Agent Proximal Policy Optimization*

MARL *Multi-Agent Reinforcement Learning*

MARLlib *Multi-Agent Reinforcement Learning Library*

MAS *Multi-Agent System*

MASCARA *Multi-Agent System Centric AICA Reference Architecture*

MAVIPER *Multi-Agent VIrtual PERimeter*

MBRL *Model-based Reinforcement Learning*

MCAS *Multi-Cyberdefense Agent Simulator*

MDP *Markov Decision Process*

MEM *Memory*

MENTOR *Multi-agent Environment for Networked Training, Operations and Response*

MIT *Massachusetts Institute of Technology*

ML *Machine Learning*

MLP *Multi-Layer Perceptron*

MMA *MOISE+MARL API*

MMD *Maximum Mean Discrepancy*

MOD-AUT *Automated Modelling*

MOD-MAN *Manual Modelling*

MSE *Mean Squared Error*

MTA Modéliser-Entraîner-Analyser

NASim Network Attack Simulator

NASimEmu *Network Attack Simulator & Emulator*

NIST *National Institute of Standards and Technology*

NVIDIA *NVIDIA Corporation*

ODec-POMDP *Observation-based Dec-POMDP*

OF *Organizational Fit*

OPM *Observation Prediction Model*

OS *Organizational Specifications*

OTAN Organisation du Traité de l'Atlantique Nord
PAM *Privileged Access Management*
PCA *Principal Component Analysis*
PenGym *Penetration Testing Gym*
PILCO *Probabilistic Inference for Learning Control*
POMDP *Partially Observable Markov Decision Process*
POSG *Partially Observable Stochastic Games*
PPO *Proximal Policy Optimization*
PS *Privileged Service*
QMIX *Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning*
RAM *Random Access Memory*
REST *Representational State Transfer*
RL *Reinforcement Learning*
RLDM *Recurrent Latent Dynamics Model*
RNN *Recurrent Neural Network*
ROMA *Role-Oriented Multi-Agent Reinforcement Learning*
ROS *Robot Operating System*
SCYTHE *SCYTHE Cybersecurity Platform*
SDN *Software Defined Networking*
SHAP *SHapley Additive exPlanations*
SMA Système Multi-Agent
SOF *Structural Organizational Fit*
SP Séquence Parente
SPOF *Single Point Of Failure*
SVM *Support Vector Machine*
TAB *Terminal Access Broker*
TEMM *Trajectory-based Evaluation in MOISE+MARL*
TPE *Tree-structured Parzen Estimator*
TRF-AUT *Automated Transferring*
TRF-MAN *Manual Transferring*
TRN-CON *Constrained/Guided Training*
TRN-UNC *Unconstrained Training*
TRPO *Trust Region Policy Optimization*
TTP Tactiques, Techniques et Procédures
VAE *Variational Auto Encoder*
VDN *Value-Decomposition Networks*
VM *Virtual Machine*
VPN *Virtual Private Network*

WS *Web Server*

XAI *eXplainable Artificial Intelligence*

INTRODUCTION GENERALE

Motivations

Depuis plus d'une décennie, la complexité, l'automatisation et la rapidité d'exécution des cyberattaques remettent en question les approches classiques de Cyberdéfense, centralisées et principalement réactives ou basées sur des règles fixes. Dans un contexte où les systèmes à défendre deviennent eux-mêmes distribués, et dynamiques (architectures microservices, essaim de drones, IoT industriels, etc.), les mécanismes de défense doivent eux aussi gagner en autonomie, en adaptabilité et en résilience.

L'idée d'un agent intelligent de Cyberdéfense, capable de détecter et contrer de manière autonome les menaces, a été incarnée par le concept d'agent *Autonomous Intelligent Cyberdefense Agent* (AICA) [42]. La première génération d'AICA a pris la forme d'un agent *monolithique*, doté de capacités de perception, décision et action sur un périmètre localisé. Rapidement, la complexité des situations traitées a justifié une évolution vers une architecture plus explicite et modulaire, illustrée par l'architecture *Multi-Agent System Centric AICA Reference Architecture* (MASCARA) [42], qui propose une architecture modulaire symbolique de type cognitif de l'agent AICA.

Dans la continuité de ces travaux, cette approche distribuée a évolué vers une version multi-agent du concept AICA, dans laquelle plusieurs agents collaborent pour assurer la défense d'un système complexe, ouvrant la voie au concept de SMA de Cyberdéfense. Cette évolution soulève de nombreuses questions : comment spécifier une organisation adaptée aux contraintes de l'environnement et aux attaques ? Quelles capacités doivent prendre ces agents et pour quel coût ? Comment garantir que leurs comportements respectent des exigences fonctionnelles (performance, adaptation) et non fonctionnelles (sûreté, explicabilité) ?

Objectifs

L'objectif global de cette thèse est d'obtenir un SMA de Cyberdéfense capable de s'adapter aux contraintes dynamiques de l'environnement à défendre, tout en maximisant sa capacité à détecter, prévenir et contrer les menaces. Plutôt que de chercher une solution de SMA de Cyberdéfense unique pour répondre à ce problème complexe, nous visons à établir une méthode de conception pour guider ou assister le processus de création d'un tel SMA. Cette méthode se veut générique et capable d'intégrer, organiser et orchestrer les différentes contributions de manière cohérente.
L'idée de mettre en place une telle méthode implique de prendre en compte plusieurs défis :

- **Modéliser le problème de conception** du SMA pour prendre en compte à la fois les objectifs de Cyberdéfense et les contraintes imposées par l'environnement ou le concepteur ;
- **Réduire le coût de conception** en s'affranchissant partiellement de la dépendance aux connaissances expertes ;
- **Évaluer quantitativement la performance** du SMA sur des critères tels que la résilience, l'autonomie ou l'adaptabilité ;

- **Assurer l'explicabilité** du comportement global du SMA, notamment en identifiant les rôles, objectifs et interactions émergents ;
- **Garantir la sûreté de fonctionnement et le respect des contraintes critiques**, par des mécanismes de contrôle organisationnel ;
- **Permettre l'adaptation dynamique** du SMA face à l'évolution des contraintes et des situations dans l'environnement réel.

Ces défis orientent la recherche vers une méthode qui couvre la conception du SMA de Cyberdéfense en cherchant à bénéficier des avantages à la fois des approches connexionniste et symbolique.

Contributions visées

Nous cherchons à établir une méthode de conception de SMA pour des SMAs de Cyberdéfense. En supposant la simulation nécessaire, notre méthode peut être abordée en quatre activités :

- **Modélisation** : création d'un environnement simulé soit manuellement en suivant un cadre Markovien générique et des connaissances expertes, soit automatiquement à l'aide de techniques *Machine Learning* (ML) capturant les dynamiques de l'environnement à partir de ses traces ;
- **Résolution** : obtention de politiques multi-agent conjointes manuellement ou par des techniques ML sous des contraintes exprimées comme spécifications organisationnelles notamment ;
- **Analyse** : inférence de spécifications organisationnelles émergentes, telles que des rôles ou des objectifs, à partir des trajectoires des agents, en s'appuyant sur des techniques d'apprentissage non supervisé notamment ;
- **Transfert** : couplage et minimisation de l'écart entre l'environnement réel et l'environnement simulé à la manière de jumeaux digitaux par la mise à jour des politiques des agents réels et de l'environnement simulé.

Ces activités nécessitent plusieurs contributions :

- Une formalisation du problème de conception comme un problème d'optimisation sous les contraintes de l'environnement et des concepteurs. Sur cette base, une formalisation de la méthode pourra être proposée pour modéliser formellement toute la chaîne de traitement orchestrant l'ensemble des contributions ;
- Une extension multi-agent de techniques ML comme les *World Models* permettra de capturer la dynamique de transition observationnelle, permettant ainsi de générer automatiquement un modèle simulant l'environnement tel que perçu par les agents ;
- Un framework qui fournira un cadre générique permettant de guider la modélisation manuelle d'un environnement réel comme une simulation à l'aide de connaissances expertes ;
- Un framework qui permettra d'intégrer les exigences des concepteurs dans le processus d'obtention des politiques, telle que dans un processus MARL où l'on pourra guider ou contraindre l'apprentissage des politiques ;

- Une méthode d'analyse des comportements émergents, capable d'inférer des spécifications organisationnelles à partir des trajectoires des agents entraînés ;
- Un outil implémentant l'ensemble de la méthode comprenant les différentes contributions et son application dans plusieurs environnements sur une partie ou l'ensemble des activités.

La méthode devra être évaluée dans des environnements de Cyberdéfense représentatifs. La méthode pourra également être évaluée dans des environnements non orientés Cyberdéfense pour montrer sa généralisabilité dans d'autres contextes multi-agents.

Plan du manuscrit

La question de l'organisation d'un SMA de Cyberdéfense nous a d'abord conduits à effectuer une revue de littérature, dont l'analyse a mis en évidence l'intérêt de spécifier la question de recherche dans le cadre d'un problème d'optimisation. Cette question spécifiée permet de structurer notre démarche autour d'une série d'hypothèses, combinant des approches symboliques et connexionnistes, qui fondent les différentes contributions de la thèse. Ces contributions sont ensuite évaluées expérimentalement, afin de tester la validité des hypothèses et, in fine, de répondre à la question de recherche.
Le manuscrit est structuré en cinq parties, composées de trois à cinq chapitres chacune, comme présenté en Figure 2. Cette organisation suit un raisonnement progressif, détaillé en Figure 1.
La Partie I : Contexte de travail introduit la question de recherche générale en mettant en évidence les limites des approches existantes pour concevoir des SMA dédiés à la Cyberdéfense. Elle motive ainsi le recours à une approche hybride, combinant modèles symboliques et techniques connexionnistes, et spécifie la question de recherche dans un cadre d'optimisation sous contraintes. Cette partie se termine par la présentation des hypothèses qui structureront l'ensemble des travaux. S'appuyant sur ces hypothèses, la Partie II : Etat de l'art établit les fondements théoriques nécessaires et identifie, pour chacune d'elles, les verrous scientifiques qui devront être levés. Elle constitue ainsi le socle conceptuel sur lequel reposera la méthode proposée. Dans la continuité, la Partie III : La méthode MAMAD décrit la méthode *MOISE+MARL Assisted MAS Design* (MAMAD) [31], conçue pour répondre à la question de recherche. Elle en détaille les quatre activités clés et explicite les contributions spécifiques associées à chacune. Pour évaluer la méthode et valider les hypothèses sous-jacentes la Partie IV : Validation expérimentale de la méthode présente le protocole expérimental mis en place. Elle décrit les environnements testés, les objectifs d'évaluation, les métriques utilisées, ainsi que les résultats obtenus et leur analyse comparative. Enfin, la Partie 14.6 : Conclusion propose une synthèse des contributions de la thèse. Elle discute les limites rencontrées et trace plusieurs perspectives pour prolonger ce travail.

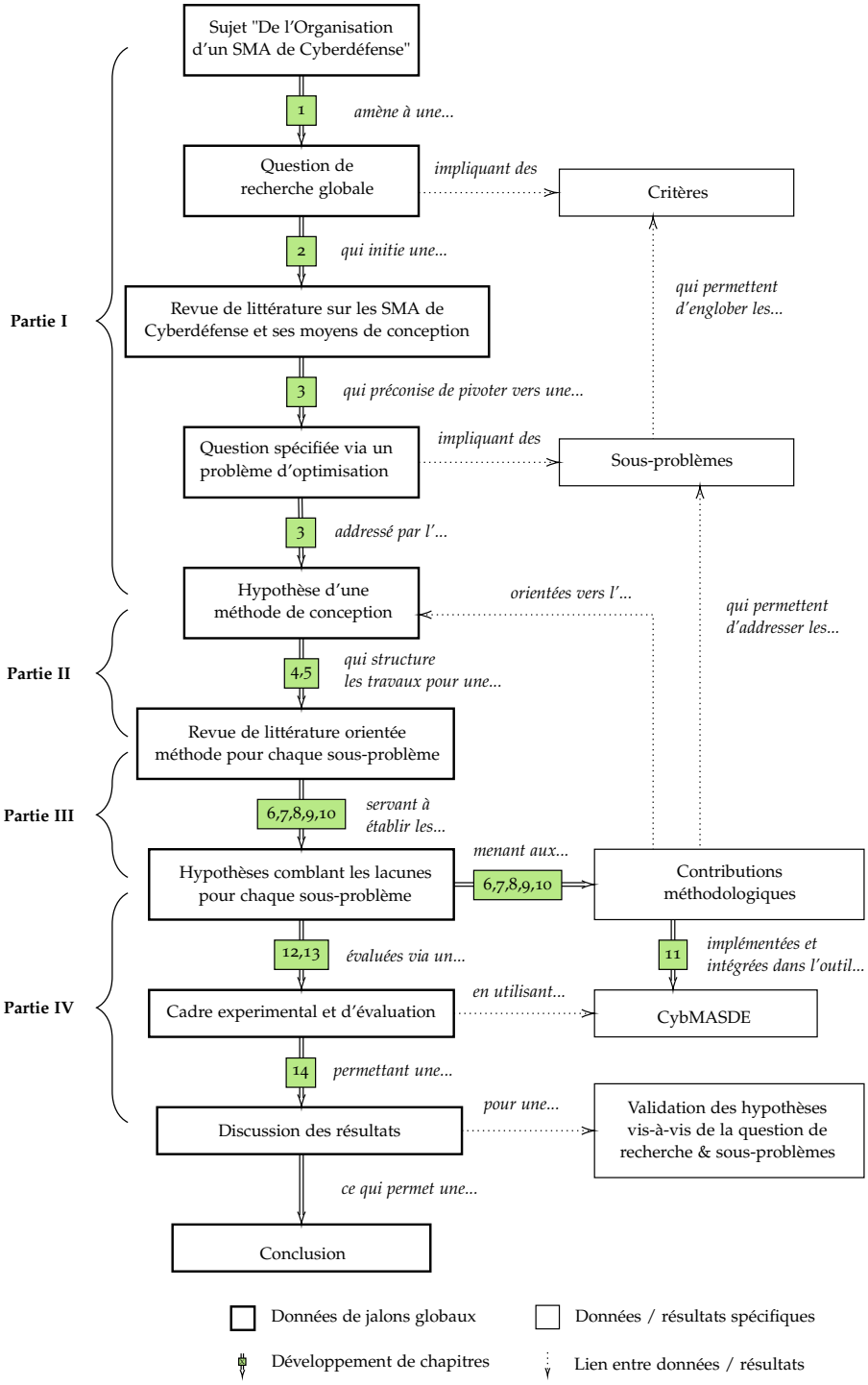


FIGURE 1 : Schéma de la logique sous-jacente de notre raisonnement sous-tendant l'organisation du manuscrit

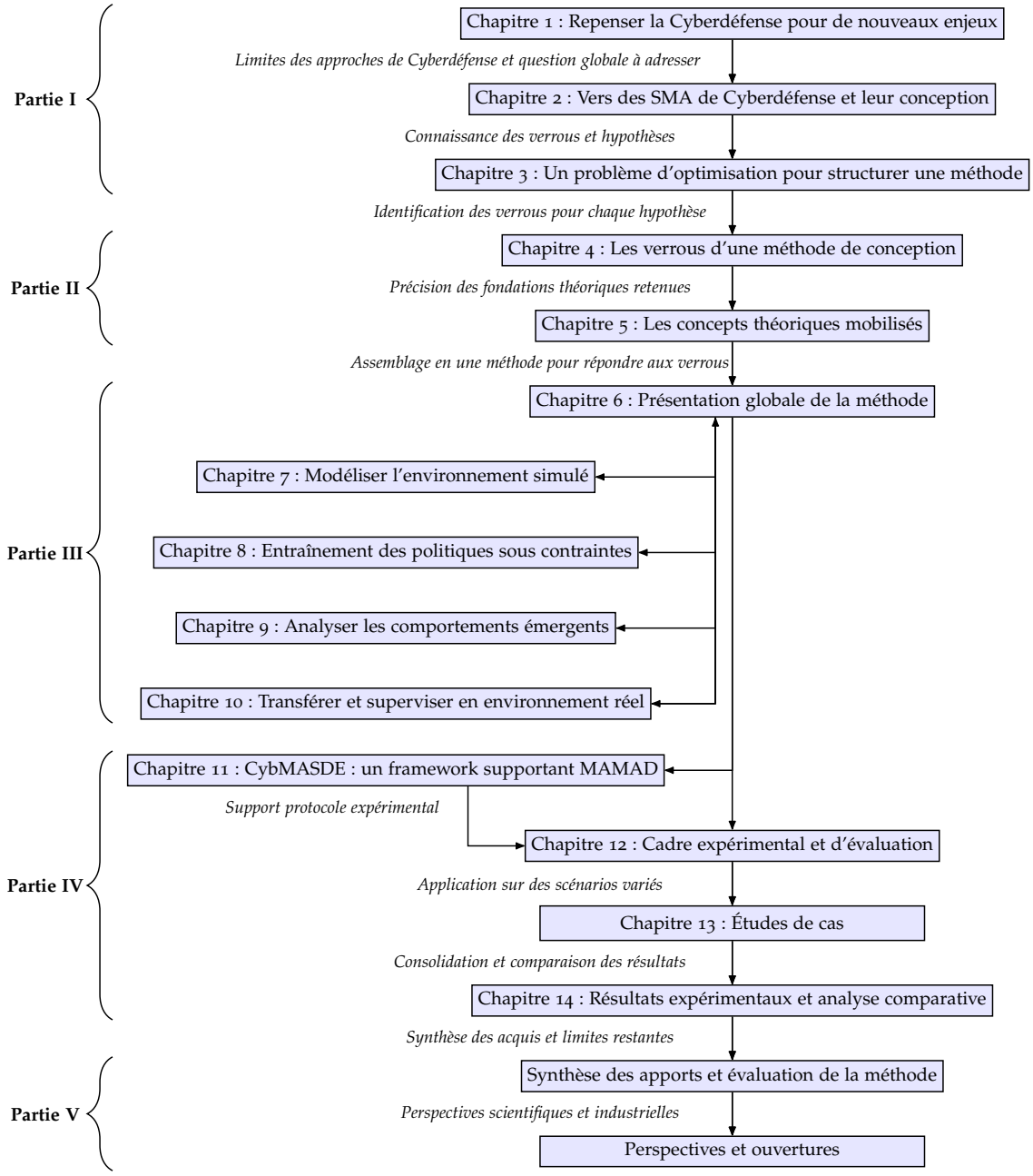


FIGURE 2 : Schéma de l'organisation du manuscrit

Première partie

CONTEXTE DE TRAVAIL

INTRODUCTION

Cette première partie introduit le cadre général de la thèse. Elle invite à explorer les fondations scientifiques, les motivations opérationnelles, et les questionnements de fond qui orientent notre démarche. Sans prétendre apporter dès à présent des réponses définitives, elle pose les jalons nécessaires pour comprendre le cheminement qui sera suivi.

Pourquoi envisager la Cyberdéfense sous un angle distribué ? Quelles promesses offrent les SMAs dans ce domaine ? En quoi une formalisation en un problème d'optimisation sous contraintes permet-elle de dépasser les tensions entre performance, contrôle et explicabilité ? Ces interrogations dessinent progressivement les contours de notre positionnement, à la croisée des approches symboliques et connexionnistes.

Le schéma présenté en Figure 3 illustre l'enchaînement logique des chapitres de cette première partie. Le premier chapitre amorce la réflexion sur la Cyberdéfense décentralisée et distribuée, introduit les concepts clés, et installe la question globale qui guidera l'ensemble du manuscrit. Le deuxième chapitre approfondit les enjeux identifiés en exposant les apports et limites des travaux actuels. Enfin, à la lumière des verrous dégagés, le troisième chapitre propose de spécifier la question de recherche dans un cadre d'optimisation sous contraintes, ouvrant ainsi la voie à une structuration claire des hypothèses sur lesquelles s'appuieront les contributions.

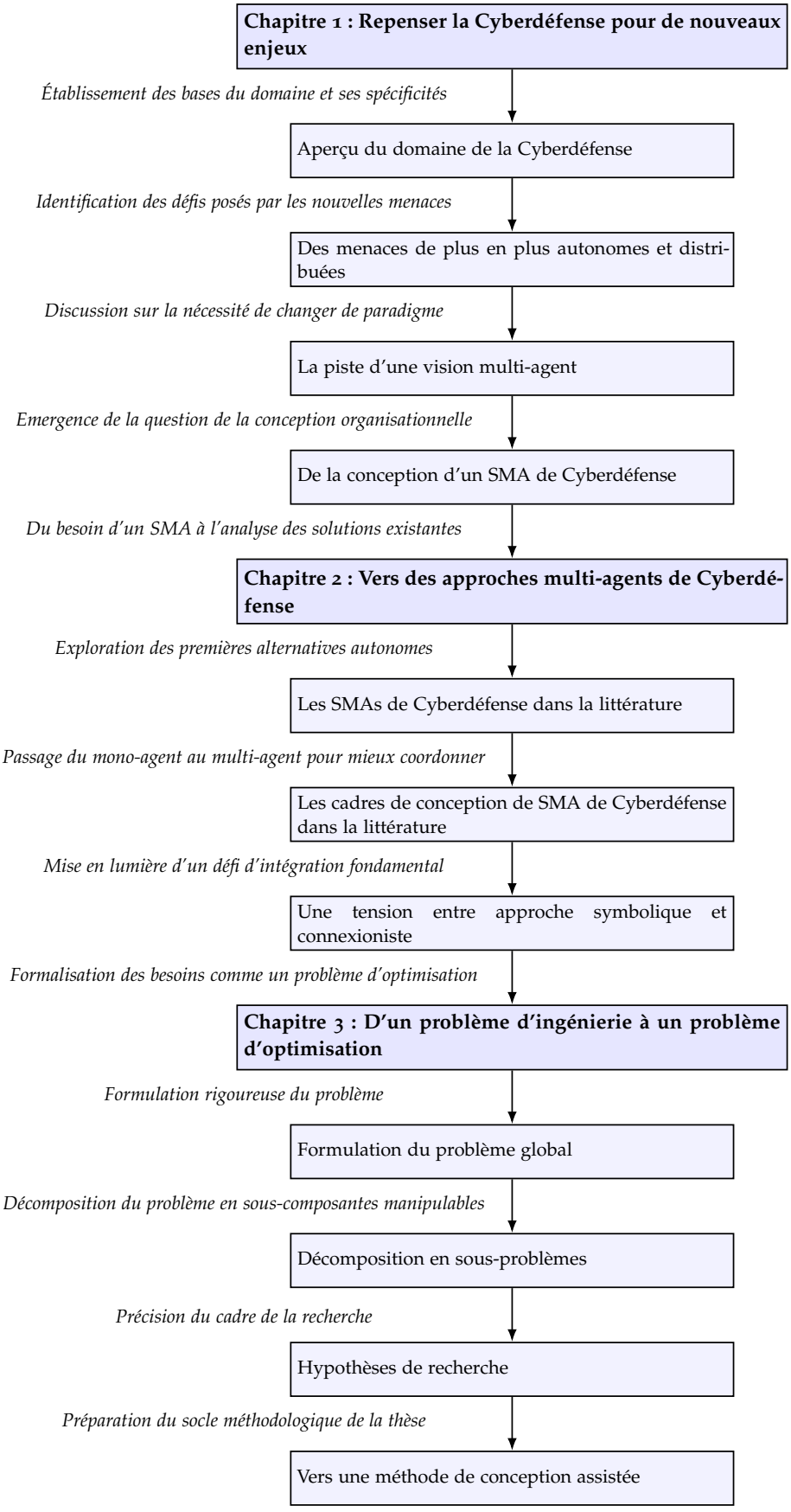


FIGURE 3 : Structure de la Partie I – Contexe de travail

1 REPENSER LA CYBERDÉFENSE POUR DE NOUVEAUX ENJEUX

À mesure que les systèmes informatiques gagnent en complexité, en interconnexion et en criticité, les menaces qui les visent se diversifient et s'intensifient. La Cybersécurité et la Cyberdéfense ne sont plus uniquement des domaines techniques : elles deviennent des piliers stratégiques au cœur des préoccupations des États, des entreprises, et des infrastructures critiques [11].

Cette thèse explore la voie d'une Cyberdéfense distribuée, dynamique, et guidée par des principes d'organisation multi-agent. Ce chapitre pose les fondations conceptuelles et problématiques de cette approche.

Nous commencerons par définir les concepts fondamentaux de la Cybersécurité et de la Cyberdéfense, en exposant les objectifs, les acteurs et les axes de recherche actuels. Nous discuterons ensuite des menaces émergentes liées à l'Intelligence Artificielle (IA) agentique, avant de présenter une approche multi-agent de la Cyberdéfense comme une réponse potentielle à ces défis. Enfin, nous formulerons la question générale à laquelle cette thèse entend répondre.

1.1 APERÇU DU DOMAINE DE LA CYBERDÉFENSE

La protection des systèmes numériques, face à des menaces en perpétuelle évolution, constitue un enjeu stratégique. Deux disciplines complémentaires s'organisent autour de cet enjeu : la **Cybersécurité**, la **Cyberdéfense** et la **Cyberrésilience**.

Cybersécurité : prévention et protection systémique

La **Cybersécurité** recouvre les pratiques, technologies et politiques visant à préserver la confidentialité, l'intégrité et la disponibilité des systèmes d'information [3]. Elle inclut la sécurisation des infrastructures, des systèmes d'exploitation et des réseaux, le contrôle d'accès et la gestion des identités, le chiffrement et la protection des données en transit et au repos, la gestion des vulnérabilités et les analyses de risque, ainsi que la formalisation de la politique de sécurité du système d'information et du plan de continuité d'activité.

Ces mesures sont essentiellement de nature préventive et systémique, mises en œuvre d'emblée pour minimiser la surface d'exposition aux attaques.

Cyberdéfense : détection active et réaction organisée

La **Cyberdéfense** adopte une posture plus réactive et adaptative. Selon l'Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI) et l'Organisation du Traité de l'Atlantique Nord (OTAN) [3, 131], elle englobe les *mesures actives, organisationnelles et opérationnelles* permettant de détecter, analyser, contrer et neutraliser les menaces, tout en rétablissant les capacités des systèmes affectés. Ses composantes incluent la supervision, qui regroupe l'agrégation et la corrélation de journaux ainsi que la détection d'anomalies à grande échelle ; la détection et l'analyse de menaces ou d'attaques en cours, fondées sur l'utilisation de sondes, d'*Indicator of Compromission* (IoC) et de méthodes comporte-

mentales reposant sur le ML notamment ; la réaction rapide, comprenant l'isolement des composants compromis, la neutralisation des menaces ou le filtrage automatique ; la restauration et la résilience, à travers des mécanismes de redéploiement, de reprise d'activité ou de reconfiguration automatisée ; enfin, le renseignement et l'attribution, permettant l'identification des tactiques adverses en tant que Tactiques, Techniques et Procédures (TTP) et le traçage des menaces.

Cette approche est incarnée dans les centres opérationnels tels que les équipes *Computer Security Incident Response Team* (CSIRT) ou les centres *Command and Control* (C2), intégrant les dimensions technique, organisationnelle et réglementaire.

Axes structurants de la recherche en Cyberdéfense

Les travaux scientifiques en Cyberdéfense se répartissent selon plusieurs axes complémentaires [129]. La détection des intrusions constitue un domaine central, reposant sur des approches par signatures [176] ou par apprentissage automatique [129, 154]. La résilience, quant à elle, s'intéresse à la robustesse et à la tolérance aux pannes, notamment via des mécanismes de redondance et des architectures multitier [33]. L'automatisation défensive (ou *Automated Cyber Defense* – ACD) s'appuie sur des playbooks, des outils d'orchestration et des agents autonomes pour accélérer la réponse [58]. L'IA, en particulier l'apprentissage par renforcement (*Reinforcement Learning* – RL) et le MARL (MARL), est exploitée pour anticiper les attaques et adapter dynamiquement les défenses dans des environnements simulés tels que CybORG, NASim ou Yawning Titan [46, 53, 74]. Par ailleurs, la modélisation des adversaires, inspirée par la théorie des jeux et les modèles probabilistes, permet d'anticiper leurs tactiques. Enfin, des environnements de simulation sont développés pour servir de bancs d'essai à l'entraînement et à l'évaluation d'agents défensifs.

Cyber-résilience : un paradigme intégratif

La **cyber-résilience** cherche à articuler Cybersécurité et Cyberdéfense en adoptant une approche globale : anticiper, résister, répondre, récupérer et évoluer face aux cyberattaques [103]. Le modèle P3R3 [144], illustré en Figure 4, formalise cette démarche en six activités complémentaires : la prévision, qui repose sur la cartographie des menaces et le renseignement ; la prévention, consistant à réduire la surface d'attaque et dissuader les agresseurs ; la protection, centrée sur la sécurisation active et la conformité aux normes ; la supervision, entendue comme la capacité à détecter des cyber-attaques et activer les mécanismes de réponse ; la réponse, qui regroupe les actions immédiates de mitigation ; et enfin la reprise, visant la restauration des services altérés [144].

Ce modèle s'inscrit dans la continuité des démarches proposées par le *National Institute of Standards and Technology* (NIST) et MITRE, tout en mettant l'accent sur la prévision proactive de menaces [144]. Dans ce cadre, la Cyberdéfense recouvre prioritairement les activités de protection, de réponse et de reprise, qui constituent le cœur de notre approche. Des travaux récents soulignent d'ailleurs l'intérêt d'y intégrer des agents autonomes, non plus uniquement observateurs, mais capables d'agir et de contribuer activement à la résilience du système [42].

L'essentiel de la cyber-résilience repose sur une combinaison de détection et de réponse face aux cybermenaces. Toutefois, l'émergence de menaces toujours plus rapides, distribuées et intelligentes, remet en cause les approches traditionnelles. C'est ce que nous

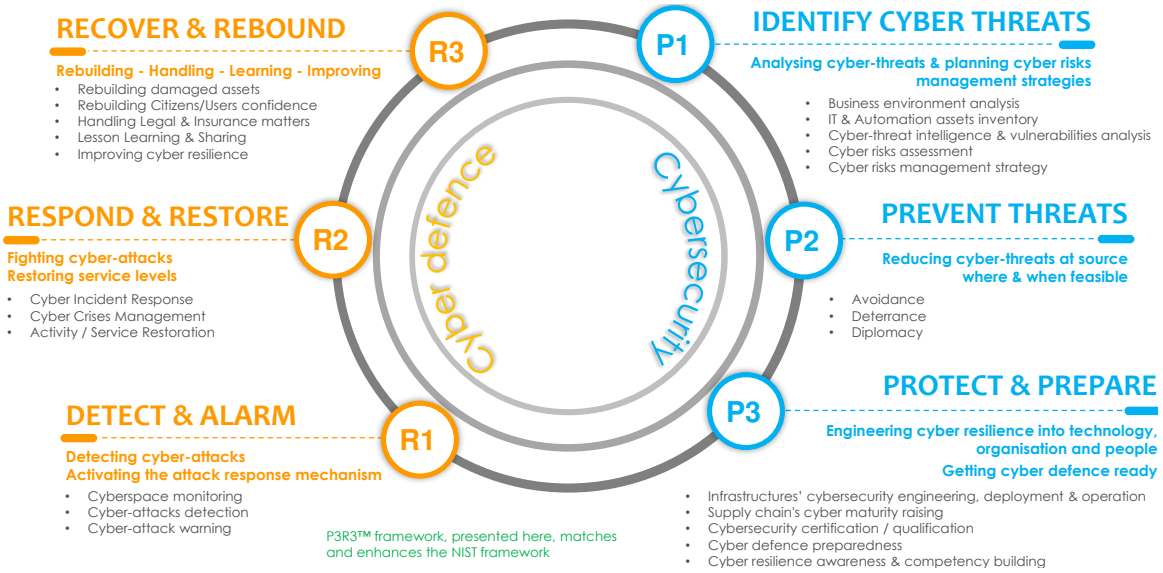


FIGURE 4 : Le modèle P3R3 pour la cyber-résilience (tirée de [42])

explorons dans la section suivante, en analysant les évolutions récentes du paysage des cyberattaques et les défis associés à leur détection et leur neutralisation.

1.2 DES MENACES DE PLUS EN PLUS AUTONOMES ET DISTRIBUÉES

Au cours des dernières années, l'écosystème des cybermenaces a subi une transformation majeure. L'arrivée de l'IA agentique permet l'émergence d'attaques plus rapides, automatisées, adaptatives et opérant en parallèle non seulement sur un hôte unique, mais sur des réseaux entiers [81]. Des travaux récents montrent que des attaquants utilisent des *Large Language Model* (LLM) pour générer des malwares, créer des campagnes de phishing ciblées, et réagir d'eux-mêmes aux systèmes de défense [34]. De tels systèmes permettent de déclencher des campagnes distribuées avec une vitesse et une efficacité très importante [9].

Limites des approches classiques de la Cyberdéfense

Les dispositifs de Cyberdéfense traditionnels (fondés sur des architectures centralisées, des signatures statiques ou des règles prédéfinies) sont aujourd'hui dépassés par cette nouvelle génération de menaces [42] :

- **Latence décisionnelle** : la centralisation de la détection engendre des délais critiques, permettant à certaines attaques de se produire avant même qu'elles ne soient identifiées et traitées par les équipes de Cyberdéfense ;
- **Rigidité adaptative** : des règles statiques ne peuvent suivre l'évolution des modes opératoires des attaquants ;
- **Peu de résilience** : lorsqu'une attaque est détectée, aucune action corrective immédiate n'est engagée. Cela retarde la restauration du système et laisse les conséquences de l'incident s'aggraver, limitant ainsi l'efficacité de la réponse post-attaque.

Ces limites identifiées soulignent la nécessité d'un paradigme plus agile, proactif et intelligent dans la Cyberdéfense. Des études récentes identifient l'avènement d'attaques basées sur l'IA [5, 34, 104], où les agents malveillants :

- **Automatisent** la recherche de vulnérabilités, le déploiement de charges utiles, et l'exfiltration ;
- **Coopèrent** en coordonnant des vecteurs d'attaque parallèles décrits dans les modèles de multi-agents adverses ;
- **Exploitent** des techniques d'**adversarial ML**, générant des formes d'échappement aux détections habituelles (poisoning, prompt injection…).

Une approche autonome de Cyberdéfense

Pour faire face à ces menaces, le domaine émergent de l'*Autonomous Cyber Operation* (ACO) cherche à développer des systèmes capables de prendre des décisions complexes de manière autonome, en tenant compte du contexte, des objectifs à atteindre et des conséquences possibles de leurs actions, le tout en temps réel et avec une supervision humaine minimale, voire inexistante [50]. Les premiers travaux dans ce domaine portent principalement sur des architectures à base d'agents logiciels autonomes, en particulier dans le cadre du groupe *IST-152* de l'OTAN, à l'origine du concept d'agent AICA. Un tel agent est théorisé comme étant capable de percevoir son environnement local (par l'analyse de journaux, de flux ou d'heuristiques), de prendre des décisions autonomes en s'appuyant sur des règles ou des mécanismes d'apprentissage, d'agir localement (par exemple via des actions de filtrage ou d'isolement) sans dépendre d'un contrôle externe permanent, et enfin de communiquer avec d'autres agents ou des opérateurs humains afin de partager des indicateurs, des intentions ou des états.

Dans ce cadre, l'architecture modulaire MASCARA [88], illustrée en Figure 5, a été introduite pour formaliser le fonctionnement interne d'un agent AICA en décomposant ses activités en plusieurs modules spécialisés : collecte de journaux, détection d'anomalies, sélection de contre-mesures, application des réponses, etc. En s'appuyant sur cette architecture générale, il devient possible de concevoir une instance concrète adaptée à un environnement spécifique, en modulant le nombre de composants, leur nature et leurs interactions. Une telle adaptation permet à l'agent AICA de répondre finement aux exigences de son contexte de déploiement et de garantir une protection *en edge*, c'est-à-dire au plus proche des ressources à défendre, tout en optimisant la performance dans l'atteinte des objectifs de Cyberdéfense.

Cependant, cette architecture reste fondamentalement monolithique et figée. Elle ne fournit pas, en l'état, les moyens d'adaptation nécessaires pour faire face aux dynamiques imprévisibles d'un environnement opérationnel réel, en particulier dans le cadre de systèmes distribués, complexes et fortement interactifs. Cette rigidité limite significativement la capacité d'un agent AICA à réagir à l'émergence de nouvelles menaces ou à s'ajuster aux contraintes environnementales fluctuantes.

Vers des approches coopératives

La simple composition modulaire proposée initialement ne suffit plus à répondre à la complexité croissante des infrastructures critiques actuelles. Celle-ci impose l'adoption

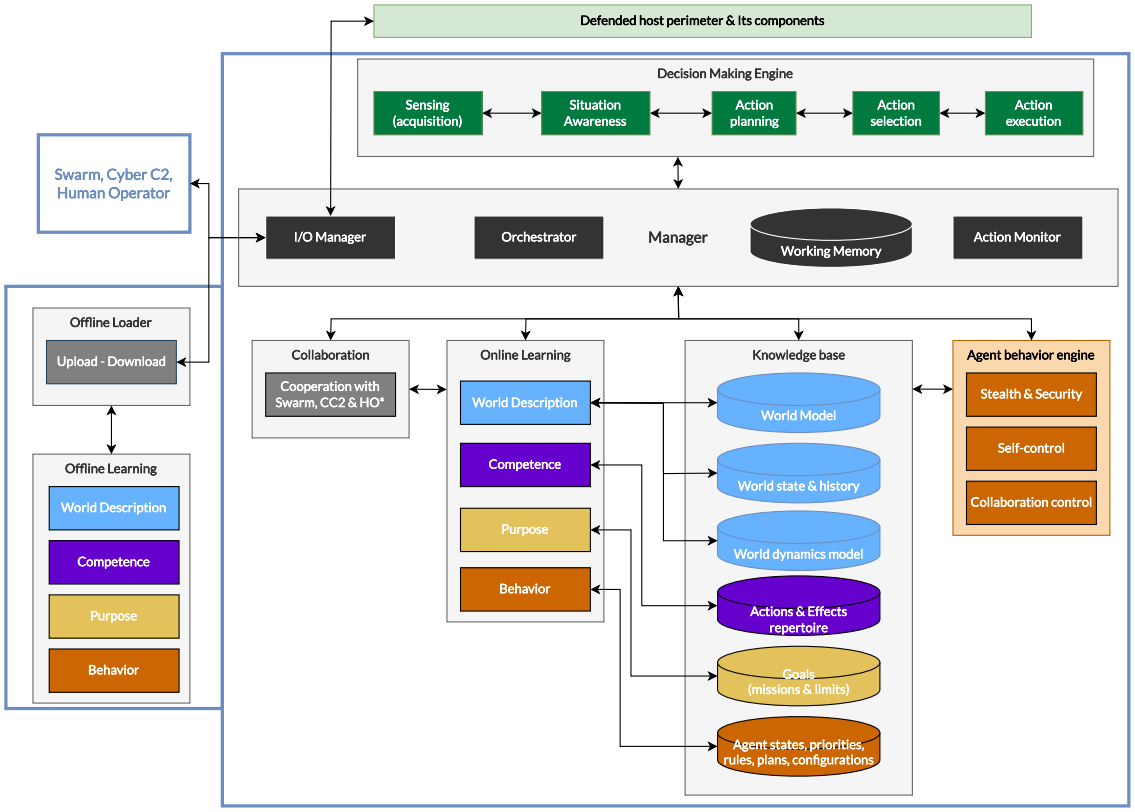


FIGURE 5 : Description de l'architecture modulaire MASCARA (tirée de [42])

d'un paradigme distribué, dans lequel plusieurs agents interconnectés coopèrent, s'auto-organisent et se réorganisent dynamiquement en fonction des besoins opérationnels [159, 178]. Dans cette perspective, l'idée a émergé de transformer chaque module de l'architecture MASCARA en un agent autonome, chargé d'exécuter une tâche spécifique de Cyberdéfense. Ces micro-agents, en interagissant de manière coordonnée, permettraient de mieux répartir les responsabilités, de renforcer la robustesse du système et d'atteindre collectivement les objectifs globaux de protection.

La notion de granularité permet ici d'opérer une distinction entre les micro-agents, chacun dédié à une fonction bien définie, et les agents AICA dits complets, capables de couvrir l'ensemble des missions prévues par le modèle. Dans la suite de ce manuscrit, nous emploierons le terme générique d'agent AICA pour désigner ces deux types d'agents, en précisant systématiquement le périmètre fonctionnel concerné selon le contexte.

Plus généralement, cette approche distribuée et coopérative s'inscrit dans une vision systémique de la Cyberdéfense, dans laquelle un ensemble d'agents autonomes interagit au sein du réseau pour garantir une sécurité globale, adaptative et résiliente. Cette orientation conceptuelle est soutenue par plusieurs travaux récents appartenant au domaine récent de l'ACD [50] (un sous-domaine de l'ACO) et constitue un fondement essentiel des développements présentés dans ce manuscrit.

Par exemple, des simulations ont montré que des équipes d'agents défensifs sont capables de surpasser un agent unique en termes de couverture du réseau et de réactivité, grâce à une coordination dynamique et distribuée [26]. Des plateformes telles que CybORG [55] illustrent également cette tendance, en proposant un environnement simulé dans lequel des agents autonomes décentralisés défendent simultanément un réseau face à des attaques coordonnées.

L'approche coopérative permet non seulement de protéger plusieurs hôtes en parallèle, mais aussi de détecter des attaques synchronisées et de s'adapter dynamiquement aux changements de topologie du réseau. Elle pose ainsi les bases d'une Cyberdéfense distribuée, proactive et évolutive, en rupture avec les architectures défensives traditionnelles.

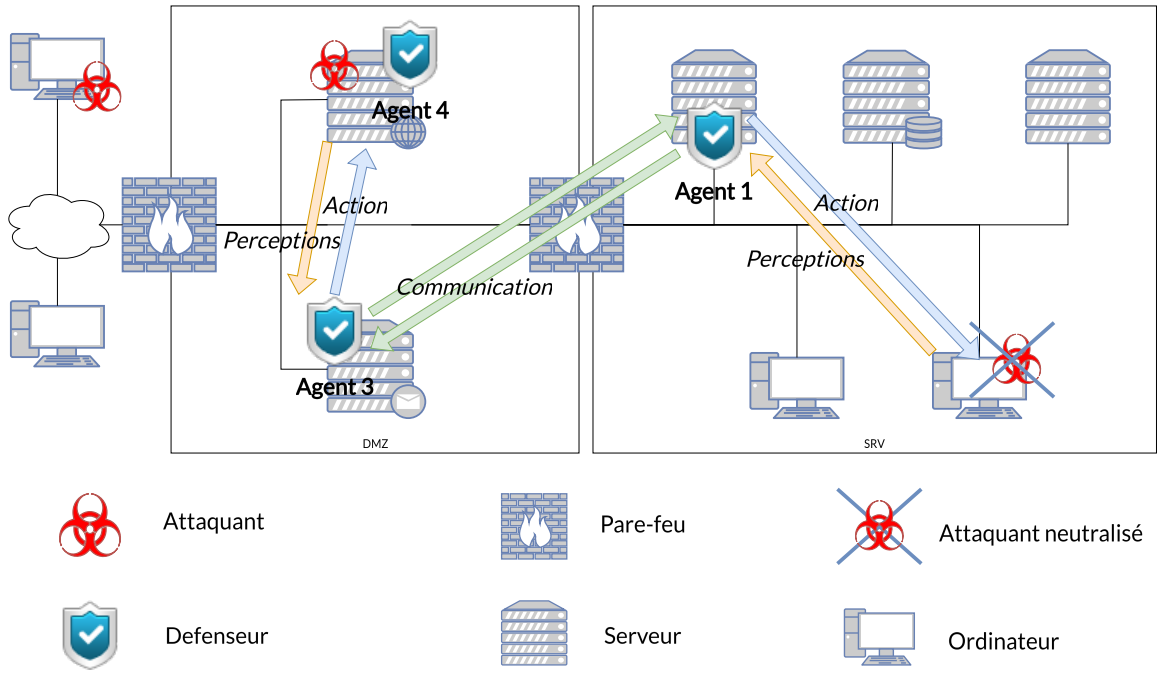


FIGURE 6 : Illustration schématique d'un SMA de Cyberdéfense dans une infrastructure d'entreprise jouet

Finalement, c'est dans ce contexte que l'idée d'un SMA de Cyberdéfense, telle qu'illustrée dans Figure 6, apparaît comme une alternative générique prometteuse par rapport aux approches centralisées existantes. Nous présentons dans la section suivante les bases conceptuelles des SMAs en vue de la conception d'un SMA de Cyberdéfense.

1.3 LA PISTE D'UNE VISION MULTI-AGENT

Les SMAs constituent un paradigme central de l'IA distribuée. Ils permettent de concevoir des systèmes complexes à partir d'agents autonomes interagissant dans un environnement partagé. Ces agents peuvent percevoir, raisonner, décider et agir de manière coordonnée pour résoudre des problèmes collectifs [174, 178].

Définitions fondamentales

Un **agent** est une entité autonome, physique ou logicielle, capable de percevoir son environnement, de prendre des décisions et d'agir pour atteindre des objectifs [153]. Un SMA regroupe plusieurs de ces agents, illustrés dans la Figure 7, qui coopèrent ou interagissent au sein d'un environnement généralement dynamique et partiellement observable [162, 184]. Chaque agent dispose d'une *zone d'observation locale* (disque pointillé) lui permettant de percevoir uniquement une partie de l'environnement et d'autres agents. À partir de ces observations partielles, et selon des **stratégies** ou **politiques** (schéma en haut de la figure), il sélectionne et exécute des **actions** dirigées vers des composants de l'environnement (carrés) ou vers d'autres agents (flèches pleines). Les agents peuvent également *échanger des*

messages (flèches en tirets) afin de coordonner leurs comportements. L'échange entre d'un agent à un autre est considéré comme l'application d'une action qui modifie les observations prochaines du second agent. L'objectif global (en haut à droite) représente un état souhaité de l'environnement que les agents cherchent à atteindre collectivement.

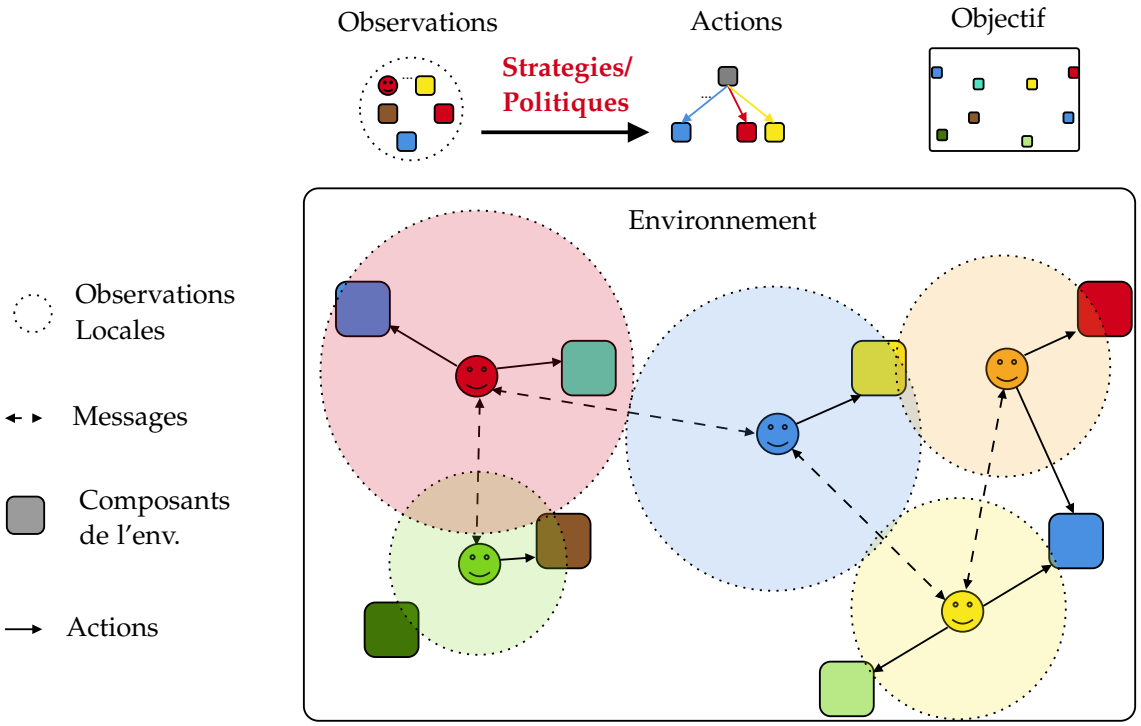


FIGURE 7 : Exemple schématique d'un SMA

Concepts structurants : autonomie, coordination et organisation

Dans le cadre de la conception d'un SMA de Cyberdéfense, trois concepts fondamentaux structurent notre approche : l'autonomie, la coordination et l'organisation.

L'**autonomie** désigne la capacité d'un agent à percevoir son environnement, à prendre des décisions et à agir sans contrôle externe immédiat [153, 169]. Elle implique un *découplage entre agents* et une *décentralisation du processus global de décision*, chaque agent étant responsable de ses propres choix et actions. Cette autonomie peut s'exprimer de manière réactive (par simple mécanisme stimulus-réponse), délibérative (par raisonnement et planification), ou sous forme hybride, combinant ces deux dimensions [192]. Dans le contexte de la Cyberdéfense, l'autonomie permet à un agent de détecter une anomalie, d'évaluer la situation localement, et de déclencher une contre-mesure sans devoir systématiquement remonter à une autorité centrale.

La **coordination** renvoie aux mécanismes par lesquels les agents gèrent leurs interdépendances pour coopérer efficacement [175, 182, 188]. Ces mécanismes incluent la négociation, la planification conjointe, l'allocation de tâches, ou encore l'engagement social. Dans un environnement de Cyberdéfense distribué, une coordination robuste est essentielle pour éviter les conflits d'intervention entre agents, assurer la cohérence des réponses, ou encore répartir dynamiquement les rôles selon les situations.

L'**organisation**, désigne la structure sociale dans laquelle les agents s'inscrivent : distribution des rôles, répartition des responsabilités, gestion des dépendances et des interac-

tions. Deux visions coexistent dans la littérature [157] et sont synthétisées dans la Figure 8. D'un côté, l'organisation peut être *explicite et réorganisable* (top-down), où des agents manipulent consciemment une spécification organisationnelle formelle, en adaptant les rôles ou les relations sociales selon les besoins. De l'autre, l'organisation peut être *émergente et auto-organisée* (bottom-up), lorsque les agents interagissent localement, sans représentation globale de la structure, et font émerger collectivement une organisation via des mécanismes décentralisés [166, 179].

Ces deux dynamiques d'adaptation organisationnelle peuvent être formalisées par les définitions suivantes. La **réorganisation** est un processus d'adaptation déclenché lorsque l'organisation en place ne permet plus de satisfaire les objectifs du système. Elle peut être initiée par un agent ou un concepteur externe, et repose sur la manipulation explicite de primitives organisationnelles telles que les rôles, les dépendances ou les règles d'interaction. Les agents sont alors conscients de l'organisation et capables de la modifier pour assurer un comportement collectif adéquat [157].

L'**auto-organisation** est un processus émergent, strictement endogène, dans lequel les agents ne possèdent qu'une connaissance locale. En réagissant à la pression environnementale et en interagissant avec leurs voisins, ils modifient indirectement la configuration globale du système (topologie, voisinages, différenciation fonctionnelle) sans recourir à une modélisation explicite [157].

Ces deux dynamiques peuvent être vues comme deux extrémités d'un continuum. Elles incarnent deux modalités d'un même processus général d'adaptation organisationnelle : détecter une inadéquation structurelle et y remédier. Tandis que l'auto-organisation privilégie une adaptation implicite, distribuée et ascendante, la réorganisation repose sur des mécanismes explicites, souvent planifiés, pouvant être centralisés ou non.

1.4 DE LA CONCEPTION D'UN SMA DE CYBERDÉFENSE

Les SMAs, et en particulier ceux inspirés du modèle AICA, offrent un cadre pertinent pour répondre à ces exigences. Cependant, la simple mise en œuvre d'agents autonomes ne suffit pas : c'est l'ensemble du processus de conception qui doit être repensé pour garantir une Cyberdéfense efficace et soutenable. La question de recherche centrale que nous adressons dans cette thèse peut donc être formulée ainsi :

> *Comment concevoir un SMA de Cyberdéfense capable d'atteindre ses objectifs de défense de manière satisfaisante, tout en s'auto-organisant pour s'adapter dynamiquement aux contraintes de l'environnement et aux exigences de conception ?*

Cette thèse vise à déterminer comment concevoir un SMA de Cyberdéfense capable d'atteindre ses objectifs de manière satisfaisante, tout en s'adaptant dynamiquement à son environnement et aux exigences de conception. Les objectifs de Cyberdéfense correspondent ici aux finalités opérationnelles assignées au système, telles que la détection des intrusions, la neutralisation des menaces, la restauration des services ou l'amélioration continue des capacités défensives, en référence au modèle P3R3 [144] (détecter, répondre, restaurer). Les contraintes de l'environnement englobent l'ensemble des limitations et caractéristiques du contexte d'opération du SMA : topologie du réseau, ressources disponibles, dynamique des menaces, exigences de temps réel, politiques de sécurité, etc. Enfin, les exigences de conception désignent les critères et attentes fixés lors de la conception du SMA, incluant l'autonomie distribuée, la résilience, l'explicabilité, la mesurabilité des

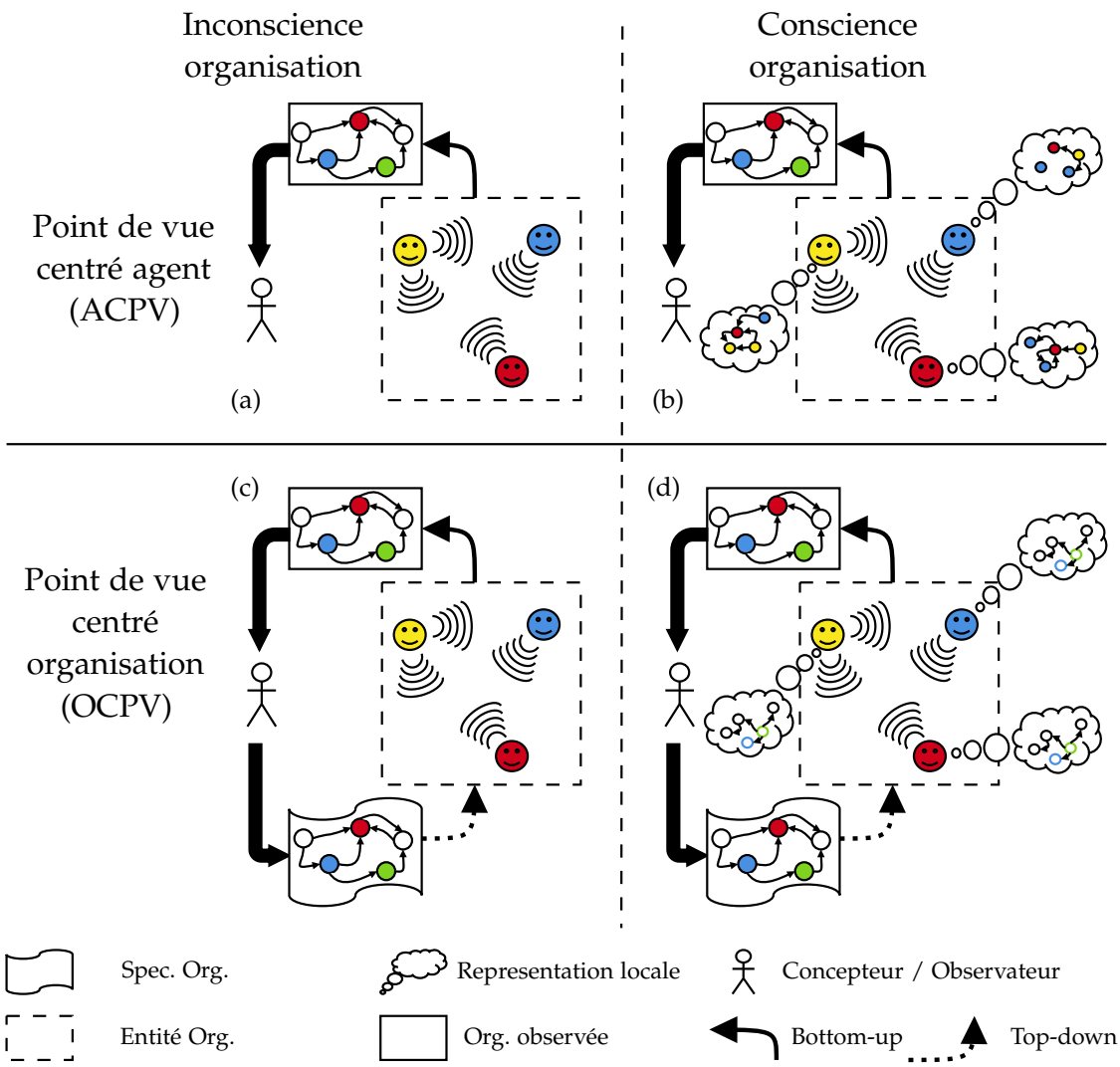


FIGURE 8 : Vue synthétique de l'organisation : (a) SMA émergents ; (b) SMA basés sur les coalitions ; (c) Ingénierie orientée agent ; (d) SMA orientés organisation (tirée de [157])

performances, la conformité réglementaire, ainsi que le coût de développement et de maintenance. La problématique consiste donc à articuler ces trois dimensions pour concevoir un SMA de Cyberdéfense à la fois efficace, adaptable et soutenable.

Il ne s'agit pas seulement de doter un SMA de capacités de détection ou de réponse, mais possiblement d'imaginer un cadre de conception où l'organisation des agents elle-même devient un levier d'efficacité et de résilience [165, 166]. Ce cadre doit répondre à un ensemble de critères clés, présentés ci-dessous, qui structurent notre réponse scientifique à cette question.

Critères pour l'évaluation des SMA de Cyberdéfense

Inspirés par les défis posés par l'*AICA IWG* [3], nous avons défini un ensemble de critères, résumé en Table 1, décrivant les propriétés qu'un SMA de Cyberdéfense doit satisfaire pour répondre à la question de recherche globale.

3 En particulier les défis de recherche liés à *Infrastructure Architecture and Engineering* et *Individual & Collective Decision Making*. Voir https://www.aica-iwg.org/research-challenges/

TABLE 1 : Grille des critères d'évaluation d'un SMA de Cyberdéfense

Critère	Résumé des attentes
C1 – Autonomie	Le SMA doit fonctionner avec un minimum d'interventions humaines, que ce soit lors de sa conception, de son implémentation, de son fonctionnement ou de son arrêt. L'objectif est de réduire la dépendance à l'opérateur en favorisant l'auto-organisation et l'autodécision des agents.
C2 – Performance	Le SMA doit démontrer des performances suffisantes pour atteindre les objectifs fixés. Cela inclut la maximisation des récompenses collectives, la capacité à converger vers des comportements stables.
C3 – Adaptation	Le SMA doit être capable de s'ajuster à des environnements dynamiques ou incertains. Les agents doivent pouvoir maintenir un niveau de performance acceptable face à des perturbations, des changements d'objectifs ou des situations non rencontrées à l'entraînement.
C4 – Contrôle	Le SMA doit rester contrôlable à différents niveaux : individuel et collectif. Cela suppose la possibilité d'imposer ou de modifier des spécifications organisationnelles et de garantir la cohérence globale du comportement des agents avec ces spécifications organisationnelles.
C5 – Explicabilité	Le SMA doit fournir des comportements compréhensibles pour un observateur humain. Les actions des agents et leurs coordinations doivent pouvoir être reliées à des spécifications organisationnelles comme des rôles, missions ou objectifs pour interpréter la logique de défense déployée.

C1 – Autonomie. L'autonomie constitue une propriété fondamentale attendue d'un SMA de Cyberdéfense. Elle ne se limite pas à la simple capacité d'un agent à prendre une décision localement, mais englobe l'ensemble du cycle de vie du SMA, depuis sa conception et son implémentation jusqu'à son exécution opérationnelle et son arrêt. Un SMA autonome doit ainsi réduire au minimum la dépendance vis-à-vis des opérateurs humains, en favorisant l'auto-organisation et la prise de décision décentralisée. Cette exigence est particulièrement cruciale dans des environnements où la vitesse de réaction est déterminante et où une supervision humaine constante se révèle impraticable. L'autonomie est donc envisagée comme une condition nécessaire pour accroître la réactivité, limiter la charge cognitive des analystes et garantir une continuité de fonctionnement face à des menaces rapides et distribuées.

C2 – Performance. La performance désigne la capacité du SMA à atteindre de manière robuste et efficace les objectifs de Cyberdéfense qui lui sont assignés. Elle se manifeste à travers la maximisation des récompenses collectives obtenues lors des expérimentations, mais également par la convergence vers des comportements stables et reproductibles. Un SMA performant doit être capable non seulement de détecter et de neutraliser des attaques, mais aussi de maintenir des performances satisfaisantes sur des exécutions répétées, sans instabilité prolongée ni oscillations comportementales. Dans ce contexte, la performance est évaluée non pas comme une réussite ponctuelle, mais comme une propriété durable et généralisable, garantissant l'efficacité du système au-delà d'un scénario unique.

C3 – Adaptation. L'adaptation reflète la capacité du SMA à ajuster ses comportements face à des environnements dynamiques, incertains ou instables. Elle suppose que les agents puissent absorber des perturbations, répondre à des situations inédites, et retrouver rapidement un niveau de fonctionnement acceptable après une dégradation de per-

formance. L'adaptation inclut également la résilience face aux changements d'objectifs ou de conditions opérationnelles, tels que des variations de topologie réseau, l'apparition de nouvelles tactiques adverses ou des contraintes de ressources fluctuantes. Elle se conçoit dès lors comme une propriété essentielle à la soutenabilité du système, garantissant que le SMA ne reste pas figé dans une logique apprise, mais demeure capable de maintenir une efficacité opérationnelle dans la durée.

C4 – Contrôle. Le contrôle renvoie à la possibilité de guider et de piloter le comportement du SMA, que ce soit au niveau individuel ou collectif. Un SMA contrôlable doit permettre l'imposition, la modification et le suivi de spécifications organisationnelles, telles que les rôles, missions ou règles d'interaction, tout en assurant la cohérence globale des comportements observés. Ce critère garantit ainsi un alignement continu entre les intentions de conception et les dynamiques effectives du système en fonctionnement. Le contrôle représente une exigence de sûreté et de confiance, en offrant aux concepteurs et aux opérateurs les moyens de limiter les comportements indésirables et de s'assurer que le SMA agit conformément aux politiques de sécurité définies. Il établit de ce fait un équilibre nécessaire entre autonomie locale et gouvernance externe.

C5 – Explicabilité. L'explicabilité vise à assurer que les comportements du SMA, qu'ils soient individuels ou collectifs, puissent être compris et interprétés par un observateur humain. Dans le cadre de la Cyberdéfense, où la transparence et la confiance conditionnent l'adoption de solutions automatisées, il est essentiel que les actions des agents soient reliées à des spécifications organisationnelles tangibles, telles que des rôles, missions ou objectifs intermédiaires. L'explicabilité ne se limite pas à rendre compte de l'action ex post, mais inclut également l'évaluation du degré de similarité entre les structures spécifiées et celles qui émergent effectivement lors de l'exécution. Elle constitue ainsi un vecteur de lisibilité et de diagnostic, permettant à la fois de renforcer la confiance des opérateurs et d'améliorer la conception future des SMAs par une meilleure compréhension des dynamiques émergentes.

1.5 BILAN

En synthèse, ce chapitre a permis de poser les bases conceptuelles et méthodologiques de la thèse, en identifiant les enjeux, critères et tensions qui structurent la conception des SMA de Cyberdéfense. Il a mis en évidence la nécessité d'une approche intégrée, capable de concilier performance, adaptation, contrôle et explicabilité, et a proposé de formaliser la conception comme un problème d'optimisation sous contraintes. Cette reformulation a conduit à la décomposition du problème en quatre sous-problèmes fondamentaux, chacun associé à une activité clé du processus de conception. La suite du manuscrit s'attachera à approfondir ces sous-problèmes, en analysant les verrous identifiés dans la littérature et en présentant les contributions proposées pour y répondre, amorçant ainsi la transition vers l'état de l'art détaillé dans le chapitre suivant.

2 VERS DES SMAS DE CYBERDÉFENSE ET LEUR CONCEPTION

Ce chapitre situe notre démarche au regard des travaux existants à l'intersection des domaines de la Cyberdéfense, SMA. Il repose sur une revue de littérature menée au début de la thèse et mise à jour tout au long des travaux. Nous adoptons une lecture critique orientée par les cinq critères de conception (C1 à C5) identifiés au chapitre précédent. Cette lecture permet de structurer les contributions existantes selon leur degré de proximité avec la vision d'un SMA de Cyberdéfense.

Nous distinguons pour cela deux axes complémentaires. Le premier (Section 2.1) s'intéresse aux **SMAs existants appliqués à la Cyberdéfense**, en examinant les objectifs pris en charge, les types d'organisation, et les environnements de déploiement associés. Le second (Section 2.2) explore les **moyens existants pour concevoir ces SMAs et les SMAs obtenus**, qu'ils soient manuels ou automatisés, symboliques ou apprenants. Les travaux identifiés concernant l'idée d'agents autonomes ou même de défense distribuée doivent prendre en compte simultanément tous les critères identifiés précédemment.

2.1 LES SMAS DE CYBERDÉFENSE DANS LA LITTÉRATURE

Un **SMA de Cyberdéfense** demande de prendre en considération les enjeux de structuration, d'adaptabilité et de coordination dans un environnement critique et dynamique. Dans un premier temps, nous avons conduit une revue de littérature[1] des travaux pouvant se comparer aux SMAs de Cyberdéfense [47]. En particulier, pour chacun des travaux identifiés, nous avons étudié le rapport entre le mécanisme d'organisation adopté et les **objectifs de Cyberdéfense** impliquant la mise en œuvre d'une ou plusieurs des fonctions de Cyberdéfense tels que décrit dans le P3R3 [144] : (R1) Détection des intrusions et alertes, (R2) Application de contre-mesures et rétablissement minimal, (R3) Apprentissage post-attaque et amélioration continue. Cette première étude permet une revue des travaux identifiés au travers de la grille d'analyse fondée sur les critères identifiés en Section 1.4.

Dans un SMA de Cyberdéfense, plusieurs agents atteignent un objectif global de Cyberdéfense par un comportement collectif résultant de la réalisation de sous-objectifs individuels et/ou de mécanismes locaux [136]. Des exemples de tels sous-objectifs pourraient être la détection des intrusions, la mise en œuvre d'un plan de récupération, la restauration d'une image, redirection des ports, etc.

2.1.1 *Mécanismes organisationnels dynamiques*

L'autonomie de fonctionnement du SMA de Cyberdéfense, obtenue en déléguant aux agents certaines missions avec peu d'interventions directes, est une réponse face aux charges de travail des équipes cyber et à la rapidité des cyber-attaques [84]. Un tel SMA doit modifier sa structure et les relations entre ses agents pour continuellement s'adapter à son environnement [76]. La réorganisation et l'auto-organisation sont alors des mécanismes clés [157].

En considérant un point de vue centré organisation, la Cyberdéfense globale est une tâche commune partagée par tous les agents à travers leur organisation. La **réorganisation**

TABLE 2 : Un aperçu de quelques organisations et des environnements hôtes utilisés dans les SMAs de Cyberdéfense étudiés

Organisation	**Avantages principaux**	**Inconvénients principaux**	**Environnement**	**Travaux**
Centralisé	Haute précision pour l'analyse de la situation	SPOF, manque de scalabilité	Petit à moyenne taille, non ouvert, petite entreprise	[120, 139, 170]
Hiérarchique (distribué)	Évolutivité, décomposition des tâches	Perte d'informations, goulots d'étranglement, retards	Taille moyenne à grande, ouvert, peu de variations	[155, 156]
Décentralisé (Peer-to-Peer)	Structure non définie a priori, Hautement adaptatif	Contrôle de l'organisation limitée, intensité de communication	Ouvert, toute taille, fortes variations	[97, 135, 149]
Coalition	Optimisation de l'organisation autour des tâches	Peu adapté sur le long terme	Toute taille, ouvert, peu de variations, peu de ressources	[147]
Équipes	Bonne performance pour des tâches régulières	Haute intensité de communication	Ouvert, hétérogène, toute taille, peu de variations	[101]
Marché	Organisation optimisée par concurrence, bonne gestion des agents	Processus d'allocation complexe et long	Toute taille, ouvert, peu de variations, peu de ressources	[66]

est un moyen de basculer entre plusieurs organisations éprouvées qui semblent adaptées dans des circonstances données [157].

En considérant un point de vue centré agent, l'**auto-organisation** est définie comme un processus ascendant où l'organisation émerge des interactions et des actions locales des agents. La Cyberdéfense globale résulte alors des actions de Cyberdéfense locales et des interactions pair à pair entre les agents [157]. L'auto-organisation semble être un des moyens à mobiliser pour faire face aux cyber-menaces en évitant les écueils d'un contrôle centralisé.

2.1.2 *Organisations des SMAs de Cyberdéfense*

Le choix d'une organisation de SMA de Cyberdéfense implique de tenir compte des relations entre les objectifs de Cyberdéfense et l'environnement de déploiement. L'analyse des SMAs de Cyberdéfense disponibles est susceptible d'indiquer des tendances pour ces relations. Cela permettrait de favoriser la mise en œuvre d'une organisation à partir des objectifs de Cyberdéfense et l'environnement de déploiement.

Notre revue de littérature s'est concentrée sur le rapprochement des notions des SMAs et de la Cyberdéfense.

Pour chacun des travaux de SMA de Cyberdéfense, nous nous sommes intéressés aux fonctions de Cyberdéfense couvertes. Un aperçu de cette classification est proposé en

Table 3. Nous avons constaté que la plupart des objectifs de Cyberdéfense des SMAs se concentrent principalement sur la détection d'anomalies et d'intrusions (plus de 50% des travaux de notre revue complète se focalisent ainsi sur la fonction R1).

TABLE 3 : Un aperçu des fonctions de Cyberdéfense prises en charge par les SMA de Cyberdéfense étudiés

Objectifs principaux	**Travaux**
R1 : détection d'intrusion, surveillance du réseau, détection de menaces possibles	[101, 120, 139, 155, 156, 170]
R2 : application de contre-mesures, contrôles d'accès, correctifs de Cyberdéfense, stratégies de Cyberdéfense	[101, 155, 156]
R3 : investigations forensiques, élaboration de contre-mesures adaptées, apprentissage des cyber-attaques, adaptation aux cyber-attaques	[66, 97, 135, 149]

Pour chacun de ces mêmes travaux, nous nous sommes aussi intéressés aux caractéristiques principales de l'organisation et de l'environnement de déploiement. Une synthèse de ce travail est présentée en Table 2. Nous constatons qu'indépendamment des objectifs de Cyberdéfense, l'organisation centralisée et/ou hiérarchique est la plus répandue parmi les SMAs de Cyberdéfense étudiés. La centralisation des données acquises de l'environnement, en un seul point, favorise de meilleures performances pour l'analyse de la situation globale et le contrôle du système de Cyberdéfense. Ces types d'organisation semblent moins facilement s'appliquer pour des réseaux dynamiques, mais sont répandus sur des systèmes de taille moyenne avec des contraintes connues [139].

Les organisations de type décentralisé tirent profit d'une approche davantage auto-organisée pour faire face aux cyber-menaces de façon à augmenter l'autonomie du SMA de Cyberdéfense comme proposée dans le « Artificial Immune System » [135] ou la « Ant-Based Cyber Security » [149] en sont des exemples. Elles sont néanmoins moins établies en tant que solutions génériques de Cyberdéfense et/ou cyber-sécurité.

2.1.3 *Revue critique*

Les travaux inventoriés mettent en évidence des familles d'organisation (centralisée, hiérarchique, décentralisée/essaim, coalitions, équipes, marché) dont les qualités et limites varient selon les critères, ainsi qu'une focalisation marquée sur **R1** (détection/surveillance) au détriment de **R2–R3** (réponse/reprise), déjà illustrée dans nos tableaux de synthèse (Table 2 et Table 3).

C1 – AUTONOMIE. L'autonomie locale de décision (percevoir, décider, agir au plus près des ressources) est généralement mieux servie par des organisations décentralisées (pair-à-pair, essaim), qui réduisent la dépendance à une autorité centrale et améliorent la réactivité. À l'inverse, les architectures centralisées/hiérarchiques facilitent l'orchestration et la mise en œuvre, mais conservent des points de défaillance et des goulots décisionnels, limitant l'autonomie systémique (conception, déploiement, arrêt). Ces arbitrages sont récurrents dans les états de l'art comparés.

C2 – PERFORMANCE. Les meilleures performances analytiques (agrégation d'événements, priorisation) sont souvent rapportées pour les organisations centralisées/hiérarchiques en contexte **R1**, tandis que la tenue en charge et la robustesse diminuent sous

dynamique adversariale ou ressources contraintes. Les formes décentralisées montrent une bonne agilité, mais peinent parfois à stabiliser des comportements collectifs efficaces sans mécanismes explicites de régulation. Un besoin méthodologique récurrent concerne la mesure transversale de la performance (récompense, *taux de convergence*, stabilité) sur des bancs partagés.

C3 – ADAPTATION. Deux leviers dominent : la réorganisation (top-down, rôles/relations explicites) et l'auto-organisation (bottom-up, structure emergente). La littérature illustre les deux (p. ex. essaims, systèmes immunitaires artificiels), mais manque d'un cadre expérimental unifié permettant d'attribuer l'adaptation à des choix organisationnels contrôlés (temps de rétablissement, sensibilité aux variations, résilience à de nouvelles TTP). D'où l'intérêt d'une grille générique appliquée à des environnements variés.

C4 – CONTRÔLE. Le contrôle (spécifier, imposer et ajuster des contraintes organisationnelles et en vérifier la *cohérence*) est naturellement favorisé par des architectures centralisées/hiérarchiques (auditabilité et gouvernance fortes), mais s'y oppose parfois à l'autonomie et à l'adaptation. Dans les organisations décentralisées, le contrôle requiert des garde-fous (règles locales, marchés, coalitions) pour éviter conflits et effets indésirables. La littérature souligne l'absence d'outillage commun pour mesurer, de façon reproductible, le respect des contraintes et la cohérence globale sous contraintes opérationnelles (latence, pertes de communications).

C5 – EXPLICABILITÉ. L'explicabilité *a priori* s'appuie sur des structures explicites (rôles, missions, hiérarchies) qui donnent une lecture claire des responsabilités ; l'explicabilité *a posteriori* des organisations émergentes (décentralisées, ré-/auto-organisées) demeure peu instrumentée : il manque des moyens pour relier trajectoires observées et adéquation organisationnelle (alignement entre organisation spécifiée et implicite). Cette lacune entrave la comparaison des approches et la confiance opérationnelle.

TABLE 4 : Lecture synthétique des organisation des SMAs de Cyberdéfense selon les critères C1–C5

Organisation	C1 Autonomie	C2 Performance	C3 Adaptation	C4 Contrôle	C5 Explicabilité
Centralisée	Réactivité locale limitée ; dépendance forte au centre (SPOF, goulots).	Très bonne agrégation et priorisation en R1 ; fragilité sous charge/adversarial.	Faible plasticité structurelle ; adaptation surtout par re-paramétrage central.	Gouvernance et audit forts ; imposabilité des règles globale élevée.	Forte lisibilité *a priori* (rôles/chaînes) ; faible lisibilité des effets émergents.
Hiérarchique (distribuée)	Autonomie partielle aux feuilles ; dépendances verticales persistantes.	Bon compromis analyse/latence ; dégradations aux interfaces de niveaux.	Réorganisation planifiée possible ; coûts de propagation des changements.	Bon levier de contrôle multi-niveaux ; cohérence inter-niveaux à surveiller.	Explications structurées par niveaux ; traçabilité ascendante.
Décentralisée / Essaim (p2p)	Haute autonomie locale et réactivité ; pas de SPOF.	Agilité en dynamique ; stabilisation globale délicate sans régulation.	Très bonne auto-organisation ; résilience aux pannes/communications partielles.	Contrôle diffus ; nécessité de garde-fous locaux et d'arbitrage.	Explicabilité *a posteriori* limitée ; besoin d'outils d'adéquation organisationnelle.
Coalitions (orientées tâches)	Autonomie par regroupements opportunistes.	Bonne performance sur tâches ciblées ; coût de formation/-dissolution.	Adaptation par re-composition rapide ; sensible aux signaux de contexte.	Contrats locaux gérables ; cohérence globale dépend de la coordination inter-coalitions.	Explicabilité locale correcte (contrats), globale plus difficile.
Équipes (rôles fixés)	Autonomie encadrée par rôles ; dépendance aux protocoles d'équipe.	Bonne performance sur routines ; communication intensive.	Adaptation par rotation/redéfinition de rôles ; inertie organisationnelle.	Contrôle aisé via rôles/protocoles ; cohérence vérifiable.	Explicabilité élevée via rôles/missions explicites.
Marché (allocation par enchères)	Autonomie individuelle forte ; décisions opportunistes.	Performance dépend des mécanismes d'allocation ; latence des enchères.	Bonne adaptation à la variabilité ; stabilité selon le design des incitations.	Contrôle via règles de marché ; régulation nécessaire pour la sûreté.	Explicabilité par traces d'allocation/coûts ; compréhension dépend du modèle.

Finalement, les organisations centralisées/hiérarchiques favorisent contrôle et explicabilité *a priori* mais limitent autonomie systémique et adaptation, tandis que les formes décentralisées/émergentes maximisent autonomie et adaptation au prix d'un contrôle plus diffus et d'une explicabilité *a posteriori* encore peu instrumentée. La littérature met en outre l'accent sur **R1**, laissant **R2–R3** moins explorées expérimentalement.

2.2 LES CADRES DE CONCEPTION DE SMA DE CYBERDÉFENSE DANS LA LITTÉRATURE

Dans cette section, nous nous intéressons aux cadres et environnements qui permettent la conception, l'entraînement ou l'évaluation de systèmes à agents pour la Cyberdéfense. Contrairement aux contributions centrées sur les architectures de SMA déjà établies, ces travaux visent à outiller le processus de développement de telles architectures, que ce soit par simulation, apprentissage ou formalisation.

2.2.1 *État des lieux et diversité des approches*

Dans cette sous-section, nous dressons un panorama des cadres et environnements existants qui permettent de concevoir ou d'entraîner des agents pour la Cyberdéfense. Nous distinguons notamment :

- les environnements de simulation permettant d'émuler des réseaux ou des comportements d'attaques/défenses (e.g., CybORG, *Network Attack Simulator & Emulator* – NASimEmu, *Emulation Laboratory* – EmuLab) ;
- les frameworks d'entraînement à base d'apprentissage automatique ou par renforcement, parfois multi-agents (e.g., *Cyber Security Learning Environment* – CSLE [39], *Automated Penetration Testing Using Deep Reinforcement Learning* – AutoPentest-DRL [65], *Cybersecurity ANomaly Detection via Learning and Evaluation System* – CANDLES [138]) ;
- les plateformes dédiées à l'optimisation de politiques d'attaque ou de défense (e.g., *Penetration Testing Gym* – PenGym [10], CSLE [39], *Cyber Battle Simulator* – CyberBattleSim).

Nous examinons également leur niveau d'abstraction, le type d'interaction permis, et leur capacité à servir de support à une démarche de conception.

Ces environnements présentent des profils variés qui sont résumés dans la Table 5. Certains visent à fournir une simulation réaliste pour le test de politiques (e.g., NASimEmu, EmuLab), tandis que d'autres s'orientent vers la génération automatique ou la co-évolution de comportements (e.g., CLAP, CANDLES). Le degré d'autonomie accordé aux agents, l'intégration de modèles d'attaquants adaptatifs, et le support pour des politiques multi-agents diffèrent selon les cadres.

L'intérêt d'une telle diversité réside dans la complémentarité des environnements. Toutefois, il en découle aussi une fragmentation des outils, rendant difficile l'établissement d'une méthode de conception généralisable ou réutilisable. Nous verrons dans les sections suivantes comment ces limites soulignent la nécessité d'un cadre unificateur structurant le processus de conception de SMA de Cyberdéfense.

TABLE 5 : Exemples de cadres de conception pour agents de Cyberdéfense

Nom	Type	Fonction principale	Travaux
CybORG	Environnement simulé	Entraînement de défenseurs/attaquants dans des réseaux virtuels	[74]
NASimEmu	Simulateur + émulateur	Simulation fine d'attaques réseau pour généralisation de politiques	[23]
CSLE	Framework RL multi-agent	Conception d'agents défensifs avec visualisation de politiques apprises	[57]
AutoPentest-DRL	Entraînement DRL	Génération automatique de scénarios d'attaque par DRL	[65]
EmuLab	Plateforme de test réseau	Reproduction de scénarios réalistes sur réseaux virtualisés SDN	[137]
CLAP	Framework DRL multi-objectif	Entraînement d'agents à objectifs multiples en attaque automatique	[35]
CyberBattleSim	Simulateur pour agents RL	Évaluation de politiques de défense/attaque dans un graphe de vulnérabilités	[75]
CANDLES	Cadre évolutionniste	Co-évolution d'agents défensifs et offensifs dans un réseau simulé	[138]
PenGym	Environnement RL	Test d'outils de pentesting avec renforcement adaptatif	[111]
ASAP	Plateforme d'analyse automatique	Analyse de vulnérabilités par agents autonomes	[80]

2.2.2 *Couverture des critères de conception (C1 à C5)*

Cette sous-section présente une étude systématique des environnements précédents au regard des cinq critères exposés dans la Section 1.4. Le Table 6 synthétise cette étude.

Quelques cas emblématiques sont détaillés ci-dessous pour illustrer les forces et limites principales de ces environnements.

CSLE. Ce framework multi-agent met l'accent sur l'entraînement et la visualisation des politiques de défense. Il offre une bonne couverture du critère **C2 Performance** grâce à des outils permettant de comparer des agents et de mesurer leurs récompenses cumulées. Sa valeur ajoutée est également visible sur **C5 Explicabilité**, où une interface de visualisation permet d'expliquer les politiques apprises. En revanche, **C1 Autonomie** et **C4 Contrôle** restent partiellement couverts, car le guidage organisationnel explicite et la gouvernance par contraintes sont peu pris en charge.

TABLE 6 : Couverture des cadres de conception de SMA de Cyberdéfense au regard des critères C1–C5

Cadre	C1 Autonomie	C2 Performance	C3 Adaptation	C4 Contrôle	C5 Explicabilité
CybORG	Partielle (agents simulés, mais guidage humain requis)	Bonne (large gamme de scénarios)	Partielle (attaquants adaptatifs, défenseurs limités)	Faible (peu de contraintes explicites)	Faible (trajectoires peu interprétables)
NASimEmu	Partielle	Bonne (finesse de simulation)	Partielle (variantes de topologie)	Faible	Faible
CSLE	Partielle	Bonne (multi-agents RL)	Partielle (adaptation limitée aux environnements fixes)	Partielle (contrôle via paramètres)	Partielle (visualisation des politiques)
AutoPentest-DRL	Faible (centré attaquant)	Bonne (génération efficace de scénarios)	Faible (adaptation limitée côté défense)	Faible	Faible
EmuLab	Faible (agents peu autonomes)	Bonne (réalisme réseau)	Faible (adaptation manuelle)	Faible	Faible
CLAP	Partielle	Bonne (objectifs multiples)	Partielle (co-évolution des stratégies)	Faible	Faible
CyberBattleSim	Partielle	Bonne (comparaison de politiques RL)	Partielle (diversité de graphes vulnérabilités)	Faible	Faible
CANDLES	Partielle	Bonne (co-évolution agents défense/attaque)	Bonne (adaptation par évolution)	Faible	Faible
PenGym	Faible	Partielle (ciblé pentesting)	Faible	Faible	Faible
ASAP	Faible	Partielle (analyse automatisée)	Faible	Faible	Faible

Échelle indicative : *Faible* = non adressé ou très limité ; *Partielle* = couvert en partie, selon scénarios ; *Bonne* = support explicite et généralisable.

CYBORG. L'un des environnements les plus utilisés dans la communauté ACD, CybORG offre une bonne couverture du critère **C2 Performance** via une large gamme de scénarios attaquants et défenseurs. Il permet également une exploration limitée de **C3 Adaptation** grâce à l'inclusion d'attaquants adaptatifs. Toutefois, **C1 Autonomie** demeure partielle (les agents simulés nécessitent souvent un guidage externe), tandis que **C4 Contrôle** et **C5 Explicabilité** sont très peu supportés, faute d'outils pour imposer des contraintes organisationnelles ou rendre les politiques intelligibles.

NASIMEMU. Ce simulateur hybride émulation/simulation est particulièrement riche pour évaluer la **performance** (C2) et la robustesse des politiques à travers des topologies variées. Il permet aussi une certaine **adaptation** (C3) en changeant la structure du réseau. En revanche, comme beaucoup d'environnements de simulation, l'**autonomie** (C1), le **contrôle** (C4) et l'**explicabilité** (C5) y sont peu représentés.

CANDLES. Ce cadre propose une approche pour la **co-évolution** d'agents défensifs et offensifs. Il obtient ainsi une couverture notable de **C3 Adaptation**, en testant des comportements émergents dans des contextes variés. Les critères **C1 Autonomie** et **C2 Performance** sont partiellement ou bien couverts, mais les dimensions **C4 Contrôle** et **C5 Explicabilité** ne sont pas intégrées dans sa méthodologie.

Les cadres actuels permettent surtout de démontrer des gains de performance dans des scénarios contrôlés, mais ils n'offrent pas encore les moyens d'évaluer et de garantir l'autonomie, le contrôle ou l'explicabilité des SMA de Cyberdéfense. Ces constats renforcent la nécessité d'un cadre de conception unificateur qui prenne explicitement en compte l'ensemble des critères C1–C5, afin de guider la conception, l'entraînement et l'analyse des agents au-delà de la seule performance brute.

2.2.3 *Limites actuelles et besoins non adressés*

Globalement, cette revue met en évidence que :

- **La performance (C2) est le critère le mieux adressé.** La majorité des cadres privilégient l'évaluation de politiques en termes de récompense cumulée, de succès dans des scénarios donnés ou de robustesse face à quelques variations. Cette focalisation traduit la prépondérance de l'apprentissage par renforcement comme méthode d'entraînement et d'évaluation, mais elle reste insuffisante pour caractériser la soutenabilité à long terme des SMA de Cyberdéfense.
- **L'autonomie (C1) demeure incomplète.** Si plusieurs environnements simulent des agents capables de percevoir et d'agir, le cycle de vie complet du SMA reste largement dépendant d'interventions humaines. L'autonomie organisationnelle et la capacité à limiter durablement la supervision externe sont donc encore peu explorées.
- **L'adaptation (C3) est abordée de manière partielle.** Certains cadres intègrent la co-évolution (e.g., CANDLES) ou des attaquants adaptatifs (e.g., CybORG), mais les mécanismes permettant une adaptation systématique et généralisable des politiques aux dynamiques complexes de l'environnement restent absents. Les temps de rétablissement, la résilience aux changements de topologie ou aux nouvelles tactiques adverses ne sont que rarement mesurés.
- **Le contrôle (C4) est très peu représenté.** Aucun cadre ne fournit de mécanismes explicites pour spécifier, imposer et vérifier le respect de contraintes organisationnelles (rôles, missions, règles d'interaction). Le contrôle est soit absent, soit réduit à quelques paramètres ajustables manuellement, ce qui limite la capacité à orienter les comportements collectifs en fonction de besoins opérationnels.
- **L'explicabilité (C5) est le critère le moins couvert.** Hormis quelques visualisations de politiques (e.g., CSLE), la littérature ne propose pas d'outils pour relier les comportements observés aux structures organisationnelles attendues ni pour analyser l'adéquation entre organisation spécifiée et organisation implicite émergente. Cela entrave la confiance des opérateurs et la transférabilité des approches.

En résumé, l'état de l'art révèle une forte concentration des efforts sur la *démonstration de performance* dans des scénarios simulés, mais un déficit marqué sur les dimensions d'autonomie, de contrôle et d'explicabilité. Ces lacunes soulignent la nécessité de développer un cadre de conception unificateur, intégrant explicitement les cinq critères C1–C5 afin de permettre une évaluation transversale, généralisable et reproductible des SMA de Cyberdéfense.

2.3 UNE TENSION ENTRE APPROCHE SYMBOLIQUE ET CONNEXIONNISTE

La revue de littérature menée précédemment révèle une couverture très partielle des cinq critères de conception (C1 à C5) identifiés au Section 1.4. Plus précisément, aucun travail ne permet aujourd'hui de concevoir un SMA de Cyberdéfense pleinement conforme à la vision AICA, à la fois robuste, explicable, généralisable et automatisable.

Deux grandes familles d'approches de fonctionnement et de conception d'un SMA de Cyberdéfense se distinguent dans la littérature : les approches *symboliques*, centrées sur la modélisation explicite de l'organisation du système, et les approches *connexionnistes*, reposant sur des techniques d'apprentissage telles que le RL ou le MARL. Ces deux paradigmes

se situent aux pôles opposés de la conception des SMAs de Cyberdéfense, et présentent des forces et faiblesses complémentaires.

2.3.1 *Les approches symboliques*

Les approches symboliques, reposant sur des formalismes comme MOISE$^+$ [173] ou *Agents, Groups, Roles* (AGR) [167], offrent un cadre rigoureux pour structurer les rôles, les permissions et les obligations au sein d'un SMA. Ces modèles facilitent l'interprétabilité, la vérifiabilité, et la maîtrise de la structure organisationnelle (C4).

Cependant, ces approches souffrent d'une forte dépendance à l'expertise humaine (C1), d'une adaptabilité limitée aux environnements dynamiques (C3), et d'une faible généralisabilité. En pratique, la mise à l'échelle de ces modèles devient vite coûteuse et laborieuse [165].

2.3.2 *Les approches apprenantes*

Les approches par apprentissage, et notamment le MARL, permettent d'obtenir des politiques adaptatives dans des environnements complexes, partiellement observables et dynamiques. Elles offrent des garanties intéressantes en matière de performance (C2), de résilience (C3), de généralisabilité et d'automatisation (C1).

Néanmoins, ces approches posent de sérieux problèmes d'explicabilité (C5), de contrôlabilité (C4), et de sûreté en environnement réel (C4). Les politiques obtenues sont difficilement interprétables, ce qui complique leur adoption dans des domaines critiques comme la Cyberdéfense [94].

2.3.3 *Une opposition structurelle révélée par les critères*

La Figure 9 synthétise cette tension conceptuelle entre approches symboliques et apprenantes à travers leur capacité respective à satisfaire les cinq critères. On y observe un compromis inhérent entre la maîtrise (C4) et la performance opérationnelle (C2).

En fin de compte, les SMAs issus d'approches symboliques peuvent pleinement tirer parti de la **réorganisation** : ils disposent de structures explicites (rôles, missions, groupes, etc.) qu'ils peuvent manipuler de manière consciente, favorisant à la fois l'**explicabilité** et le **contrôle**. En effet, la reconfiguration organisationnelle s'appuie sur des représentations internes accessibles à l'analyse ou à la vérification formelle. De plus, ces SMAs peuvent également implémenter des mécanismes d'**auto-organisation**, notamment en utilisant un raisonnement symbolique. Bien que cette forme d'auto-organisation puisse introduire une certaine complexité dans les comportements, elle reste néanmoins interprétable, car fondée sur des règles ou structures modélisées explicitement.

À l'inverse, les SMAs produits par des approches **connexionnistes**, notamment via le MARL, reposent sur des **politiques implicites** encodées dans des réseaux de neurones. Ces politiques, apprises automatiquement à partir d'interactions environnementales, ne rendent pas directement compte de l'organisation du système. En particulier, la **granularité** des approches connexionnistes classiques est souvent réduite à des transitions observation-action, ce qui rend l'analyse organisationnelle difficile et accroît la complexité interprétative. Toutefois, cela n'empêche pas l'émergence de dynamiques auto-organisées : les agents peuvent collectivement adopter des structures ou des fonctions distribuées sans qu'elles aient été spécifiées ou modélisées *a priori*. Ces formes d'organisation sont cepen-

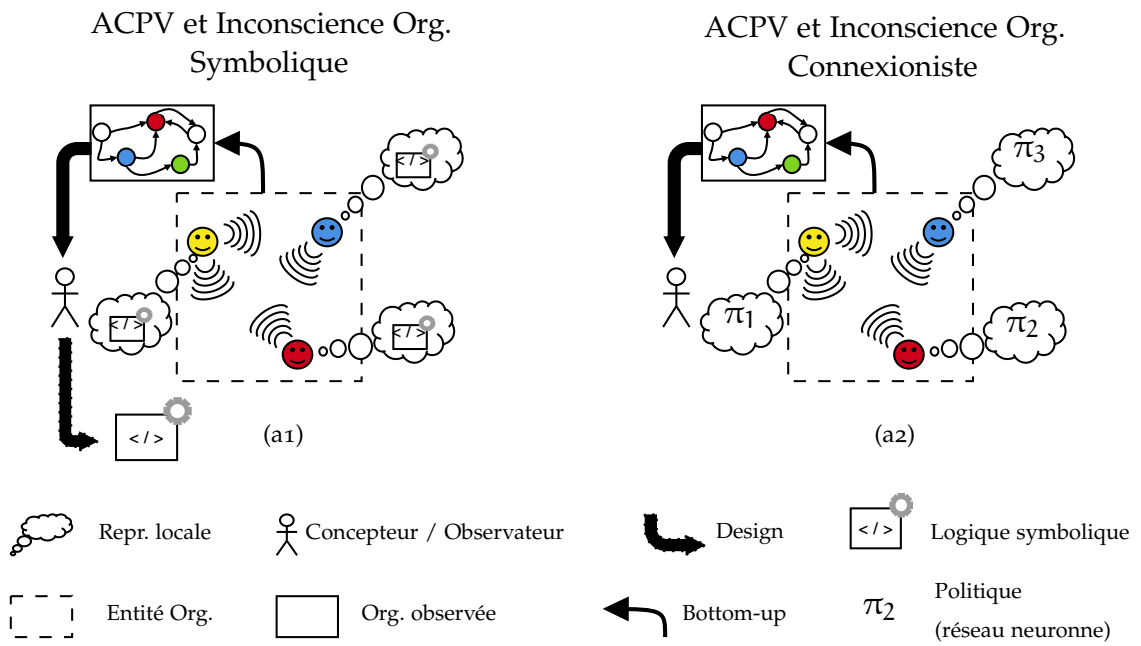


FIGURE 9 : Tension entre approches symboliques et apprenantes : couverture respective des critères C1 à C5

dant **non conscientes**, c'est-à-dire qu'elles ne sont pas manipulables ou accessibles au raisonnement des agents. Par conséquent, la **réorganisation explicite** est, en l'état, hors de portée dans ces SMAs connexionnistes, sauf à intégrer directement les spécifications organisationnelles dans le processus d'apprentissage, ce qui n'est pas le cas des approches MARL classiques.

Vers une approche intégrée

Ce constat nous conduit à envisager une approche hybride : une structuration organisationnelle explicite pour assurer le contrôle, la vérification et l'explicabilité ; couplée à des mécanismes d'apprentissage automatique pour assurer l'adaptabilité, l'optimisation et la montée en charge. C'est cette ambition que nous poursuivrons dans les chapitres suivants.

2.4 BILAN

Ce chapitre a mis en lumière la tension fondamentale entre approches symboliques et connexionnistes dans la conception des SMA de Cyberdéfense, chacune présentant des avantages et des limites au regard des critères d'autonomie, de performance, d'adaptation, de contrôle et d'explicabilité. Ce constat justifie la nécessité d'une approche intégrée, capable de concilier maîtrise organisationnelle et adaptabilité opérationnelle. La transition vers une formalisation du problème de conception sous la forme d'un problème d'optimisation sous contraintes ouvre ainsi la voie à une méthode hybride, articulant spécifications organisationnelles explicites et apprentissage automatique. Le chapitre suivant s'attachera à détailler cette reformulation, à identifier les sous-problèmes associés et à préciser les hypothèses de recherche qui guideront la suite du manuscrit.

3

UN PROBLÈME D'OPTIMISATION POUR STRUCTURER UNE MÉTHODE

La conception d'un SMA de Cyberdéfense, capable d'opérer dans un environnement dynamique, distribué et potentiellement hostile, suppose de satisfaire simultanément des exigences variées : autonomie, coordination, sûreté, adaptabilité, explicabilité, mesurabilité, résilience et soutenabilité.

Au vu des limites identifiées dans les approches existantes (symboliques ou apprenantes), nous proposons de formaliser cette conception comme un **problème d'optimisation sous contraintes**, où l'objectif global est maximisé tout en respectant des contraintes structurelles, environnementales et organisationnelles. Cette formalisation présente plusieurs intérêts majeurs. D'une part, elle permet de bénéficier de la puissance des méthodes connexionnistes, notamment l'apprentissage machine (ML) et le MARL, pour explorer efficacement de vastes espaces de solutions et optimiser les politiques des agents dans des environnements complexes et dynamiques. D'autre part, l'introduction explicite de contraintes structurelles, environnementales et organisationnelles offre la possibilité d'intégrer les apports des approches symboliques, en garantissant la conformité aux exigences de sûreté, d'explicabilité et de contrôle. Ainsi, le cadre d'optimisation sous contraintes agit comme un pont entre ces deux paradigmes : il autorise l'adaptabilité et la performance des approches apprenantes, tout en assurant la maîtrise et la robustesse des approches symboliques. Cette hybridation ouvre la voie à la conception de SMAs de Cyberdéfense à la fois efficaces, adaptatifs et interprétables, capables de répondre aux enjeux opérationnels et scientifiques identifiés.

Ce chapitre introduit cette reformulation et en déduit une décomposition en **sous-problèmes fondamentaux** qui structurent la suite du manuscrit.

3.1 FORMULATION DU PROBLÈME GLOBAL

3.1.1 *Enjeux de conception*

La conception d'un SMA de Cyberdéfense ne peut être envisagée comme un processus exclusivement humain (ingénierie symbolique), ni totalement automatique (apprentissage autonome). Dans un contexte marqué par l'incertitude, la complexité et la vitesse d'évolution des cybermenaces, une démarche hybride s'impose. Les critères C1 à C5 identifiés précédemment rappellent l'ampleur des exigences à satisfaire : **autonomie** (C1), **performance** (C2), **adaptation** (C3), **contrôle** (C4) et **explicabilité** (C5). Nous faisons donc le l'hypothèse générale d'une approche unificatrice, dans laquelle la conception d'un SMA est envisagée comme un processus guidé ou contraint par des spécifications organisationnelles traduisant des exigences de conception, au croisement de l'optimisation, de l'organisation et de méthodes de résolution automatique comme le MARL. Cette vision est illustrée dans la Figure 10.

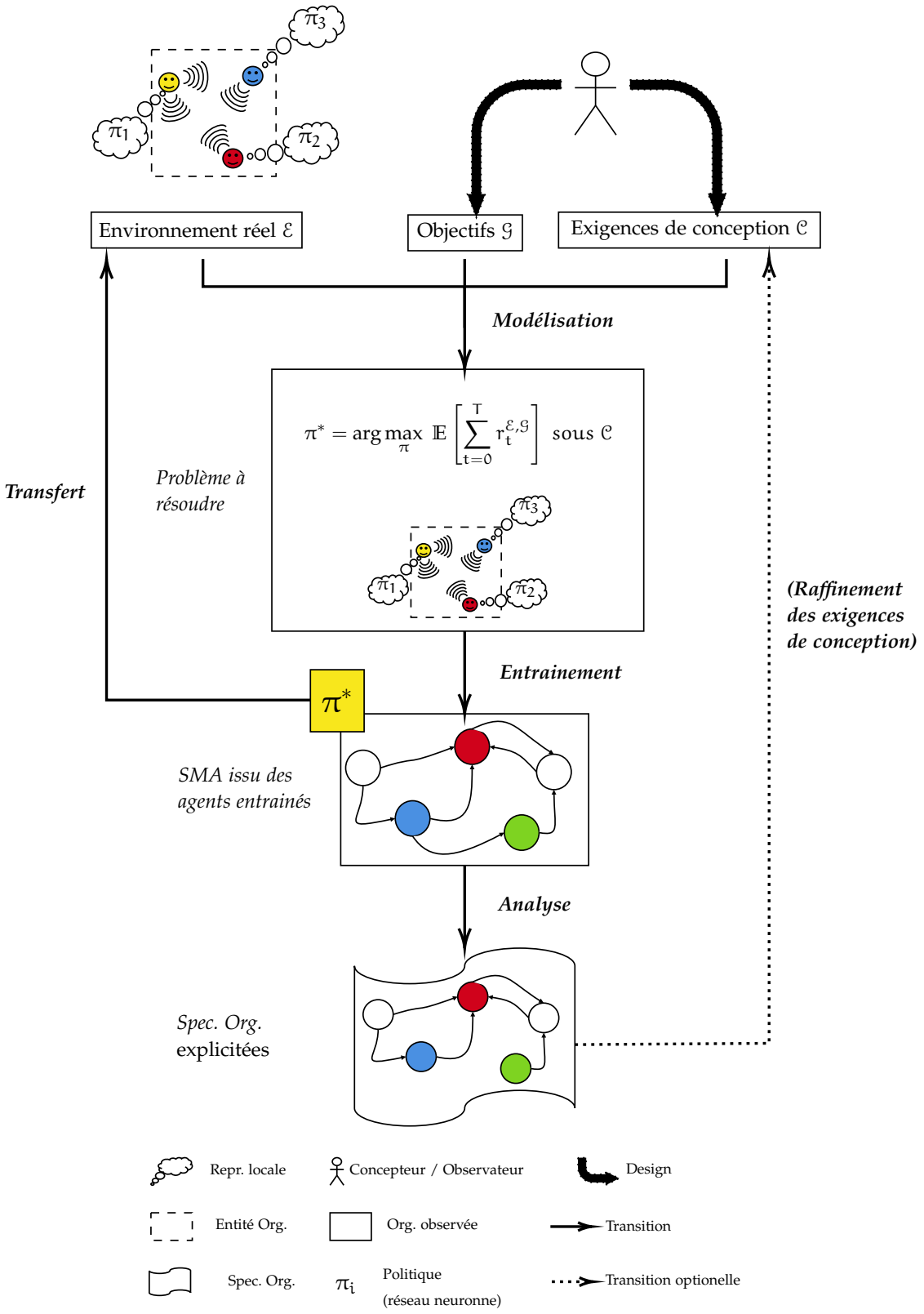


FIGURE 10 : Spécification de la question sous l'angle d'un problème d'optimisation sous contraintes

3.1.2 *Problème d'optimisation sous contraintes*

Dans cette perspective, en nous inspirant du formalisme issu du MARL, nous faisons l'hypothèse générale que la conception d'un SMA revient à rechercher un ensemble de poli-

tiques $\pi^* = \{\pi_1^*, \ldots, \pi_n^*\} \in \Pi$ (aussi appelée **politique conjointe**) qui, dans un environnement cible donné $\mathcal{E}$, permettent de maximiser un objectif global que l'on peut retranscrire par des retours quantitatifs $r_t^{\mathcal{E},\mathcal{G}}$ prenant en compte un ensemble de contraintes $\mathcal{C}$ définies a priori :

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\mathcal{E}} \left[\sum_{t=0}^{T} r_t^{\mathcal{E},\mathcal{G}} \right] \quad \text{sous} \quad \pi \in \mathcal{C}$$

Cette formulation inscrit la conception de SMAs dans un cadre d'optimisation combinatoire contrainte, où le but est non seulement d'obtenir des politiques performantes, mais aussi conformes et compréhensibles. Les contraintes $\mathcal{C}$ englobent notamment :

- **Des contraintes organisationnelles** : respect des rôles, des hiérarchies, des missions, et des modes de coordination définis dans une spécification organisationnelle ;
- **Des contraintes de sûreté ou de conformité** : règles à ne jamais violer pour éviter des comportements dangereux ou illégaux (ex. ne jamais désactiver un capteur critique, ne jamais interrompre une communication vitale) ;
- **Des contraintes d'explicabilité ou de traçabilité** : exigence que les comportements soient décomposables, interprétables ou analysables a posteriori.

Dans de nombreux contextes, entraîner des agents directement dans l'environnement réel peut s'avérer coûteux, risqué, voire irréalisable. C'est notamment le cas pour les SMAs critiques, tel que la robotique, la cybersécurité ou les systèmes embarqués. Pour contourner ces limitations, il est courant de recourir à un environnement simulé pour l'entraînement des agents. Compte tenu de la difficulté, voire du risque, d'interagir expérimentalement avec l'environnement réel cible $\mathcal{E}$, il apparaît indispensable de le modéliser sous la forme d'une simulation. Celle-ci offre un espace d'expérimentation sécurisé, dans lequel les agents peuvent tester et apprendre différentes stratégies sans conséquence sur le système réel. Au cœur de cette simulation, deux éléments sont essentiels : la capacité à attribuer une récompense via une fonction dédiée, et la capacité à générer l'observation suivante à partir de l'état courant et de l'action réalisée, grâce aux fonctions de transition et d'observation.

Cette simulation peut être obtenue de plusieurs manières :

- *par reproduction explicite*, à partir de spécifications, de règles et de topologies connues (par exemple, des modèles de réseau statiques) ;
- *par apprentissage à partir de données*, en utilisant des techniques de ML ;
- *par connexion à un environnement émulé*, via une plateforme d'émulation.

L'objectif global $\mathcal{G}$ correspond à la réussite d'un ensemble d'objectifs rassemblé (par exemple, la défense du réseau, la neutralisation d'une menace, la résilience à une attaque, etc.). Nous formalisons que les récompenses $r_t^{\mathcal{E},\mathcal{G}}$ correspondant à cet objectif $g \in \mathcal{G}$ au cours du temps sont obtenues par une fonction de récompense globale prenant un état $s_t \in S$ (ou observation conjointe $\omega_t^j \in \Omega^j$), l'action conjointe $a_t^j \in A^j$ menant à l'état suivant $s_{t+1} \in S$ (ou l'observation conjointe suivante $\omega_{t+1}^j \in \Omega^j$) :

$$r_t^{\mathcal{E},\mathcal{G}} = R_g(s_t, \omega_t^j, a_t, s_{t+1}, \omega_{t+1}^j)$$

Cette double exigence (maximiser la performance tout en respectant des contraintes hétérogènes) justifie le recours à des techniques hybrides mêlant apprentissage, organisation explicite et techniques d'analyse post-apprentissage. Elle constitue le socle sur lequel s'appuiera notre méthode complète de conception dans les chapitres suivants.

3.1.3 *Spécification organisationnelle vs émergence*

La littérature distingue deux grandes approches :

- L'**organisation spécifiée** (top-down), dans laquelle les rôles, missions et règles de coordination sont explicitement définis, souvent dans un cadre *Agent Oriented Software Engineering* (AOSE) ;
- L'**organisation émergente** (bottom-up), issue des politiques apprises par les agents, sans spécifications préalables.

Nous proposons une approche intermédiaire : guider l'émergence par des contraintes organisationnelles exprimées comme contraintes dans le processus de résolution automatique, permettant d'allier flexibilité et contrôlabilité.

3.2 DÉCOMPOSITION EN SOUS-PROBLÈMES

L'hypothèse d'une conception d'un SMA au travers d'un problème d'optimisation sous contraintes, conduit à mettre en évidence quatre sous-problèmes fondamentaux :

MOD – Modélisation réaliste de l'environnement Il s'agit d'obtenir une simulation crédible des interactions, des menaces et des comportements réseau.

TRN – Résolution sous contraintes Le cœur du problème réside dans l'obtention de politiques d'agents respectant les contraintes organisationnelles et de sûreté.

ANL – Analyse et explicabilité des comportements Il s'agit de rendre explicite le comportement des agents après apprentissage, en extrayant les spécifications organisationnelles comme des rôles ou des objectifs.

TRF – Déploiement et transfert vers le réel Ce sous-problème vise à maintenir la cohérence entre le jumeau numérique et l'environnement réel en gérant notamment le transfert des politiques apprises en simulation vers l'environnement réel tout en maintenant la fidélité de l'environnement simulé par rapport à l'environnement réel.

3.3 HYPOTHÈSES DE RECHERCHE

Chacun des sous-problèmes identifiés recouvre un vaste champ de recherches spécifiques, nécessitant une analyse approfondie de la littérature afin d'identifier les verrous à lever et les contributions à apporter pour que chaque activité puisse atteindre les objectifs fixés, et ainsi répondre à la question de recherche. Cependant, étant donné l'ampleur des états de l'art associés à chacun de ces sous-problèmes, il apparaît pertinent de circonscrire dès à présent ces espaces de recherche à travers plusieurs hypothèses. Celles-ci, bien qu'elles ne résultent pas d'une revue exhaustive de la littérature, s'appuient sur notre connaissance actuelle du domaine et permettent de guider la suite de notre démarche :

- **H-MOD :** Il est possible d'obtenir un environnement simulé réaliste, soit en facilitant la modélisation manuelle, soit par des techniques d'apprentissage machine à partir de données.
- **H-TRN :** Il est possible d'obtenir une politique conjointe via le MARL et d'y intégrer les exigences de conception en tant que spécifications organisationnelles dans le processus d'apprentissage pour améliorer la conformité et la sûreté des politiques apprises.

- **H-ANL :** Les rôles et objectifs organisationnels peuvent être inférés automatiquement à partir des trajectoires comportementales d'agents entraînés via des techniques d'apprentissage machine.
- **H-TRF :** Un couplage entre environnement réel et simulé permet un déploiement sûr, adaptatif et maintenable des politiques apprises.

L'hypothèse **H-MOD** s'appuie sur le constat que, dans les approches de conception AOSE et au-delà, il est généralement indispensable de disposer d'un modèle simulé de l'environnement pour réaliser des analyses et des évaluations avant tout déploiement réel, en particulier dans des contextes critiques. Or, la modélisation manuelle de l'environnement est une tâche complexe, longue et sujette à erreurs. Une première piste consiste à structurer ce travail à l'aide d'un cadre de conception dédié, qui organise et optimise la modélisation tout en limitant les interventions manuelles et les risques d'erreur. Par ailleurs, les avancées récentes en apprentissage automatique, notamment en apprentissage par renforcement [108], ouvrent la voie à une automatisation accrue de cette modélisation : il devient possible d'apprendre les dynamiques de l'environnement à partir de données historiques, et ainsi de reproduire la perception dynamique des agents sans recourir à un modèle interne explicite de l'état caché, comme c'est traditionnellement le cas dans une modélisation entièrement manuelle.

L'hypothèse **H-TRN** repose sur une analyse de la littérature montrant que, parmi les différentes approches de conception de SMAs en Cyberdéfense (comme le *Distributed Constraint Optimization Problem* – DCOP ou les algorithmes génétiques), le MARL s'est imposé comme méthode de référence, notamment dans des environnements complexes et dynamiques, grâce à sa capacité à produire des politiques conjointes efficaces là où les approches classiques, souvent exactes, mais coûteuses, peinent à s'adapter à la réalité opérationnelle [77]. Cette tendance est renforcée par l'apparition de nouvelles techniques facilitant l'apprentissage automatisé des interactions entre agents [107]. Toutefois, la littérature souligne aussi les limites du MARL concernant le contrôle du processus d'apprentissage et la maîtrise des politiques obtenues, des enjeux cruciaux en Cyberdéfense où la sûreté et la conformité sont essentielles. Plusieurs travaux relèvent ainsi le manque de garanties sur la conformité des comportements émergents et la difficulté à imposer des contraintes organisationnelles explicites lors de l'apprentissage multi-agent. Dès lors, il apparaît pertinent d'envisager une approche hybride, combinant la puissance adaptative du MARL et la rigueur d'un modèle d'organisation explicite, afin d'intégrer directement des spécifications organisationnelles dans le processus d'apprentissage pour guider l'émergence des comportements, garantir leur conformité aux exigences de conception, et préserver la flexibilité propre à l'apprentissage automatique. Cette orientation, encore peu explorée, répond à un besoin identifié de concilier performance, contrôle et sûreté dans la conception de SMAs pour la Cyberdéfense.

L'hypothèse **H-ANL** s'inscrit dans la continuité des travaux en ingénierie organisationnelle orientée agents (AOSE) où la validation de la conformité des comportements par rapport aux spécifications organisationnelles est reconnue comme un enjeu central pour la sûreté et la fiabilité des systèmes [157, 163]. Cependant, la littérature souligne la difficulté à automatiser cette validation, du fait de la complexité des dynamiques collectives et de la diversité des structures organisationnelles possibles. Nous postulons alors que les méthodes d'apprentissage automatique, en particulier l'apprentissage non supervisé, constituent des leviers prometteurs pour extraire automatiquement des spécifications organisationnelles à partir des historiques comportementaux des agents entraînés. Cette

hypothèse s'appuie sur des avancées récentes en analyse de séries temporelles et de trajectoires, qui ont démontré leur capacité à révéler des motifs récurrents et des structures cachées dans des données séquentielles complexes [77, 94]. En mobilisant des techniques telles que le clustering, la classification non supervisée ou l'analyse de séquences, il devient envisageable d'identifier, sans supervision humaine, des rôles, des objectifs émergents ou des schémas d'interaction caractéristiques au sein du SMA. Une telle démarche permettrait à la fois d'automatiser l'analyse de conformité organisationnelle, en réduisant la dépendance à l'expertise humaine, et de renforcer l'explicabilité des SMAs, en offrant une interprétation claire et objectivée des comportements collectifs observés. Ce levier est d'autant plus crucial dans des domaines critiques comme la Cyberdéfense, où la confiance dans les systèmes automatisés repose sur la capacité à comprendre, justifier et auditer les décisions prises par les agents [94].

L'hypothèse **H-TRF** s'appuie sur la nécessité d'un couplage étroit entre environnement simulé et environnement réel, mais va plus loin en postulant qu'un système inspiré des jumeaux digitaux, capable de synchroniser et d'enrichir dynamiquement le modèle simulé à partir d'informations collectées sur l'environnement réel, permettrait d'itérer efficacement le processus d'entraînement et d'analyse pour produire des politiques conjointes robustes et adaptées. Concrètement, nous faisons l'hypothèse qu'un framework dédié peut orchestrer la récupération continue de données réelles, améliorer itérativement la fidélité du modèle simulé, et enclencher un cycle d'apprentissage multi-agent dont les résultats (politiques conjointes) sont ensuite déployés dans l'environnement réel. Ce déploiement peut s'effectuer soit en installant la politique dans chaque agent (si les ressources embarquées le permettent), soit en exécutant la politique conjointe sur un nœud externe qui transmet les actions à appliquer aux agents, évitant ainsi la nécessité d'embarquer des politiques coûteuses et facilitant la gestion à distance. Cette approche vise à garantir une adaptation continue, une robustesse accrue et une cohérence opérationnelle entre simulation et réalité, répondant ainsi aux défis identifiés dans la littérature tout en ouvrant la voie à une Cyberdéfense multi-agent réellement dynamique et soutenable.

3.4 VERS UNE MÉTHODE DE CONCEPTION ORGANISATIONNELLE

La structuration du problème de conception d'un SMA de Cyberdéfense autour des quatre activités fondamentales (**modélisation**, **apprentissage**, **analyse** et **transfert**) ouvre la voie à une démarche intégrée, articulant ces activités de façon cyclique et complémentaire pour répondre simultanément aux cinq critères majeurs identifiés précédemment : autonomie (C1), performance (C2), adaptation (C3), contrôle (C4) et explicabilité (C5). Chacune de ces activités cible des enjeux spécifiques : la modélisation vise à garantir un environnement d'expérimentation crédible et adaptable (C1, C3), l'apprentissage permet d'optimiser la performance tout en intégrant des contraintes organisationnelles (C2, C4), l'analyse favorise l'explicabilité et la vérification des comportements (C5, C4), tandis que le transfert assure la cohérence et l'adaptation continue entre simulation et réalité (C1, C3). Cette vision intégrée laisse ainsi entrevoir la possibilité d'une méthode générale, capable de couvrir l'ensemble des critères de conception et de répondre à la question de recherche, sans toutefois la détailler à ce stade. La démarche d'élaboration de cette méthode peut être vue comme un processus structuré décrivant comment pour chaque critère, des activités sont associées ainsi que leurs objectifs qui elles-mêmes liées aux sous-problèmes à résoudre grâce aux hypothèses qui délimitent l'espace de recherche d'où des verrous sont

identifiés dans la littérature, et enfin permettre d'établir des contributions pour lever ces verrous.

TABLE 7 : Synthèse des relations entre critères, activités, sous-problèmes et hypothèses ainsi que futurs verrous et contributions

Critères	**Activités (et objectifs)**	**Sous-problèmes**	**Hypothèses**	**Verrous et contributions**
C1 Autonomie, C3 Adaptation (Obtenir un environnement crédible et adaptable)	Modélisation **MOD** : Modélisation réaliste de l'environnement	**H-MOD** : Possibilité d'obtenir un environnement simulé réaliste (manuellement ou par apprentissage)	Difficulté de modélisation manuelle, manque d'automatisation, fidélité limitée des simulations	...
C2 Performance, C4 Contrôle (Optimiser la performance sous contraintes organisationnelles)	Apprentissage **TRN** : Résolution sous contraintes	**H-TRN** : Intégration des exigences organisationnelles dans l'apprentissage multi-agent	Difficulté à imposer des contraintes explicites dans le MARL, manque de garanties de conformité	...
C5 Explicabilité, C4 Contrôle (Expliciter et vérifier les comportements organisationnels)	Analyse **ANL** : Analyse et explicabilité des comportements	**H-ANL** : Extraction automatique de structures organisationnelles à partir des trajectoires d'agents	Difficulté d'automatisation de l'analyse, manque d'outils pour relier comportements et organisation	...
C1 Autonomie, C3 Adaptation (Assurer la cohérence et l'adaptation entre simulation et réalité)	Transfert **TRF** : Déploiement et transfert vers le réel	**H-TRF** : Couplage inspiré des jumeaux digitaux entre simulation et environnement réel	Difficulté de transfert des politiques, écart simulation/réalité, manque de synchronisation dynamique	...

Cette chaîne logique structure la progression de la thèse : chaque activité vise à satisfaire un ou plusieurs critères, se décline en sous-problèmes, dont la résolution est guidée par des hypothèses de recherche, confrontées aux verrous identifiés dans la littérature, et aboutit à des contributions permettant de couvrir ces mêmes critères.

3.5 BILAN

Ce chapitre propose une formalisation de la conception d'un SMA de Cyberdéfense comme un problème d'optimisation sous contraintes, permettant d'articuler les exigences de performance, d'adaptation, de contrôle et d'explicabilité. Il clarifie les critères de conception, décompose le problème en quatre sous-problèmes fondamentaux (**MOD**, **TRN**, **ANL**, **TRF**), et formule des hypothèses de recherche qui structurent la démarche scientifique. Ce chapitre pose les bases méthodologiques pour aborder la conception des SMA de Cyberdéfense de manière rigoureuse et reproductible, tout en préparant la transition vers l'analyse des verrous et des contributions dans les parties suivantes.

CONCLUSION

La Partie I a posé les fondations conceptuelles et scientifiques de notre travail, centré sur la conception d'un SMA pour la Cyberdéfense. Face à des menaces toujours plus rapides, distribuées et adaptatives, les architectures comme AICA, proposées par l'OTAN, constituent une réponse prometteuse. Elles définissent un agent capable de percevoir, décider, agir et coopérer, qui peut être décliné en une organisation distribuée d'agents Cyberdéfenseurs.

Nous avons formulé la question centrale de la thèse : comment concevoir automatiquement un SMA de Cyberdéfense auto-organisé, capable de s'adapter à un environnement dynamique tout en respectant un ensemble d'exigences critiques. Cette réflexion a conduit à l'identification de cinq critères de conception (C1 à C5), qui servent de fil conducteur pour évaluer les approches existantes et structurer notre méthode.

L'état de l'art met en évidence une opposition entre approches connexionnistes (où l'organisation émerge de l'apprentissage) et approches symboliques (fondées sur des mécanismes organisationnels explicites). Aucune des solutions étudiées ne satisfait pleinement l'ensemble des critères. Les approches connexionnistes offrent adaptation et performance, mais au prix d'une moindre maîtrise et d'une explicabilité réduite ; à l'inverse, les approches symboliques privilégient le contrôle et la transparence, mais se heurtent à des limites en termes d'automatisation et d'adaptabilité.

Cette tension fondamentale nous conduit à reconsidérer la conception du SMA non comme un simple problème d'ingénierie, mais comme un problème d'optimisation sous contraintes. Il s'agit alors de faire émerger une organisation adaptée à partir de contraintes organisationnelles formalisées. Cette reformulation permet de décomposer la complexité du problème en quatre activités principales :

- **Modélisation** : obtenir un environnement simulé crédible et adaptable pour l'expérimentation et l'entraînement des agents ;
- **Apprentissage** : optimiser les politiques conjointes des agents sous contraintes organisationnelles et de sûreté ;
- **Analyse** : expliciter, vérifier et interpréter les comportements collectifs et leur adéquation aux spécifications organisationnelles ;
- **Transfert** : assurer la cohérence et l'adaptation continue entre simulation et environnement réel lors du déploiement.

Cette décomposition conduit à identifier quatre sous-problèmes fondamentaux, correspondant aux objectifs de chaque activité : (**MOD**) la modélisation réaliste de l'environnement (**TRN**) la résolution sous contraintes (**ANL**) l'analyse et l'explicabilité des comportements, et (**TRF**) le déploiement et le transfert vers le réel. Pour chacun de ces sous-problèmes, nous avons formulé des hypothèses de recherche (**H-MOD**, **H-TRN**, **H-ANL**, **H-TRF**) qui délimitent l'espace d'investigation et orientent la suite du manuscrit.

Afin que chaque activité atteigne ses objectifs, il est nécessaire de résoudre les sous-problèmes définis dans le cadre de ces hypothèses. Cela implique d'identifier, à partir d'une analyse de la littérature, les verrous qui freinent la résolution de ces sous-problèmes, puis de proposer des contributions permettant de les lever. Ce travail de revue de littérature et d'identification des verrous sera présenté dans la Partie II.

Deuxième partie

ÉTAT DE L'ART

INTRODUCTION

Cette partie vise à identifier et analyser les approches les plus pertinentes pour résoudre les quatre sous-problèmes issus de la décomposition du problème de conception d'un SMA de Cyberdéfense, formalisé comme un problème d'optimisation sous contraintes. L'objectif est de recenser les travaux permettant de répondre aux critères spécifiques de chaque sous-problème, tout en mettant en lumière les verrous scientifiques et techniques qui subsistent, afin de capitaliser au mieux sur les avancées existantes.

Le premier chapitre propose une revue critique de la littérature pour chacun des sous-problèmes, en s'appuyant sur les quatre grandes activités de la démarche. Il s'agit d'identifier, pour chaque activité, les contributions majeures susceptibles de satisfaire les critères fixés, ainsi que les principaux verrous qui justifient la nécessité de nouvelles avancées méthodologiques.

Le second chapitre approfondit l'étude des travaux identifiés comme les plus prometteurs, afin de fournir un socle théorique solide pour expliquer et circonscrire techniquement les verrous identifiés. Ce chapitre introduit ainsi les concepts fondamentaux qui serviront de base aux futures contributions méthodologiques de la méthode proposée.

La Figure 11 synthétise l'organisation de cette partie et les liens logiques entre chapitres et sous-sections.

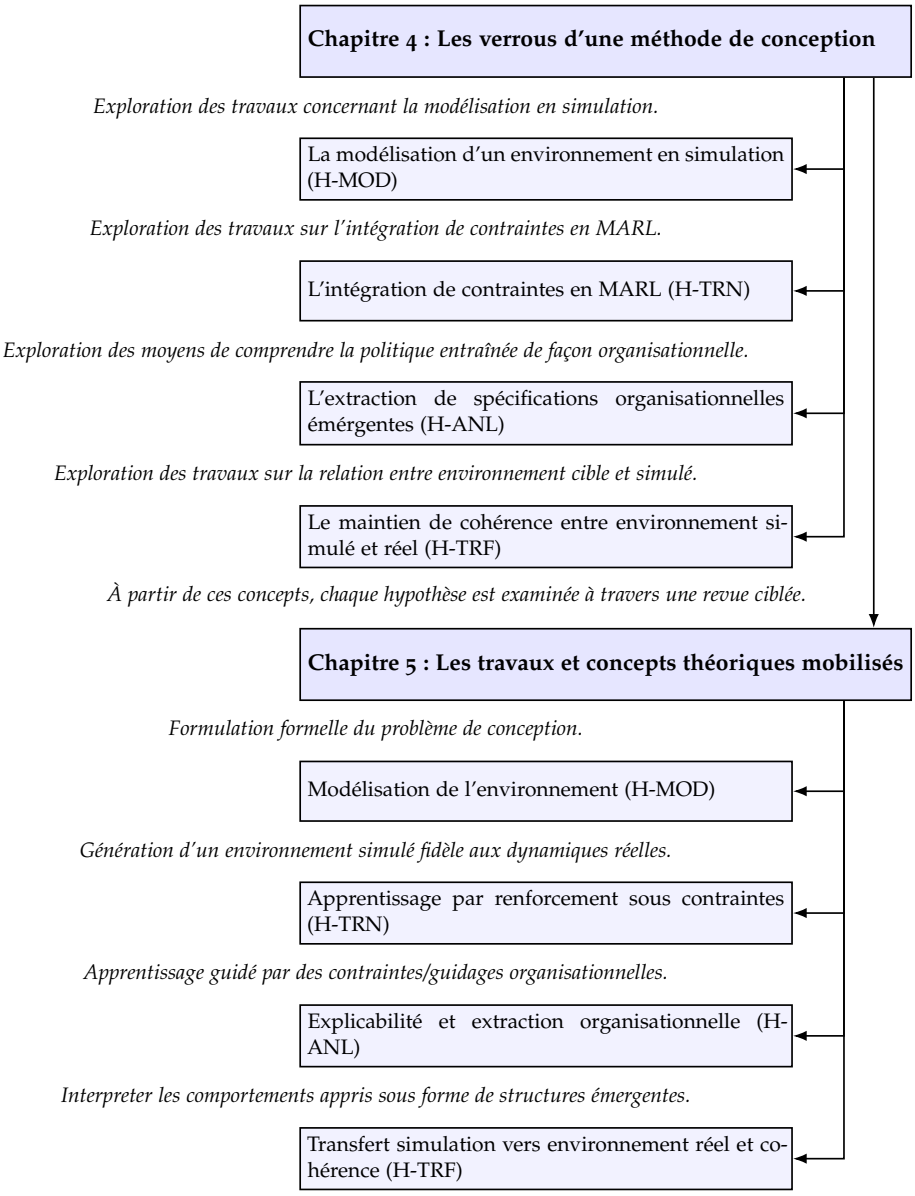


FIGURE 11 : Structure de la Partie II : État de l'art

4

LES VERROUS D'UNE MÉTHODE DE CONCEPTION

Ce chapitre a pour objectif d'introduire et d'analyser les principaux verrous scientifiques liés à la conception automatisée de SMAs, tels qu'ils émergent de la décomposition du problème en quatre grandes activités : modélisation, résolution, analyse et transfert. Pour chacune de ces activités, un sous-problème fondamental (**MOD** à **TRF**) a été identifié, associé à une hypothèse de restriction de l'espace de recherche (**H-MOD** à **H-TRF**). Nous définissons pour chaque sous-problème un ensemble de critères spécifiques permettant d'évaluer la couverture des approches existantes. La revue de littérature conduite selon ces critères permet de cartographier les travaux les plus pertinents, d'identifier les avancées majeures, ainsi que les verrous et lacunes qui subsistent. Ce chapitre propose ainsi une analyse critique de l'état de l'art pour chaque hypothèse, en vue de dégager les défis à relever et de motiver les contributions méthodologiques présentées dans la suite du manuscrit.

Repères de lecture. Afin de rendre la lecture plus fluide malgré la densité des acronymes et tableaux croisés, chaque section suit la même structure : (i) recontextualisation du sous-problème, (ii) hypothèse de restriction, (iii) couverture des critères, puis (iv) analyse des verrous. Les sous-problèmes **MOD**, **TRN**, **ANL** et **TRF** sont systématiquement reliés aux critères **C1** à **C5** pour garder un fil directeur unique.

4.1 LA MODÉLISATION D'UN ENVIRONNEMENT EN SIMULATION (H-MOD)

Recontextualisation du sous-problème dans la démarche de la thèse

Le premier sous-problème fondamental identifié dans notre approche concerne la **modélisation réaliste de l'environnement** (**MOD**). Cette étape est cruciale, car elle conditionne la crédibilité des expérimentations, l'entraînement des agents et, in fine, la validité des politiques de Cyberdéfense obtenues. Dans le cadre de la thèse, la simulation d'un environnement pertinent permet de tester, d'évaluer et d'optimiser les comportements multi-agents sans exposer de systèmes réels à des risques opérationnels. Elle constitue ainsi le socle expérimental sur lequel reposent les phases ultérieures d'apprentissage, d'analyse et de transfert.

Objectif global et critères spécifiques

L'objectif principal de cette activité est d'obtenir un environnement simulé qui soit à la fois **réaliste**, **adaptable**, **fidèle aux dynamiques du monde réel** et **exploitable pour l'entraînement et l'évaluation des agents**. Pour guider la revue de littérature, nous retenons les critères spécifiques suivants :

- **Fidélité** : capacité à reproduire les dynamiques, menaces et interactions observées dans des environnements réels ; *ce critère est essentiel pour garantir que les politiques apprises en simulation soient pertinentes et transférables au monde réel.*

- **Adaptabilité** : possibilité de modifier facilement la topologie, les scénarios ou les paramètres pour explorer différents contextes ; *cela permet de tester la robustesse des agents face à des environnements variés et d'étudier des cas d'usage multiples*.
- **Automatisation** : degré d'automatisation de la génération ou de l'évolution de l'environnement simulé ; *une automatisation élevée facilite la création rapide de nouveaux scénarios et réduit la dépendance à l'expertise humaine*.
- **Interopérabilité** : aptitude à intégrer ou à échanger des données avec d'autres outils ou plateformes ; *ce critère favorise la réutilisation, la comparaison et l'intégration de résultats issus de différents systèmes ou sources de données*.
- **Facilité d'utilisation** : accessibilité pour les concepteurs, possibilité de personnalisation sans expertise avancée ; *une utilisation aisée accélère le prototypage et démocratise l'expérimentation auprès d'utilisateurs non spécialistes*.
- **Multi-agent** : capacité de l'environnement à accueillir un ou plusieurs agents, et à gérer leurs interactions ; *ce critère est indispensable pour étudier des scénarios collaboratifs ou compétitifs, et pour évaluer des approches multi-agents*.

*Hypothèse de restriction de l'espace de recherche (**H-MOD**)*

Afin de circonscrire l'espace de recherche à un domaine pertinent et exploitable, nous posons l'hypothèse suivante : *Il est possible d'obtenir un environnement simulé réaliste, soit en facilitant la modélisation manuelle, soit par des techniques d'apprentissage machine à partir de données*. Cette hypothèse nous conduit à concentrer la revue de littérature sur deux grandes familles de travaux :

- **La modélisation manuelle** : regroupe tous les travaux où la création de l'environnement simulé repose sur une intervention humaine. Cela inclut les simulateurs (CybORG, NASimEmu, *Cyber Security Simulation Testbed* – CYST) et modèles d'environnements de Cyberdéfense génériques, dans lesquels la structure, les règles et les scénarios sont explicitement définis, totalement ou partiellement, par des experts. Cette approche permet de réutiliser des modèles existants, soit directement si l'environnement simulé correspond à l'environnement réel cible, soit après adaptation et instanciation du modèle de simulation générique. Bien que largement répandue, cette méthode limite souvent la généricité, car peu de modèles sont à la fois faciles à adapter et applicables à une grande diversité d'environnements de Cyberdéfense. On y retrouve également les formalismes Markoviens (*Markov Decision Process* – MDP, *Partially Observable Markov Decision Process* – POMDP, *Decentralized Partially Observable Markov Decision Process* – Dec-POMDP, *Partially Observable Stochastic Games* – POSG), qui servent de base théorique et pratique à la plupart des modélisations manuelles.
- **La modélisation basée sur l'apprentissage** : regroupe les approches où la dynamique de l'environnement est apprise automatiquement à partir de données collectées dans l'environnement réel ou issues d'interactions. Cela inclut les travaux d'identification de systèmes (*System Identification*), de modélisation de substitution (*Surrogate Modeling*) ou de simulation guidée par les données (*Data-Driven Simulation*).

Ce choix est motivé par l'état de l'art, qui montre que ces deux axes concentrent l'essentiel des avancées récentes et offrent des compromis différents entre fidélité, automatisation et adaptabilité.

Couverture des critères par les travaux identifiés

Une synthèse de la couverture des critères par les travaux identifiés dans les sujets évoqués dans l'hypothèse (**H-MOD**) est présentée dans le Table 8.

TABLE 8 : Couverture des critères spécifiques par les principales familles de travaux de modélisation d'environnements de Cyberdéfense

Travaux / Critères	Fidélité	Adaptabilité	Automatisation	Interopérabilité	Facilité d'utilisation	Multi-agent
Modèles formels génériques (MDP, POMDP, Dec-POMDP, POSG, graphes d'attaque, arbres AD, réseaux de Petri, modèles de jeu)	✓	✓	✗	✓	✗	✓
Frameworks génériques et configurables (CyberBattleSim, NASim, NASimEmu, DETERLab, CyberVAN, CYST)	✓	✓	✗	✓	✓	✓
Simulateurs spécialisés de Cyberdéfense (CybORG, CybORG++, CyberWheel, SCYTHE, CTF)	✓	✗	✗	✓	✓	✓
System Identification	✓	✓	✓	✗	✗	✓
Surrogate Modeling	✓	✓	✓	✗	✓	✓
Data-Driven Simulation (World Models, simulation guidée par les données)	✓	✓	✓	✗	✗	✓

Concernant l'approche de modélisation manuelle, les travaux identifiés sont :

LES MODÈLES FORMELS GÉNÉRIQUES. Au niveau le plus abstrait, la modélisation des environnements de Cyberdéfense repose sur des formalismes mathématiques issus de la théorie de la décision séquentielle. Le cadre de base est le MDP [190], qui décrit un environnement comme un ensemble d'états, d'actions et de transitions probabilistes. Lorsque l'information disponible est partielle, les POMDP [185] offrent un cadre adapté. Pour les SMAs coopératifs, les variantes décentralisées (Dec-POMDP) [132] sont utilisées, tandis que les environnements compétitifs (attaquant/défenseur) sont modélisés par les POSG [168]. D'autres extensions, comme les MDP factorisés [171], facilitent la modélisation de systèmes complexes, et la théorie des jeux de sécurité [143] permet de formaliser explicitement les stratégies adversariales. Ces formalismes constituent la base théorique commune à la plupart des simulateurs et frameworks de cybersécurité. À côté des modèles markoviens, des représentations graphiques intermédiaires sont également utilisées. Les *graphes d'attaque* [186] décrivent les différentes voies d'exploitation des vulnérabilités d'un réseau, tandis que les *arbres attaque-défense* [151] intègrent explicitement les contre-mesures défensives. Les *réseaux de Petri* [62, 78, 90] permettent de représenter et d'analyser des stratégies concurrentes d'attaque et de défense. Enfin, les *modèles de jeu* appliqués à la cybersécurité [102, 112, 158] formalisent les interactions attaquant/défenseur comme des jeux dynamiques, souvent non coopératifs, et constituent une passerelle naturelle vers les cadres POSG et Dec-POMDP [87, 142, 150]. En résumé, ces modèles formels et graphiques offrent une boîte à outils générique et flexible pour construire des environnements de simulation adaptés à une grande variété de scénarios de cybersécurité.

LES FRAMEWORKS GÉNÉRIQUES ET CONFIGURABLES. Entre les formalismes abstraits et les simulateurs spécialisés, la littérature met en avant plusieurs frameworks visant à

offrir un compromis entre généricité et exploitabilité. Parmi ceux-ci, CyberBattleSim [75], développé par Microsoft, propose une modélisation configurable d'un réseau sous forme de graphe, où chaque nœud représente un service vulnérable. De manière similaire, *NASim* [46] et son extension NASimEmu [21] permettent de définir des topologies, vulnérabilités et scénarios d'attaque variés, avec une compatibilité avec OpenAI Gym facilitant l'usage en apprentissage par renforcement. D'autres approches se situent dans des infrastructures de test à grande échelle comme *DETERLab* et *CyberVAN* [152], qui offrent des environnements configurables pour la simulation et l'émulation cyber. De plus, l'outil *CYST* [82] propose une plateforme modulaire pour la création et l'évaluation de scénarios de Cyberdéfense, avec un accent sur la flexibilité et l'extensibilité. Ces frameworks se distinguent par leur capacité à être adaptés à différents contextes, tout en fournissant des environnements suffisamment réalistes pour l'entraînement et l'évaluation des agents de Cyberdéfense.

LES SIMULATEURS SPÉCIALISÉS DE CYBERDÉFENSE. Enfin, au niveau le plus concret, plusieurs simulateurs dédiés ont été développés pour fournir des environnements instanciés dans lesquels les agents de Cyberdéfense peuvent être entraînés et évalués. Parmi eux, CybORG [74] est devenu une référence, avec des scénarios Red Team/Blue Team exploitables dans des compétitions comme le *Cyber Automated Game-based Evaluation* (CAGE) Challenge, et qui a été étendu dans CybORG++ pour intégrer des modèles Dec-POMDP et des scénarios multi-agents complexes [6]. Le simulateur *CyberWheel* [17] a également été proposé pour l'entraînement académique de défenseurs automatisés, avec un accent sur la formation. Outre ces outils de recherche, des plateformes orientées *capture-the-flag* comme *SCYTHE* ou des environnements de type CTF [43] ont été détournés pour servir de bancs d'essai aux approches d'apprentissage par renforcement en Cyberdéfense. Ces simulateurs spécialisés se distinguent par leur haut degré de fidélité et leur orientation vers des cas d'usage précis, mais au prix d'une adaptabilité plus limitée par rapport aux frameworks génériques.

Concernant l'approche de modélisation automatique, les travaux identifiés sont :

SYSTEM IDENTIFICATION. Une première famille de travaux automatise la construction des environnements simulés via des méthodes d'identification de systèmes, où la dynamique du monde réel est reconstruite à partir de données empiriques. Ces approches cherchent à extraire automatiquement les équations ou modèles décrivant l'évolution d'un système, par exemple dans des contextes de micro-réseaux ou de SMAs soumis à des attaques [2, 38]. L'identification peut se faire à travers l'estimation paramétrique ou structurelle de modèles de contrôle [51], mais aussi par l'ajustement de modèles probabilistes/stochastiques qui intègrent directement l'incertitude des dynamiques et sont calibrés sur des données réelles. Ces modèles probabilistes incluent notamment les processus de Markov bayésiens, les réseaux bayésiens dynamiques et les processus gaussiens, qui permettent de capturer des comportements non linéaires et incertains dans des environnements complexes. De tels travaux illustrent comment l'identification de systèmes fournit une base automatisée et statistiquement robuste pour simuler des environnements de Cyberdéfense à partir de traces et de mesures.

SURROGATE MODELING. Une seconde famille regroupe les travaux de *modélisation de substitution*, où un modèle approximatif et léger est entraîné pour reproduire le comportement d'un simulateur ou d'un système coûteux. Ces modèles sont particulièrement utiles

dans les environnements cyber-physiques où l'exécution d'un simulateur de haute fidélité est trop lourde pour permettre l'entraînement massif d'agents. On retrouve ici les approches par réseaux neuronaux ou graph neural networks servant de modèles substituts [16, 20], mais également des méthodes probabilistes permettant d'estimer la distribution des sorties du système [27]. Ces surrogate models peuvent être distribués et intégrés dans des architectures fédérées pour préserver la confidentialité des données tout en améliorant la vitesse et la performance des simulations [16]. Ils constituent un compromis efficace entre réalisme et exploitabilité, en offrant aux agents des environnements proches de la réalité, mais plus accessibles computationnellement.

DATA-DRIVEN SIMULATION. Enfin, la troisième famille repose sur des approches de simulation directement guidées par les données collectées dans les environnements réels. Ces approches construisent un jumeau numérique partiel ou complet du système cible en utilisant les traces, journaux et trajectoires observées. Un exemple emblématique est celui des *World Models* [109], qui utilisent des architectures neuronales pour apprendre un espace latent compact permettant de simuler et de généraliser les dynamiques d'un environnement à partir d'interactions passées. Dans le contexte de la cybersécurité, ces modèles peuvent simuler des comportements d'attaquants et de défenseurs en exploitant directement les logs réseau ou les données issues d'incidents passés [49]. Plus largement, des approches de simulation data-driven combinent apprentissage profond, représentations symboliques et renforcement multi-agent [40, 63], afin de créer des environnements artificiels, mais réalistes, capables de capturer la co-évolution attaque/défense [15]. Ces travaux ouvrent la voie à des environnements simulés dont la fidélité et l'adaptabilité croissent avec la richesse des données collectées.

Analyse des travaux et verrous

L'analyse des travaux de modélisation manuelle montre que les modèles formels génériques, en particulier le cadre Dec-POMDP, offrent la meilleure adaptabilité et généricité pour représenter des environnements de Cyberdéfense multi-agents. Contrairement aux frameworks configurables ou aux simulateurs spécialisés, qui sont souvent limités par leur structure interne et leur difficulté d'adaptation à de nouveaux contextes, un modèle Dec-POMDP permet de formaliser n'importe quel scénario en s'appuyant sur des abstractions standardisées (états, actions, observations, récompenses). Cependant, cette flexibilité a un coût : la modélisation manuelle d'un environnement réaliste reste une tâche lourde, nécessitant une expertise poussée et un effort important de formalisation. De plus, l'automatisation de cette étape est très limitée, car il n'existe pas de bibliothèque de modèles Dec-POMDP pré-spécialisés pour la cybersécurité qui factoriserait les invariants communs à ces environnements. Ainsi, même si le Dec-POMDP est le choix le plus pertinent pour garantir l'adaptabilité, il ne couvre pas le critère d'automatisation et impose une barrière d'entrée élevée pour la modélisation de nouveaux environnements.

Du côté de la modélisation automatique, les approches d'identification de systèmes et de surrogate modeling apportent des solutions intéressantes pour automatiser la génération de modèles à partir de données, mais elles requièrent souvent une pré-modélisation mathématique ou une calibration fine sur des jeux de données spécifiques. Les *World Models* [108] se distinguent comme la solution la plus prometteuse pour automatiser la modélisation, car ils apprennent directement la dynamique observationnelle de l'environnement sans nécessiter de structure explicite préalable. Cette approche offre une fidélité élevée et une grande agilité, indépendamment du domaine d'application. Toutefois, un

verrou subsiste : les World Models actuels sont principalement conçus pour des contextes mono-agent et leur extension au multi-agent reste un défi ouvert, notamment pour capturer les interactions complexes entre agents. Par ailleurs, l'automatisation complète est freinée par la nécessité de régler des hyperparamètres et d'adapter l'architecture du modèle à chaque nouveau cas d'usage. En résumé, si les World Models représentent l'option la plus automatisée et fidèle, ils ne couvrent pas encore pleinement les besoins d'un environnement multi-agent de Cyberdéfense sans intervention humaine.

4.2 L'INTÉGRATION DE CONTRAINTES EN MARL (H-TRN)

Recontextualisation du sous-problème dans la démarche de la thèse

Le second sous-problème fondamental de notre démarche concerne la **capacité à intégrer des contraintes ou des guidages organisationnels explicites dans le processus d'apprentissage multi-agent** (MARL). Alors que le MARL permet aux agents de découvrir de manière autonome des politiques coopératives dans des environnements complexes, il ne garantit pas, en l'état, le respect de spécifications organisationnelles essentielles telles que la répartition des rôles, la coordination structurée ou la conformité à des règles de sûreté.

Dans le contexte de la Cyberdéfense, cette question revêt une importance particulière. Les environnements sont non seulement dynamiques et partiellement observables, mais ils imposent également des exigences fortes en matière de sécurité, de robustesse et d'explicabilité. Il ne suffit pas que les agents apprennent à coopérer efficacement : il est souvent indispensable qu'ils respectent des contraintes organisationnelles définies a priori, par exemple pour garantir la séparation des responsabilités, la hiérarchie des décisions, ou la conformité à des protocoles de défense.

Ce sous-problème s'inscrit donc au cœur de la démarche de la thèse, qui vise à concilier l'autonomie d'apprentissage offerte par le MARL avec la nécessité de garantir des propriétés organisationnelles critiques. L'intégration de telles contraintes ou guidages organisationnels dans le processus d'apprentissage doit permettre d'assurer la cohérence, la sûreté et l'explicabilité des comportements collectifs, tout en préservant la capacité d'adaptation des agents face à des menaces évolutives. Cette problématique structure ainsi l'un des axes majeurs de la méthode proposée, en cherchant à dépasser les limites des approches purement émergentes ou purement prescriptives dans la conception des SMAs pour la Cyberdéfense.

Pour évaluer la capacité des approches existantes à intégrer des contraintes ou guidages organisationnels dans le processus d'apprentissage multi-agent, nous retenons les critères spécifiques suivants :

- **Expressivité des contraintes** : aptitude à exprimer des exigences organisationnelles complexes (rôles, missions, interactions, hiérarchies) ; *ce critère est essentiel pour garantir que les contraintes organisationnelles pertinentes du domaine puissent être formalisées et prises en compte dans l'apprentissage.*

- **Niveau d'intégration** : niveau auquel les contraintes sont prises en compte (action, politique, trajectoire, organisation) ; *ce critère permet d'évaluer la granularité et la portée des contraintes, et leur impact sur le comportement collectif des agents.*

- **Garantie de respect** : existence de garanties théoriques ou empiriques sur la satisfaction des contraintes ; *ce critère est indispensable pour assurer que les propriétés organisationnelles critiques ne soient pas violées lors de l'exécution des politiques apprises.*

- **Compatibilité avec l'apprentissage** : maintien de la capacité d'adaptation et d'optimisation des agents ; *ce critère vise à s'assurer que l'intégration des contraintes n'entrave pas la flexibilité et l'efficacité de l'apprentissage multi-agent.*

- **Explicabilité** : possibilité d'expliquer ou de vérifier le respect des contraintes dans les politiques apprises ; *ce critère est fondamental pour permettre l'analyse, la validation et la certification des comportements collectifs vis-à-vis des exigences organisationnelles.*

*Hypothèse de restriction de l'espace de recherche (**H-TRN**)*

Afin de circonscrire l'espace de recherche à un domaine pertinent et exploitable, nous reprenons l'hypothèse suivante :

Il est possible d'intégrer des contraintes ou des guidages organisationnels explicites dans le processus d'apprentissage multi-agent, de façon à orienter ou restreindre l'espace des politiques apprises, tout en préservant la capacité d'adaptation et d'optimisation des agents.

Cette hypothèse conduit à explorer principalement trois axes complémentaires dans la littérature :

- **L'intégration de contraintes explicites** : travaux où des contraintes de sûreté, de structure ou de mission sont formellement ajoutées au processus d'apprentissage (par exemple via des Constrained MDP, *Constrained Policy Optimization* – CPO, Deep Constrained Q-Learning, etc.) ;

- **L'utilisation de mécanismes de guidage** : approches qui orientent l'apprentissage par des techniques telles que le reward shaping, le shielding, ou l'incorporation de feedback humain, afin d'influencer indirectement les politiques apprises ;

- **L'incorporation de modèles organisationnels symboliques** : tentatives d'intégrer des spécifications organisationnelles issues de l'ingénierie des SMA (principalement AGR [167] et $\mathcal{M}$OISE$^+$ [173]) dans le processus d'apprentissage, afin de garantir le respect de structures, de rôles ou de missions collectives.

Le périmètre de la revue de littérature est ainsi délimité par ces trois axes, qui couvrent l'ensemble des approches visant à combiner apprentissage multi-agent et respect de contraintes organisationnelles, qu'elles soient imposées de manière explicite, implicite ou symbolique.

Couverture des critères par les travaux identifiés

La synthèse de la couverture de ces critères par les principales familles de travaux identifiés est présentée dans le Table 9.

Le Table 9 synthétise les travaux identifiés dans les trois axes définis par l'hypothèse (**H-TRN**) vis à vis des critères spécifiques.

Tout d'abord, les approches issues du champ du Safe Reinforcement Learning, comme CPO [118] ou Deep Constrained Q-Learning [83], se distinguent par leur capacité à fournir certaines garanties de sécurité ou de respect partiel des contraintes [134]. Ces travaux opèrent généralement à un niveau local, souvent au niveau des actions ou des politiques, et sont bien compatibles avec les algorithmes d'apprentissage profond. Toutefois, leur expressivité organisationnelle demeure faible, car elles ne modélisent pas explicitement les structures collectives ou les règles organisationnelles. De même, leur explicabilité reste

TABLE 9 : Couverture des critères par les principales familles de travaux sur l'intégration de contraintes/guidages organisationnels dans le MARL

Travaux / Critères	**Expressivité organisationnelle**	**Niveau d'intégration**	**Garantie de respect**	**Compatibilité apprentissage**	**Explicabilité**
Safe RL [134], CPO [118], Deep Constrained Q-Learning [83]	Faible	Action/Politique	Oui (partiel)	Oui	Faible
Reward shaping [181], shielding [128], feedback humain [18, 117]	Moyenne	Action/Trajectoire	Non (souple)	Oui	Faible
Constraint-Guided RL [73], MENTOR [18], RL contraint sans récompense explicite [61]	Moyenne	Politique	Oui (partiel)	Oui	Moyenne
Intégration de modèles organisationnels ($\mathcal{M}$OISE$^+$ [161], AGR [167]), hybridation apprentissage/organisation [140, 164]	Forte (potentielle)	Organisation	Non (verrou)	Non (verrou)	Forte

limité, rendant difficile l'interprétation des comportements multi-agents vis-à-vis des attentes organisationnelles. En somme, ces approches privilégient l'efficacité opérationnelle locale à la cohérence globale avec des contraintes organisationnelles complexes.

Ensuite, les méthodes comme le reward shaping [181], le shielding [128] ou l'intégration de feedback humain [18, 117] cherchent à influencer les agents via des mécanismes souples ou indirects. Bien que ces approches permettent une intégration à différents niveaux (actions ou trajectoires) et soient compatibles avec l'apprentissage, elles n'offrent pas de garanties formelles sur le respect des contraintes organisationnelles. Leur expressivité est relativement moyenne, dans la mesure où elles peuvent encoder des préférences, mais sans structure hiérarchique explicite. Par ailleurs, l'explicabilité est souvent faible, car la source de certaines décisions ou adaptations des agents est diluée dans la dynamique d'apprentissage et l'optimisation implicite des récompenses modifiées. Ces méthodes sont donc utiles pour influencer le comportement des agents sans imposer de structure rigide, mais elles peinent à faire émerger une coordination organisationnelle explicite.

Les travaux plus récents sur l'intégration de contraintes dans l'apprentissage multi-agent incluent notamment les approches de Constraint-Guided RL [73], MENTOR [18] ou l'apprentissage contraint sans fonction de récompense explicite [61]. Ces méthodes visent à intégrer des contraintes explicites ou des guidages hiérarchiques dans le processus d'apprentissage, offrant ainsi une meilleure compatibilité avec les exigences organisationnelles, bien que la couverture reste partielle. Elles constituent une avancée vers une prise en compte plus systématique des contraintes dans le MARL, tout en maintenant la capacité d'adaptation des agents.

Enfin, les approches basées sur l'intégration de modèles organisationnels explicites offrent une forte expressivité organisationnelle, car elles permettent de représenter formellement les rôles, missions et relations entre entités collectives. Parmi les modèles existants, $\mathcal{M}$OISE$^+$ s'impose comme le plus abouti et le plus largement utilisé dans la communauté SMA, surpassant notamment AGR [167]. Il permet de modéliser de manière intégrée les

spécifications structurelles, fonctionnelles et déontiques, ce qui facilite l'expression de contraintes organisationnelles complexes et leur compatibilité avec les approches d'apprentissage. Cependant, un verrou technique majeur subsiste : l'intégration directe de MOISE$^+$ avec les méthodes d'apprentissage est encore très limitée, voire inexistante, ce qui ne permet pas de garantir le respect effectif des contraintes organisationnelles lors de l'apprentissage. En revanche, ce modèle se distingue par son potentiel élevé d'explicabilité, puisqu'il rend transparentes les intentions collectives et la structure des décisions. Cette caractéristique ouvre une piste prometteuse pour l'hybridation entre apprentissage autonome et pilotage organisationnel, une direction encore peu explorée dans la littérature [140, 164].

Enfin, il convient de noter que la littérature sur le MARL a été largement synthétisée dans plusieurs revues récentes [71, 77], qui mettent en évidence la difficulté d'intégrer des contraintes organisationnelles explicites dans les approches classiques d'apprentissage multi-agent.

Analyse des travaux et verrous

L'analyse de l'état de l'art met en évidence plusieurs avancées dans l'intégration de contraintes dans le processus d'apprentissage multi-agent, notamment via des approches de *Safe RL*, de *reward shaping*, ou de *shielding*. Ces méthodes permettent d'orienter l'apprentissage en imposant des contraintes sur les actions, en modifiant la fonction de récompense, ou en guidant les agents à travers des feedbacks humains ou des mécanismes de filtrage d'actions. Elles offrent ainsi un certain contrôle sur les politiques apprises, en particulier pour éviter des comportements indésirables ou garantir le respect de règles de sûreté locales.

Cependant, ces approches présentent des limites majeures dès lors qu'il s'agit d'exprimer et de garantir le respect de contraintes organisationnelles complexes, telles que la répartition des rôles, la coordination structurée ou la réalisation de missions collectives. Les contraintes sont généralement intégrées à un niveau local (action, trajectoire, politique individuelle), sans prise en compte explicite de la structure organisationnelle globale du SMA.

Par ailleurs, les travaux issus de l'ingénierie des SMA, et notamment les modèles organisationnels symboliques comme MOISE$^+$, proposent une formalisation riche des rôles, missions et relations entre agents. Toutefois, il n'existe pas, à ce jour, de méthode permettant d'intégrer directement ces spécifications organisationnelles dans le processus d'apprentissage MARL, que ce soit via le reward shaping (pour inciter les agents à adopter des comportements conformes à leur rôle ou à leurs objectifs) ou via la restriction de l'espace des actions (pour forcer le respect de certaines contraintes organisationnelles).

En synthèse, le verrou principal identifié est l'absence d'un cadre unifié permettant de combiner :

- l'expressivité des modèles organisationnels symboliques (rôles, missions, permissions, obligations) ;
- et l'efficacité des techniques MARL sous contraintes (reward shaping, action masking, etc.)

pour guider ou contraindre l'apprentissage de politiques collectives respectant explicitement les spécifications organisationnelles.

Ce manque limite la capacité à concevoir des SMA à la fois adaptatifs, sûrs et explicables dans des contextes critiques comme la Cyberdéfense. Il justifie la nécessité de contributions méthodologiques permettant d'articuler modèles organisationnels et apprentissage, ce qui constituera l'un des axes centraux de la méthode proposée dans la partie suivante.

4.3 L'EXTRACTION DES SPÉCIFICATIONS ORGANISATIONNELLES ÉMERGENTES (H-ANL)

Recontextualisation du sous-problème dans la démarche de la thèse

L'extraction des spécifications organisationnelles émergentes constitue le troisième sous-problème central de la démarche de la thèse. Alors que les modèles organisationnels symboliques (i.e. $\mathcal{M}$OISE$^+$) permettent de prescrire explicitement des structures, rôles et missions lors de la conception d'un SMA, ces spécifications sont généralement absentes ou perdues dans les approches fondées sur le MARL. Les politiques apprises sont souvent représentées sous forme de réseaux de neurones opaques, rendant difficile l'analyse, la vérification ou l'explicabilité des comportements collectifs obtenus.

Dans ce contexte, la capacité à extraire a posteriori des structures organisationnelles émergentes (rôles, missions, objectifs collectifs, relations) à partir des trajectoires et politiques apprises devient un enjeu clé. Cette extraction permettrait de combler le fossé entre les approches prescriptives (où l'organisation est définie a priori) et les approches émergentes (où l'organisation résulte de l'apprentissage), en offrant un moyen d'analyser, d'expliquer et de comparer l'organisation implicite qui se forme au sein d'un SMA entraîné.

Ce sous-problème est donc directement lié à l'explicabilité, à la rétro-ingénierie et à l'amélioration continue des SMAs. Il ouvre la voie à une boucle de conception organisationnelle itérative, où les spécifications extraites peuvent être utilisées pour diagnostiquer des écarts, guider de nouveaux apprentissages ou réinjecter des contraintes organisationnelles adaptées. L'extraction des spécifications organisationnelles émergentes est ainsi essentielle pour garantir la transparence, la robustesse et l'évolutivité des SMA conçus par apprentissage.

Pour évaluer la pertinence des travaux identifiés, les critères spécifiques retenus sont :

- **Explicabilité** : capacité à fournir une description intelligible des décisions ou comportements des agents, localement ou globalement ; *ce critère est essentiel pour permettre l'analyse, la validation et la compréhension des politiques apprises, en particulier dans des contextes critiques où la transparence est requise*.

- **Extraction organisationnelle** : aptitude à inférer ou diagnostiquer des structures collectives, des rôles ou des missions à partir des trajectoires ou interactions observées ; *ce critère permet d'identifier les schémas organisationnels émergents et d'évaluer la cohérence des comportements collectifs vis-à-vis des objectifs du système*.

- **Automatisation** : degré d'automatisation du processus d'extraction ou d'analyse organisationnelle, sans intervention humaine majeure ; *une automatisation élevée facilite l'analyse à grande échelle, réduit la dépendance à l'expertise humaine et accélère le diagnostic des organisations apprises*.

- **Lien symbolique** : possibilité de relier les dynamiques collectives apprises à des modèles organisationnels symboliques ou à des structures explicites ; *ce critère est*

fondamental pour permettre la rétro-ingénierie, la comparaison et l'intégration des organisations extraites dans des cadres formels, favorisant ainsi la réutilisation et l'explicabilité globale.

*Hypothèse de restriction de l'espace de recherche (**H-ANL**)*

Nous reprenons l'hypothèse suivante : *Il est possible d'extraire, à partir des trajectoires et politiques apprises d'un SMA, des spécifications organisationnelles émergentes (telles que des structures de rôles, des missions ou des objectifs collectifs), permettant d'analyser, d'expliquer et de comparer l'organisation implicite via des techniques d'apprentissage machine.* Cette hypothèse restreint le périmètre de la revue de littérature aux travaux qui visent à :

- améliorer l'interprétabilité ou l'explicabilité des politiques multi-agents (analyse post-hoc, modèles interprétables, attribution de concepts) ;
- inférer ou diagnostiquer des structures collectives, des rôles ou des missions à partir de trajectoires ou d'interactions observées ;
- relier les dynamiques collectives apprises à des modèles organisationnels symboliques ou à des structures explicites.

Couverture des critères par les travaux identifiés

La synthèse de la couverture des critères par les principales familles de travaux identifiés est présentée dans le Table 10.

TABLE 10 : Couverture des critères par les principales familles de travaux sur l'extraction organisationnelle émergente

Travaux / Critères	**Explicabilité**	**Extraction organisationnelle**	**Automatisation**	**Lien symbolique**
Modèles interprétables (arbres, concepts, décomposition de valeur) [7, 8, 22, 25, 36, 52, 60]	Locale	Faible à moyenne	Moyenne	Faible
Analyse post-hoc (relevance, patching, Shapley, attribution de concepts) [7, 12, 56]	Locale	Faible	Moyenne	Faible
Inférence de rôles ou de dépendances (analyse sociale, clustering, modèles bayésiens, inférence de rôles) [32, 89, 100, 146, 177]	Moyenne	Moyenne	Faible à moyenne	Faible
Extraction organisationnelle symbolique (non couvert à ce jour)	–	–	–	–

La Table 10 synthétise les contributions des travaux identifiés en matière d'explicabilité, d'extraction organisationnelle, d'automatisation et de lien symbolique.

La première famille de travaux, centrée sur les modèles interprétables comme les arbres de décision ou les représentations conceptuelles, vise à rendre les politiques apprises plus

transparentes. Ces approches permettent d'obtenir une explicabilité principalement locale, c'est-à-dire au niveau de la décision d'un agent donné, en décomposant la politique en structures arborescentes ou symboliques lisibles. Le travail de Zhang [36] propose ainsi une méthode d'extraction d'arbres à partir de politiques MARL, combinant efficacité d'échantillonnage et transparence. De même, les travaux *Multi-Agent VIrtual PERimeter* (MAVIPER) de Milani et al. [25, 60] traduisent les réseaux de neurones en politiques arborescentes, exploitant la structure des décisions pour améliorer la lisibilité des comportements collectifs. D'autres contributions, comme celles de Zabounidis et al. [52], Iturria-Rivera et al. [22], Liu et al. [8] ou Li et al. [7], proposent respectivement l'intégration de concepts interprétables, la décomposition de la récompense (*Value-Decomposition Networks* – VDN [116], *Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning* – QMIX [114]), ou l'utilisation d'architectures hybrides et d'approximations Shapley pour rendre les politiques plus lisibles et interprétables. Si ces approches permettent une certaine extraction de structures organisationnelles implicites (rôles ou schémas d'action), leur couverture reste limitée à des missions simples ou à des environnements structurés. L'automatisation de cette extraction est toutefois bien amorcée grâce à des pipelines d'extraction semi-supervisés, bien que le lien avec des modèles organisationnels symboliques comme $\mathcal{M}$OISE$^+$ demeure marginal, voire inexistant.

La seconde catégorie, celle des approches d'analyse post-hoc, regroupe les méthodes qui n'interviennent pas lors de l'apprentissage, mais après coup, pour expliquer les décisions prises par un SMA entraîné. Ces méthodes incluent des techniques de type *SHapley Additive exPlanations* (SHAP), patching ou attribution de concepts. Elles permettent également une explicabilité locale, souvent centrée sur l'importance des caractéristiques dans les décisions individuelles. Grupen et al. [56] ont par exemple utilisé une approche fondée sur les concepts latents pour analyser les comportements émergents dans des environnements multi-agents, sans modifier l'entraînement initial. D'autres travaux, comme ceux de Poupart et al. [12] ou Li et al. [7], introduisent des méthodes post-hoc telles que la rétropropagation de la pertinence ou l'attribution basée sur la valeur de Shapley. Bien que ces méthodes soient prometteuses pour interpréter les politiques individuelles, elles restent limitées en ce qui concerne la détection de structures organisationnelles globales. L'automatisation de l'analyse est modérée : elle repose sur l'instrumentation du modèle entraîné, mais nécessite encore un effort d'interprétation humaine significatif. Comme pour les modèles interprétables, ces méthodes restent déconnectées de toute formalisation symbolique, ne produisant pas de représentation formelle d'une organisation émergente.

Enfin, les travaux centrés sur l'inférence de rôles ou de dépendances sociales cherchent à reconstituer des structures collectives ou hiérarchiques à partir des trajectoires d'agents. Cette ligne de recherche se distingue par une explicabilité plus globale, avec des tentatives d'identifier des rôles émergents, des missions collectives, ou encore des relations de dépendance entre agents. L'approche *Role-Oriented Multi-Agent Reinforcement Learning* (ROMA) de Wang et al. [89] repose sur la maximisation de l'information mutuelle entre les rôles et les trajectoires pour faire émerger des spécialisations comportementales. D'autres contributions, comme celles de Berenji et Vengerov [177], Yusuf et Baber [146], ou Serrino et al. [100], modélisent les dépendances ou identifient des rôles émergents à partir des interactions sociales ou de raisonnements bayésiens. Enfin, certaines approches récentes (comme [32]) commencent à explorer des méthodes hybrides combinant règles symboliques et apprentissage profond.

Analyse des travaux et verrous

L'analyse de l'état de l'art sur l'extraction des spécifications organisationnelles émergentes met en évidence plusieurs avancées, mais aussi des limites importantes. Les travaux existants se répartissent principalement en trois familles : (i) les modèles interprétables visant à rendre les politiques plus lisibles, (ii) les méthodes d'analyse post-hoc pour expliquer les décisions individuelles, et (iii) les approches d'inférence de rôles ou de dépendances sociales à partir des trajectoires.

Si ces contributions permettent d'améliorer l'explicabilité locale (niveau agent) ou d'identifier certains schémas collectifs, elles restent limitées pour ce qui est de l'extraction automatisée de structures organisationnelles complètes (rôles, missions, objectifs collectifs) et de leur formalisation symbolique. En particulier, aucune méthode ne propose aujourd'hui de relier systématiquement les comportements émergents à des modèles organisationnels formels tels que $\mathcal{M}$OISE$^+$.

Les travaux les plus prometteurs pour répondre à ce sous-problème proviennent du domaine de l'apprentissage non supervisé, et en particulier des techniques de clustering appliquées aux trajectoires d'agents. Il devient ainsi envisageable d'utiliser le clustering sur les observations ou les actions pour inférer des rôles (via l'identification de patterns comportementaux fréquents) et des objectifs (via la détection d'observations ou de situations récurrentes). Cette approche ouvre la voie à une extraction plus automatisée et potentiellement plus globale des structures organisationnelles émergentes.

Cependant, deux verrous majeurs subsistent :

1. **L'absence de cadre théorique pour l'évaluation de l'explicabilité organisationnelle :** Il manque aujourd'hui un framework et des notions quantitatives permettant de mesurer le niveau d'explicabilité atteint par les méthodes d'extraction, que ce soit au niveau des rôles, des missions ou des relations collectives.

2. **Le manque d'automatisation dans le paramétrage des méthodes :** Les techniques de clustering et d'inférence nécessitent souvent un réglage manuel des hyperparamètres (nombre de clusters, seuils, etc.), ce qui limite leur automatisation et leur applicabilité à grande échelle.

En résumé, bien que l'utilisation de l'apprentissage non supervisé et du clustering de trajectoires constitue une piste prometteuse pour l'extraction organisationnelle émergente, il reste nécessaire de lever ces deux verrous pour permettre une analyse automatisée, explicable et généralisable des organisations implicites dans les SMA entraînés par apprentissage. Cela justifie le besoin de contributions méthodologiques nouvelles, à la fois sur le plan théorique (cadre d'évaluation de l'explicabilité) et pratique (automatisation du processus d'inférence organisationnelle).

4.4 LE MAINTIEN DE COHÉRENCE ENTRE ENVIRONNEMENT SIMULÉ ET RÉEL (H-TRF)

Recontextualisation du sous-problème dans la démarche de la thèse

Le maintien de la cohérence entre l'environnement simulé (ou jumeau numérique) et l'environnement réel constitue un enjeu transversal et critique dans la démarche de la thèse. En effet, l'ensemble du processus de conception, d'entraînement et d'analyse des

politiques multi-agents repose sur la capacité à disposer d'une simulation fidèle, représentative des dynamiques et incertitudes du monde réel. Cette cohérence est essentielle pour garantir que les politiques apprises en simulation soient effectivement transférables, robustes et sûres lors de leur déploiement opérationnel.

Dans le contexte de la Cyberdéfense, où les environnements sont particulièrement dynamiques, partiellement observables et sujets à des évolutions rapides (nouvelles menaces, changements de topologie, comportements adverses imprévus), le risque de divergence entre simulation et réalité est élevé. Un écart non détecté ou non corrigé peut conduire à des politiques inefficaces, voire dangereuses, lors du passage au réel. Il est donc indispensable de mettre en place des mécanismes permettant de synchroniser, recalibrer ou adapter en continu le jumeau numérique, afin de maintenir sa pertinence tout au long du cycle de vie du système.

Ce sous-problème s'articule ainsi avec les autres étapes de la démarche : il conditionne la validité des apprentissages réalisés en simulation, la capacité à analyser les comportements émergents dans un contexte réaliste, et la possibilité de réinjecter les retours du terrain pour améliorer ou corriger le modèle simulé. Il s'agit donc d'un verrou central pour assurer la robustesse, la maintenabilité et la sécurité des SMAs conçus par apprentissage, en particulier dans des domaines critiques comme la Cyberdéfense. Pour conduire la revue de littérature, nous retenons les critères spécifiques suivants :

- **Fidélité du jumeau numérique** : capacité à représenter fidèlement les dynamiques et incertitudes du monde réel ; *ce critère est essentiel pour garantir que les politiques apprises en simulation soient pertinentes et applicables dans l'environnement réel, en évitant les écarts de comportement dus à une modélisation trop simplifiée ou inexacte*.

- **Robustesse au transfert** : aptitude à maintenir les performances et la sûreté lors du passage de simulation vers l'environnement réel ; *ce critère permet d'assurer que les politiques restent efficaces et sûres malgré les différences entre simulation et réalité, limitant ainsi les risques opérationnels lors du déploiement*.

- **Adaptation en ligne** : possibilité de recalibrer ou d'ajuster le modèle simulé en fonction des retours du réel ; *ce critère est indispensable pour suivre l'évolution de l'environnement réel et maintenir la pertinence du jumeau numérique dans la durée, notamment face à des menaces ou des changements imprévus*.

- **Sécurité et maintenabilité** : garanties sur l'absence de comportements dangereux ou imprévus lors du transfert et sur la capacité à maintenir le système dans la durée ; *ce critère vise à prévenir les défaillances ou dérives du système après transfert, tout en assurant la possibilité d'effectuer des mises à jour ou des corrections sans compromettre la sécurité globale*.

*Hypothèse de restriction de l'espace de recherche (**H-TRF**)*

Nous reprenons l'hypothèse suivante : *Il est possible de maintenir la cohérence entre l'environnement simulé (jumeau numérique) et l'environnement réel grâce à des mécanismes de couplage, d'adaptation ou de recalibrage, permettant un transfert sûr, adaptatif et maintenable des politiques apprises en simulation vers le réel, tout en assurant la fidélité du jumeau numérique face aux évolutions de l'environnement.*

Cette hypothèse restreint l'espace de recherche aux travaux qui visent à :

- synchroniser ou adapter dynamiquement le modèle simulé à partir des données ou des retours issus de l'environnement réel (domain adaptation, Sim2Real, calibration dynamique) ;
- assurer le transfert robuste des politiques apprises en simulation vers le réel (policy transfer, transfer learning, robust RL) ;
- détecter et corriger les écarts entre simulation et réalité (model discrepancy, online adaptation, feedback loop) pour garantir la robustesse et la sécurité du déploiement.

Couverture des critères par les travaux identifiés

La synthèse de la couverture des critères par les principales familles de travaux sur le maintien de cohérence entre environnement simulé et environnement réel est présentée dans le Table 11.

TABLE 11 : Couverture des critères par les principales familles de travaux sur le maintien de cohérence simulation/réel

Travaux / Critères	**Fidélité du jumeau**	**Robustesse au transfert**	**Adaptation en ligne**	**Sécurité / maintenabilité**
Domain adaptation / Sim2Real [126, 130]	Moyenne à forte	Moyenne à forte	Moyenne	Moyenne
Policy transfer / Robust RL [123]	Moyenne	Forte	Faible à moyenne	Moyenne
Online model calibration / Feedback loop [148]	Forte	Moyenne	Forte	Moyenne à forte
Synchronisation manuelle (recalibrage ponctuel) [74, 75]	Forte (statique)	Faible	Faible	Faible

Le **Domain adaptation** est une approche s'inscrivant dans le **Sim2Real** et vise à réduire l'écart entre simulation et réalité, soit en adaptant les données simulées, par exemple via la *domain randomization* [126], soit en utilisant des techniques adversariales pour rendre les représentations latentes invariantes au domaine [130]. Ces approches offrent généralement une bonne couverture en termes de fidélité et de robustesse, bien que l'adaptation en ligne reste partielle.

Les méthodes de **policy transfer** et de **robust reinforcement learning** [123] s'attachent à transférer des politiques apprises en simulation vers le monde réel, en maximisant la robustesse face aux incertitudes du domaine cible. Elles reposent souvent sur des mécanismes d'entraînement adversarial ou de perturbations, mais intègrent peu d'adaptation continue après le transfert.

Les approches d'**online model calibration** ou de **feedback loop**, telles que l'algorithme *Probabilistic Inference for Learning Control* (PILCO) [148], permettent de mettre à jour dynamiquement les modèles à partir de données issues du réel, assurant ainsi une adaptation permanente et une meilleure sécurité, au prix d'une complexité computationnelle accrue. Cette calibration dynamique relève des techniques d'*online system identification*.

Enfin, la **synchronisation manuelle** par recalibrage ponctuel reste une pratique courante dans les simulateurs industriels ou cybersécuritaires, comme CybORG [74] ou CyberBattleSim [75], bien qu'elle offre peu de garanties face à des environnements dynamiques et évolutifs (*manual resynchronization*).

En synthèse, les principales familles de travaux couvrent :

- *Robust RL* [123] pour le transfert de politiques ;
- *Domain randomization* [126] et *adversarial domain adaptation* [130] pour le Sim2Real ;
- *Online system identification* [148] pour la calibration dynamique ;
- *Manual resynchronization* dans les simulateurs industriels ou cyber (ex. : CybORG [74], CyberBattleSim [75]).

Analyse des travaux et verrous

L'analyse de l'état de l'art sur le maintien de cohérence entre environnement simulé et environnement réel met en évidence plusieurs avancées, mais aussi des limites structurantes. Les approches de *domain adaptation* et *Sim2Real* permettent de réduire l'écart entre simulation et réalité, en adaptant soit les données, soit les politiques pour les rendre plus robustes aux variations du monde réel. Cependant, ces méthodes se concentrent principalement sur le transfert initial et n'intègrent que rarement des mécanismes d'adaptation continue ou de recalibrage dynamique du modèle simulé après le déploiement.

Les travaux sur le *policy transfer* ou le *robust RL* visent à garantir que les politiques apprises en simulation restent performantes et sûres lors de leur transfert dans l'environnement réel. Néanmoins, ils n'abordent pas la question de la mise à jour du modèle simulé lui-même : une fois la politique transférée, le jumeau numérique n'est généralement pas recalibré pour suivre les évolutions du réel, ce qui peut conduire à une divergence progressive et à une perte de pertinence.

À l'inverse, les approches d'*online calibration* ou de *feedback loop* se focalisent sur la mise à jour dynamique du modèle simulé à partir des retours du réel, assurant ainsi une meilleure fidélité du jumeau numérique. Toutefois, ces méthodes n'intègrent pas explicitement le transfert ou l'adaptation des politiques multi-agents déjà déployées, ce qui limite leur capacité à garantir la robustesse et la sécurité du système dans son ensemble.

Enfin, la synchronisation manuelle (recalibrage ponctuel du modèle) reste une pratique courante dans l'industrie, mais elle est peu adaptée aux environnements dynamiques et évolutifs, car elle ne permet ni une adaptation fine ni une automatisation du processus.

En synthèse, il n'existe pas aujourd'hui de framework unificateur permettant de coupler de façon intégrée la mise à jour des politiques des agents déployés dans l'environnement réel et la mise à jour du modèle d'environnement simulé. La plupart des travaux identifiés couvrent une partie des objectifs (transfert de politiques ou recalibrage du modèle), mais aucun ne permet de les atteindre simultanément et de façon coordonnée. Par exemple, le *Robust RL* permet de transférer des politiques, mais ne prend pas en compte l'évolution du modèle simulé, tandis que la calibration en ligne ajuste le modèle, mais sans garantir l'adaptation des politiques en conséquence.

Le verrou principal réside donc dans l'absence d'un cadre méthodologique ou d'un framework de jumeau numérique capable d'assurer à la fois :

- la mise à jour continue du modèle simulé en fonction des évolutions de l'environnement réel ;
- et la mise à jour ou l'adaptation des politiques multi-agents déployées, pour garantir leur robustesse, leur sécurité et leur maintenabilité.

Ce manque limite la capacité à déployer durablement des SMA dans des environnements réels évolutifs, en particulier dans des domaines critiques comme la Cyberdéfense.

Il justifie la nécessité de contributions nouvelles visant à articuler, dans un même cadre, l'adaptation conjointe du jumeau numérique et des politiques multi-agents, afin d'assurer une cohérence et une performance durables du système.

SYNTHÈSE DES TRAVAUX RETENUS ET VERROUS IDENTIFIÉS

La revue conduite sur les quatre activités fondamentales a permis (i) d'identifier les familles de travaux les plus prometteuses pour la conception automatisée d'un SMA de Cyberdéfense, (ii) de mettre en évidence les verrous scientifiques qui justifient la nécessité de nouvelles contributions méthodologiques.
La Table 12 condense les principaux enseignements de la revue de littérature menée dans ce chapitre, en croisant les travaux prometteurs identifiés, les verrous scientifiques, les limites des approches existantes et les besoins méthodologiques associés à chacune des quatre hypothèses (**H-MOD**, **H-TRN**, **H-ANL**, **H-TRF**). Cette vue d'ensemble permet de mieux situer les points d'appui et les manques actuels dans la conception de SMA de Cyberdéfense.

Concernant la modélisation (**H-MOD**), les formalismes markoviens et les World Models apparaissent comme des contributions majeures pour représenter ou apprendre les dynamiques d'environnements complexes. Toutefois, leur extension au multi-agent reste limitée, et la lourdeur de la modélisation manuelle constituent un verrou majeur. Cela révèle un besoin méthodologique fort : concevoir des modèles hybrides exploitant à la fois la généricité des cadres markoviens et l'automatisation offerte par les approches data-driven, afin de faciliter la construction d'environnements simulés pertinents pour la Cyberdéfense.

Pour l'intégration de contraintes organisationnelles (**H-TRN**), les approches issues du *Safe RL* ou du *Constraint-Guided RL* montrent une capacité à contraindre ou guider localement l'apprentissage, mais elles manquent d'expressivité organisationnelle et ne permettent pas de représenter des structures collectives explicites. En parallèle, les modèles organisationnels symboliques comme $\mathcal{M}$OISE$^+$ offrent une forte expressivité, mais restent difficilement intégrables au processus d'apprentissage. Le besoin identifié est donc celui d'un cadre unifié permettant d'hybrider apprentissage connexionniste et contraintes symboliques, afin de guider l'apprentissage multi-agent tout en garantissant le respect de structures organisationnelles.

Du point de vue de l'analyse et de l'explicabilité organisationnelle (**H-ANL**), des travaux comme MAVIPER ou ROMA ont ouvert la voie à l'extraction de structures implicites (par exemple des rôles émergents) et à une meilleure lisibilité des politiques apprises. Cependant, les méthodes actuelles restent majoritairement locales et peu automatisées, et aucun cadre théorique ne permet d'évaluer de manière systématique l'explicabilité organisationnelle. Il apparaît donc nécessaire de développer des approches permettant d'inférer automatiquement des rôles, missions ou objectifs collectifs à partir de trajectoires, et de relier ces résultats à des modèles symboliques existants, afin d'offrir une explicabilité globale et opérationnelle.

Enfin, le maintien de la cohérence entre environnement simulé et environnement réel (**H-TRF**) a été largement abordé dans la littérature via des techniques de *domain adaptation*, de *robust RL* ou de calibration dynamique. Si chacune de ces approches couvre une partie des objectifs, aucune ne propose de mécanisme intégré permettant à la fois de mettre à jour le jumeau numérique et d'adapter en conséquence les politiques déployées. Ce manque souligne la nécessité d'un framework unificateur articulant la mise à jour continue du modèle simulé et l'adaptation conjointe des politiques multi-agents,

TABLE 12 : Synthèse des travaux retenus, verrous, limites et besoins méthodologiques par hypothèse

Hyp.	Travaux retenus	Verrous principaux	Limites des travaux existants	Besoins méthodologiques
H-MOD	Dec-POMDP comme cadre adaptable ; World Models pour la modélisation automatique	Lourdeur de la modélisation manuelle ; absence de bibliothèques spécialisées cyber ; extension multi-agent des World Models non résolue	World Models limités au mono-agent ou à des contextes simples ; peu d'approches exploitant des observations distribuées ; modèles markoviens trop simples pour aider à la modélisation manuelle	Apprendre un World Model multi-agent structuré autour de représentations latentes ; proposer un modèle markovien utilisable pour modéliser un environnement de cyberdéfense
H-TRN	Safe RL (CPO, DCQL) ; Constraint-Guided RL ; intégration potentielle de modèles organisationnels (MOISE+)	Faible expressivité organisationnelle des méthodes existantes ; absence de cadre unifié entre organisation symbolique et MARL ; manque de garanties globales	Constrained RL ne prend pas en compte les structures organisationnelles ; intégration de spécifications symboliques limitée à l'exécution	Introduire des contraintes symboliques dans le processus MARL pour guider l'apprentissage et filtrer les actions
H-ANL	MAVIPER et modèles interprétables ; clustering de trajectoires pour inférence de rôles ; ROMA	Absence de lien avec modèles symboliques ; manque d'automatisation ; pas de cadre d'évaluation de l'explicabilité organisationnelle	Explicabilité surtout locale (niveau agent/action) ; peu de travaux inférant des rôles, missions ou objectifs collectifs	Inférer automatiquement rôles, missions et objectifs à partir de trajectoires ; proposer un cadre théorique pour l'explicabilité organisationnelle
H-TRF	Domain adaptation / Sim2Real ; Robust RL ; Online calibration (PILCO)	Approches partielles (transfert ou recalibrage, mais pas les deux) ; absence de cadre intégré de jumeau numérique adaptatif et de mise à jour conjointe des politiques	Domain adaptation et Sim2Real se concentrent sur le transfert initial sans recalibrage dynamique ; robust RL ne met pas à jour le modèle simulé ; calibration en ligne sans adaptation des politiques	Développer un framework unificateur de jumeau numérique couplant mise à jour continue du modèle simulé et adaptation conjointe des politiques multi-agents

condition indispensable pour garantir la robustesse et la sécurité des systèmes dans des environnements dynamiques.

4.5 BILAN

En conclusion, ce chapitre a permis de dresser un panorama critique et structuré de l'état de l'art, en mettant en évidence les avancées majeures et les verrous persistants relatifs à la conception automatisée de SMAs pour la Cyberdéfense. En résumé, un double constat peut être établi : d'une part, l'existence de travaux prometteurs offrant des briques méthodologiques solides pour chaque sous-problème ; d'autre part, la persistance de verrous

transverses, liés notamment au manque d'automatisation, à la faible intégration entre approches symboliques et connexionnistes, et à l'absence de cadres unifiés couvrant plusieurs dimensions simultanément. Ces constats motivent directement les contributions proposées dans la suite du manuscrit, qui visent à lever ces limites par une méthode orchestrant les quatre activités de manière cohérente et intégrée [31]. L'analyse croisée des différentes approches a révélé la nécessité d'une intégration plus étroite entre modélisation, apprentissage, organisation et transfert, afin de répondre aux exigences de robustesse, d'adaptabilité et d'explicabilité propres à ce domaine. Le chapitre suivant approfondit donc sur le plan technique les travaux identifiés comme les plus prometteurs, afin de faire ressortir plus finement les verrous identifiés et de préparer leur levée par de nouvelles contributions.

5

LES TRAVAUX ET CONCEPTS THÉORIQUES MOBILISÉS

Ce chapitre complète l'analyse critique menée au Chapitre 4 en introduisant les fondements théoriques qui serviront de base aux contributions méthodologiques présentées dans la Partie III. Il présente de manière formelle les principaux cadres, méthodes et concepts associés aux quatre sous-problèmes identifiés (**H-MOD**, **H-TRN**, **H-ANL**, **H-TRF**). L'objectif est double : (i) disposer d'un socle théorique solide pour circonscrire les verrous mis en évidence, et (ii) préparer la formalisation de la méthode proposée.

Guide de lecture du chapitre. Ce chapitre peut être lu en suivant la chaîne **MOD** → **TRN** → **ANL** → **TRF**. Pour chaque bloc, nous explicitons d'abord les concepts mobilisés, puis leur rôle dans la méthode MAMAD. Ce choix vise à limiter la surcharge cognitive liée aux acronymes, tout en conservant la précision de la revue.

5.1 MODÉLISATION DE L'ENVIRONNEMENT (MOD)

La modélisation de l'environnement constitue la première étape fondamentale dans la conception de SMAs pour la Cyberdéfense. Elle vise à fournir une représentation formelle ou apprise des dynamiques, des interactions et des contraintes propres au domaine étudié. Cette section présente les principaux cadres et approches permettant de représenter un environnement de Cyberdéfense, en mettant l'accent sur les modèles formels issus de la théorie de la décision séquentielle, ainsi que sur les méthodes d'apprentissage automatique permettant de construire des modèles de monde à partir de données. L'objectif est de poser les bases nécessaires à l'intégration de l'apprentissage multi-agent et des contraintes organisationnelles dans les étapes ultérieures de la démarche.

5.1.1 *Les modèles de représentation d'un environnement de Cyberdéfense*

Peu de travaux traitent directement de la modélisation des cyberattaquants et des Cyberdéfenseurs qui s'affrontent dans un système hôte en réseau. En effet, les travaux disponibles sur le sujet proposent principalement une méthode de modélisation des actions d'un cyberattaquant unique dans des scénarios d'attaque spécifiques, tandis que la Cyberdéfense est généralement envisagée de manière optionnelle, en réaction.

À notre connaissance, il n'existe aucun cadre formel qui modélise précisément à la fois les agents collaboratifs attaquants et défenseurs dans un réseau tout en étant indépendant du contexte d'application. Néanmoins, certains travaux fournissent des approches potentielles pour modéliser un environnement de nœuds en réseau et/ou les interactions entre agents. De plus, pour de nombreux travaux de modélisation avancés, l'approche multi-agents n'est pas entièrement satisfaite dans le sens où les agents sont conçus à partir de la connaissance de l'ensemble de l'environnement. Néanmoins, indépendamment du niveau d'abstraction et du type de support, les travaux considérés pourraient être étendus pour modéliser l'impact des actions des agents cyberattaquants et Cyberdéfenseurs sur un environnement en réseau. Peu d'autres modélisations proviennent d'approches de simulation ou de réseaux réels par le biais de l'émulation/virtualisation.

Graphe d'attaque : Les graphiques d'attaque [186] sont des représentations graphiques des différentes façons dont un attaquant peut exploiter les vulnérabilités d'un système en réseau. Ils représentent le système comme un ensemble de nœuds (tels que des ordinateurs, des applications ou des connexions réseau) et les attaques possibles comme des arêtes entre ces nœuds. Le graphique montre comment un attaquant peut se déplacer d'un nœud à un autre en exploitant des vulnérabilités et exprime les conséquences sur le réseau [186]. Les graphiques d'attaque peuvent être utilisés pour identifier les vulnérabilités les plus critiques d'un système en réseau et aider le défenseur à hiérarchiser ses efforts pour sécuriser ces vulnérabilités dans ce système. Un exemple de graphe d'attaque est donné en Figure 12.

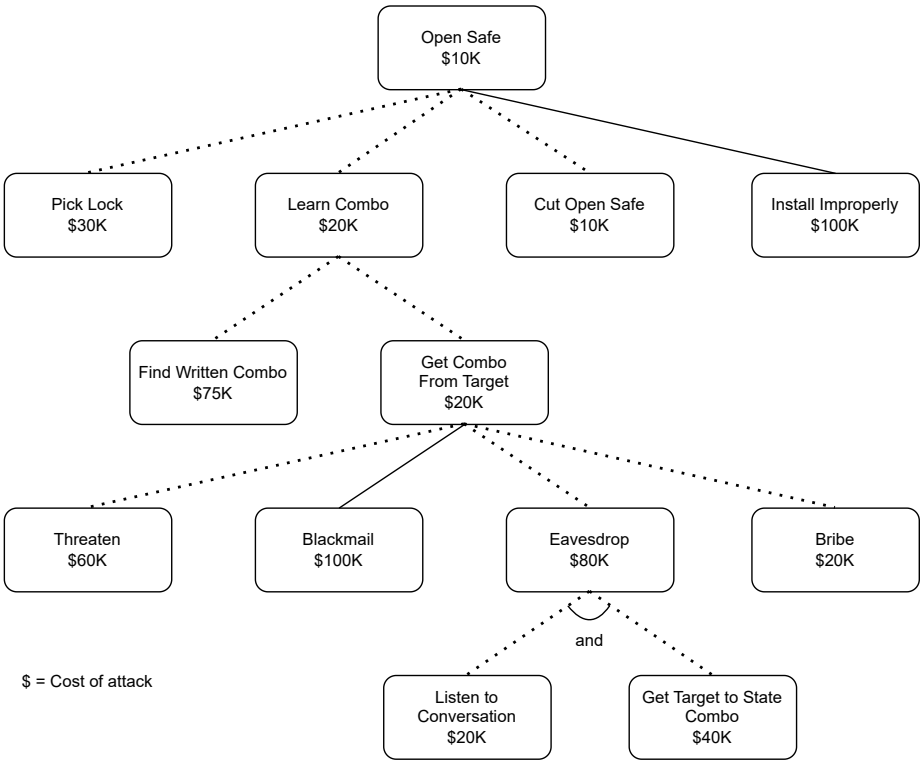


FIGURE 12 : Illustration d'un graphe d'attaque décrivant un scénario de compromission d'un coffre-fort. (adapté de [183]) : L'objectif racine *Open Safe* est décomposé en quatre voies principales : *Pick Lock* ($30K), *Learn Combo* ($20K), *Cut Open Safe* ($10K) et *Install Improperly* ($100K). Par défaut, un nœud est de type OU (réaliser l'un des enfants suffit) ; lorsqu'un *and* est indiqué, il s'agit d'un ET (tous les enfants sont requis). Les montants représentent des coûts estimés : pour un OU, le coût du parent est le *minimum* des coûts enfants ; pour un ET, les coûts *s'additionnent*.

Arbres attaque-défense : Les arbres attaque-défense [151] (arbres AD) sont des modèles graphiques représentant les objectifs de l'attaquant et les contre-mesures du défenseur sous la forme d'une structure arborescente. Les arbres AD fournissent une représentation plus abstraite du système et des objectifs des attaquants, tandis que les graphes d'attaque fournissent une représentation plus concrète des composants du système et de leurs relations. Un exemple d'arbre AD est illustré en Figure 13. La racine de l'arbre AD représente l'objectif ultime des cyberattaquants. Les sous-nœuds associés aux branches représentent les différentes stratégies d'attaque que l'attaquant pourrait utiliser pour atteindre son objectif. Ils peuvent être accompagnés de contre-mesures préventives ou réactives du

défenseur (pare-feu, systèmes de détection d'intrusion, plans d'intervention en cas d'incident, etc.). Les arbres AD permettent d'identifier les points faibles de la défense d'un système [151].

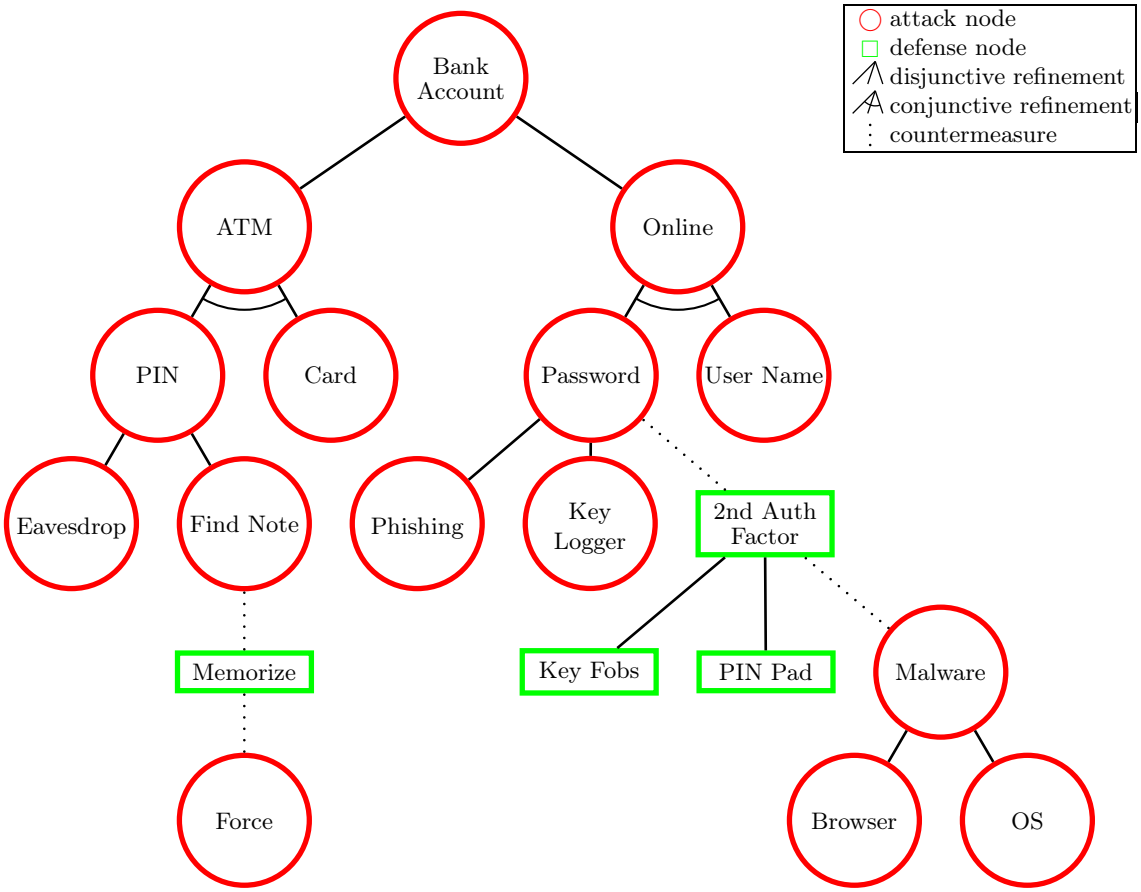


FIGURE 13 : Illustration d'ADTree décrivant un scénario d'attaque sur un compte bancaire (tirée de [151]) : L'accès au compte peut être fait via un guichet automatique ou en ligne. Pour ce dernier cas, il est nécessaire d'avoir un identifiant et un mot de passe obtenables par phishing ou un *Key Logger*. Une contremesure à ces attaques est la double authentification construite avec une clé *fob* ou un code pin.

Modélisation par réseaux de Petri : Les réseaux de Petri pouvant être utilisés pour décrire des processus concurrents, certains travaux ont cherché à modéliser les attaquants et les défenseurs dans un système en réseau. Les attaques extraites de bases de données peuvent être modélisées à l'aide de réseaux de Petri afin d'intégrer les cyberattaquants et les Cyberdéfenseurs, leurs stratégies et le coût de leurs actions, comme dans [62]. Les réseaux de Petri se révèlent également utiles pour modéliser les attaques par injection de langage de requête structuré afin d'inclure les stratégies des joueurs [78]. Ils sont utilisés comme cadre pour évaluer et comparer plusieurs modèles d'attaque. Dans [90], le logiciel malveillant *Mirai* a aussi été exprimé sous la forme d'un modèle formel avec des réseaux de Petri, permettant de simuler un combat entre un agent défenseur et *Mirai*.

Modèles de jeu : Certains travaux ont proposé de modéliser les interactions entre les attaquants et les défenseurs dans un réseau comme des joueurs dans un jeu, où chaque joueur dispose d'un ensemble d'actions qu'il peut effectuer. Parmi les travaux notables,

citons : Panfili et al. [112], où un jeu à somme générale multi-agents opposant un attaquant à un défenseur est utilisé pour trouver un compromis optimal entre les actions de prévention et les coûts ; Attiah et al. [102], où un cadre théorique de jeu dynamique est proposé pour analyser les interactions entre l'attaquant et le défenseur comme un jeu de sécurité non coopératif ; et Xiaolin et al. [158], qui utilisent des modèles de processus de Markov pour évaluer les risques dans les systèmes en réseau.
Certaines approches fondées sur la théorie des jeux s'inscrivent dans le cadre des "jeux stochastiques partiellement observables" (POSG) ou, plus précisément, dans celui des "processus de décision markoviens décentralisés partiellement observables" (Dec-POMDP). Les POSG et les Dec-POMDP sont tous deux des cadres de modélisation mathématique des problèmes de prise de décision dans lesquels des agents interagissent entre eux et dans un environnement stochastique [150]. Dans un POSG, un groupe d'agents interagit avec un environnement stochastique et partiellement observable. Chaque agent agit en fonction de ses propres observations et d'une politique locale. Les agents peuvent avoir des objectifs différents, car chaque agent a sa propre fonction de récompense et le jeu est généralement supposé être non coopératif [87]. Dans un Dec-POMDP, plusieurs agents peuvent avoir une fonction de récompense commune et peuvent coordonner leurs actions pour atteindre un objectif commun, notamment en étant capables de communiquer [142].

5.1.2 *Positionnement vis-à-vis des Stackelberg Security Games*

Les *Stackelberg Security Games* (SSG) constituent un cadre majeur de la cybersécurité décisionnelle [143]. Ils formalisent une interaction leader/suiveur : le défenseur choisit une stratégie de protection, puis l'attaquant réagit.

Ce cadre partage un objectif commun avec notre formulation : optimiser une politique de défense sous contraintes de ressources et face à un adversaire. Néanmoins, il existe trois différences structurantes avec notre périmètre :

- les SSG reposent souvent sur une vision stratégique statique ou quasi-statique, alors que nous traitons un processus séquentiel et partiellement observable ;
- notre cadre vise des SMAs avec coordination explicite entre plusieurs défenseurs (et éventuellement plusieurs attaquants), plutôt qu'un couple leader/suiveur agrégé ;
- les contraintes organisationnelles (rôles, missions, obligations/permissions) sont représentées explicitement via $\mathcal{M}$OISE$^+$ puis injectées dans l'apprentissage, ce qui dépasse la seule allocation stratégique.

Notre choix de formulation par Dec-POMDP/*Observation-based Dec-POMDP* (ODec-POMDP) et MARL contraint permet ainsi d'articuler, dans une même boucle, les quatre activités **MOD**–**TRN**–**ANL**–**TRF**. Les SSG restent néanmoins complémentaires : ils peuvent servir de cadre de positionnement théorique et de point d'entrée pour des allocations défensives tactiques en amont de l'apprentissage.

5.1.3 *Le modèle Dec-POMDP*

Pour appliquer des techniques MARL, il est nécessaire de s'appuyer sur un cadre Markovien pour formaliser les observations, actions, récompense, etc. Nous nous basons sur le cadre du Dec-POMDP [132]. Les Dec-POMDP permettent de modéliser la coordination décentralisée entre agents dans des contextes à observabilité partielle, ce qui les rend

particulièrement adaptés à l'intégration de contraintes organisationnelles. Contrairement aux POSG, le Dec-POMDP utilise une fonction de récompense commune, favorisant ainsi la collaboration [141].

Un Dec-POMDP $d \in D$ (avec D l'ensemble des Dec-POMDP) est défini par un 7-uplet $d = (S, \{A_i\}, T, R, \{\Omega_i\}, O, \gamma)$ où :

- $S = \{s_1, ..., s_{|S|}\}$: l'ensemble des états possibles.
- $A_i = \{a_1^i, ..., a_{|A_i|}^i\}$: l'ensemble des actions possibles pour l'agent i.
- T tel que $T(s, a, s') = \mathbb{P}(s'|s, a)$: la probabilité de transition conditionnelle entre états.
- $R : S \times A \times S \to \mathbb{R}$: la fonction de récompense.
- $\Omega_i = \{o_1^i, ..., o_{|\Omega_i|}^i\}$: l'ensemble des observations possibles pour l'agent ag_i.
- O tel que $O(s', a, o) = \mathbb{P}(o|s', a)$: la probabilité conditionnelle d'observer o depuis s' après avoir effectué a.
- $\gamma \in [0, 1]$: le facteur d'actualisation qui décrit l'importance des récompenses futures par rapport aux récompenses immédiates (i.e spectre entre un comportement glouton et un comportement prévenant).

En considérant m **équipes** (ou **groupes**) contenant chacune plusieurs agents parmi $\mathcal{A}$, nous reprenons le formalisme minimal nécessaire à la résolution d'un Dec-POMDP pour une équipe donnée $i, 0 \leqslant i \leqslant m$, composée de n agents [19, 141] :

- Π : l'ensemble des politiques. Une **politique** $\pi \in \Pi, \pi : \Omega \to A$ est une fonction déterministe qui associe à chaque observation une action. Elle représente la logique interne de l'agent.
- Π_{joint} : l'ensemble des politiques conjointes. Une **politique conjointe** $\pi_{joint} \in \Pi_{joint}, \pi_{joint} : \Omega^n \to A^n = \Pi^n$ associe une action à chaque agent en fonction de son observation, et peut être vue comme l'ensemble des politiques utilisées par les agents.
- H : l'ensemble des historiques. Un **historique** sur $z \in \mathbb{N}$ étapes est un z-uplet $h = ((\omega_k, a_k)|k \leqslant z, \omega \in \Omega, a \in A)$.
- H_{joint} : l'ensemble des historiques conjoints. Un **historique conjoint** sur z étapes $h_{joint} \in H_{joint}, h_{joint} = \{h_1, h_2, ..., h_n\}$ est l'ensemble des historiques des agents.
- $U_{joint,i}(\langle \pi_{joint,i}, \pi_{joint,-i} \rangle) : \Pi_{joint} \to \mathbb{R}$: la **récompense cumulée espérée** pour l'équipe i sur un horizon fini, avec $\pi_{joint,i}$ la politique conjointe de l'équipe i et $\pi_{joint,-i}$ les politiques conjointes des autres équipes (considérées comme fixes).
- $BR_{joint,i}(\pi_{joint,i}) = \arg\max_{\pi_{joint,i}} U(\langle \pi_{joint,i}, \pi_{joint,-i} \rangle)$: le **meilleur répondant** $\pi^*_{joint,i}$ tel qu'aucune modification de politique ne permettrait d'obtenir une récompense supérieure à $U_i^* = U_{joint,i}(\langle \pi^*_{joint,i}, \pi_{joint,-i} \rangle)$.
- $SR_{joint,i}(\pi_{joint,i}, s) = \{\pi_{joint,i} \mid U(\langle \pi_{joint,i}, \pi_{joint,-i} \rangle) \geqslant s\}$: la **réponse suffisante**, c'est-à-dire l'ensemble des politiques conjointes atteignant au moins une récompense cumulée attendue $s \in \mathbb{R}, s \leqslant U_i^*$.

On appelle **résolution du Dec-POMDP** la recherche d'une politique conjointe $\pi^j \in \Pi^j$ telle que $U_{joint,i}(\pi^j) \geqslant s$, atteignant une récompense cumulée espérée au moins égale à un seuil $s \in \mathbb{R}$.

5.1.4 *Les modèles de monde*

En RL, et en particulier en contexte d'observabilité partielle, les **modèles du monde** [96, 108] ou *World Models* visent à apprendre des modèles internes approximant à la fois la dynamique de la fonction de transition et d'observation conjointement. Les *World Models* permettent aux agents d'effectuer de la planification, d'améliorer l'efficacité échantillonnale, et de faciliter l'exploration sûre en permettant à l'agent de simuler des scénarios futurs. Cette approche appartient au paradigme du *Model-based Reinforcement Learning* (MBRL) [85], et se révèle particulièrement utile pour construire automatiquement des modèles de simulation à haute fidélité même en l'absence de représentation explicite de l'environnement.

Formellement, à chaque pas de temps t, on note $o_t \in \Omega$ l'observation courante, $a_t \in A$ l'action réalisée, et $\tilde{h}_{t-1} \in \mathcal{H}$ l'état caché récurrent résumant l'historique d'interaction jusqu'à $t-1$. Étant donné que les observations sont généralement de grande dimension (par exemple, des images ou des vecteurs d'état complexes), un encodeur $\mathrm{Enc} : \Omega \to Z$ est appliqué pour projeter les observations dans un espace latent compact Z, avec $z_t = \mathrm{Enc}(o_t)$, où $\dim(Z) \ll \dim(\Omega)$.

La structure temporelle principale est modélisée à l'aide d'un **Modèle Dynamique Latent Récurrent (*Recurrent Latent Dynamics Model* – RLDM)** [96] $\mathcal{T}^z = f(g(h_{t-1}, z_t, a_t))$, qui prédit le prochain état latent $\hat{z}_{t+1}$ en mettant à jour l'état récurrent via f et en appliquant une dynamique latente via g :

$$h_t = f(h_{t-1}, z_t, a_t), \quad \hat{z}_{t+1} = g(h_t)$$

où $f(\cdot)$ correspond typiquement à un réseau de neurones récurrent *Recurrent Neural Network* (RNN) (par exemple un *Long-Short Term Memory* – LSTM [187]) appliqué à la concaténation de h_{t-1}, z_t et a_t, et $g(\cdot)$ est une fonction (souvent implémentée par un *Multi-Layer Perceptron* – MLP) mappant l'état récurrent vers la représentation latente de la prochaine observation.

L'état latent prédit est ensuite décodé par $\mathrm{Dec} : Z \to \Omega$ pour produire l'observation prédite $\hat{o}_{t+1} = \mathrm{Dec}(\hat{z}_{t+1})$. L'ensemble du modèle est entraîné conjointement pour minimiser à la fois la *perte de reconstruction* $\|o_{t+1} - \hat{o}_{t+1}\|$ dans l'espace d'observation, et éventuellement une *perte de prédiction latente* pour stabiliser l'apprentissage de la dynamique latente.

L'état caché récurrent $\tilde{h}_t$ joue le rôle d'un résumé compact de l'historique complet d'interaction jusqu'au temps t, évitant ainsi d'avoir à stocker explicitement de longues séquences observation-action.

Par souci de concision, nous définissons la composition complète qui associe directement observation courante, action et état récurrent à l'observation prédite suivante sous la forme du **Modèle de Prédiction d'Observation** (*Observation Prediction Model* – OPM) :

$$\mathcal{T}(h_{t-1}, o_t, a_t) := \mathrm{Dec}(g(f(h_{t-1}, \mathrm{Enc}(o_t), a_t))) = \hat{o}_{t+1}.$$

La Figure 14 illustre l'architecture d'un *World Model* comprenant l'Auto-encodeur et l'OPM.

Phase d'entraînement de l'auto-encodeur : Un auto-encodeur, tel qu'un *Variational Auto Encoder* (VAE), est d'abord entraîné à encoder et décoder les observations en représentations latentes. L'objectif est de minimiser l'écart entre les observations réelles et les observations décodées.

Initialisation et traitement des transitions : Initialement, l'état caché récurrent $\tilde{h}_{t-1}$ est initialisé au vecteur nul. Pour chaque historique et chaque transition, un vecteur d'entrée

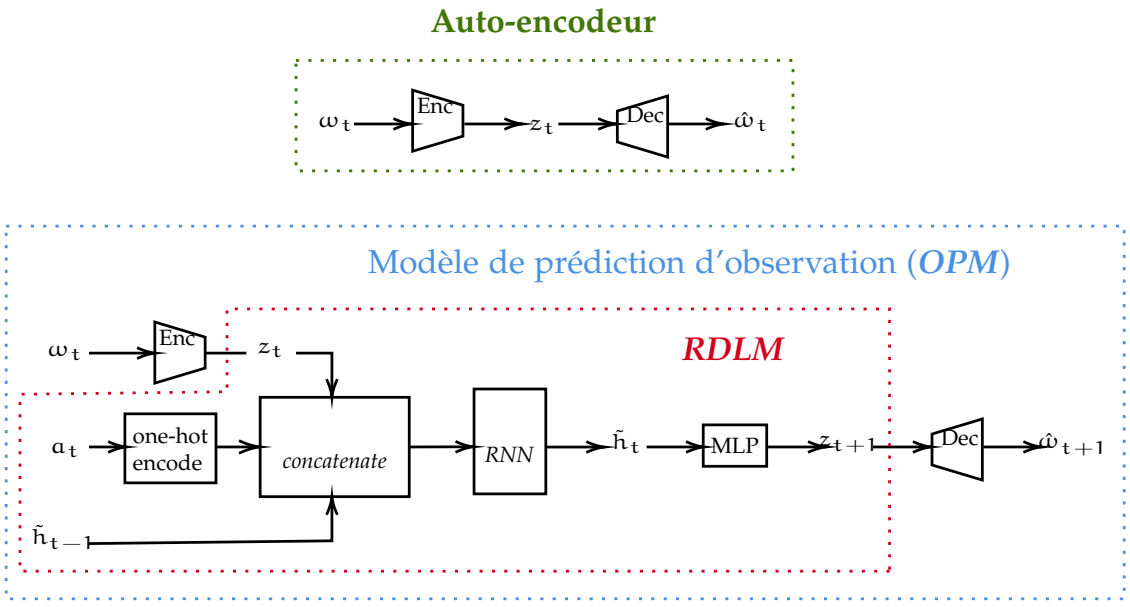


FIGURE 14 : Illustration de l'architecture d'un *World Model* comprenant l'Auto-encodeur et l'OPM

est construit en concaténant trois éléments : la représentation de l'observation z_t, l'action a_t (après encodage one-hot) et l'état caché récurrent $\tilde{h}_{t-1}$.

Fonctionnement du RLDM : Ce vecteur d'entrée est traité par le RLDM selon un processus en deux étapes. D'abord, il passe par le RNN qui met à jour l'état caché récurrent avec les nouvelles transitions pour obtenir $\hat{h}_t$. Ensuite, ce vecteur est transmis à un MLP qui détermine la représentation latente de l'observation suivante $\hat{z}_{t+1}$.

Entraînement et prédiction : Le RLDM est entraîné à minimiser l'erreur quadratique entre l'observation prédite et l'observation réelle. Une fois l'entraînement terminé, une représentation latente d'observation prédite peut être décodée en une observation prédite ω_{t+1}.

5.2 APPRENTISSAGE PAR RENFORCEMENT SOUS CONTRAINTES (TRN)

L'apprentissage par renforcement sous contraintes vise à doter les agents de la capacité à optimiser leur comportement tout en respectant des exigences additionnelles, telles que la sûreté, l'équité ou des règles organisationnelles. Cette section présente les principaux cadres théoriques et techniques permettant d'intégrer explicitement ou implicitement des contraintes dans le processus d'apprentissage, en s'appuyant sur les extensions du RL classique, les méthodes de Safe RL, ainsi que les mécanismes de guidage organisationnel.

5.2.1 *Apprentissage par renforcement multi-agent*

L'apprentissage par renforcement (RL) est un cadre formel dans lequel un agent apprend à agir dans un environnement inconnu en interagissant avec lui. À chaque étape, l'agent observe un état (ou une observation partielle), exécute une action, reçoit une récompense, et perçoit un nouvel état. L'objectif est de maximiser la récompense cumulée à long terme, généralement modélisée par une fonction de retour espéré.

Formellement, le problème est souvent représenté comme un processus de décision de Markov (MDP), défini par un quintuplet $\langle S, A, T, R, \gamma \rangle$, où :

- S est l'ensemble des états ;
- A est l'ensemble des actions possibles ;
- $T : S \times A \to \mathcal{P}(S)$ est la fonction de transition ;
- $R : S \times A \to \mathbb{R}$ est la fonction de récompense ;
- $\gamma \in [0, 1]$ est le facteur d'actualisation.

L'agent apprend une politique $\pi : S \to A$ (ou stochastique) qui maximise la somme des récompenses escomptées. Dans le cas partiellement observable (POMDP), les états sont inaccessibles, et l'agent agit à partir d'observations et d'un historique.

Dans le cas multi-agent, plusieurs agents interagissent simultanément avec l'environnement. Le problème devient plus complexe, car :

- **L'environnement devient non-stationnaire** : chaque agent modifie l'environnement et perturbe l'apprentissage des autres ;
- **L'exploration devient conjointe** : les conséquences d'une action peuvent dépendre du comportement des autres ;
- **Le crédit d'attribution est difficile** : relier une récompense à l'action d'un agent spécifique devient ambigu.

Le MARL traite de ces difficultés en adaptant les méthodes de RL à ce contexte. Deux grandes approches peuvent être distinguées :

- **Apprentissage indépendant (Independent Learners)** : chaque agent apprend sa politique en considérant les autres comme partie de l'environnement (simplifie la mise en œuvre, mais génère de l'instabilité) ;
- **Apprentissage centralisé avec exécution décentralisée (*Centralized Training Decentralized Execution* – CTDE)** : l'apprentissage est fait de manière coordonnée, avec accès à des informations globales (états, récompenses), mais les politiques finales doivent pouvoir s'exécuter de façon autonome.

Le MARL a été appliqué avec succès dans plusieurs domaines : coordination de robots, jeux coopératifs, gestion de trafic, systèmes énergétiques, etc. Dans le contexte de la Cyberdéfense, il offre un potentiel intéressant pour concevoir des politiques adaptatives capables de répondre à des menaces dynamiques et partiellement observées.

Cependant, plusieurs limites persistent :

- **La difficulté de convergence** dans des environnements complexes ou compétitifs ;
- **Le manque de garanties de sûreté ou de respect de contraintes** ;
- **Le peu d'explicabilité des politiques apprises**, souvent représentées par des réseaux de neurones ;
- **L'absence de structuration organisationnelle** explicite dans les architectures existantes.

Ces limitations motivent une intégration plus étroite entre méthodes d'apprentissage et modèles organisationnels, ce que nous explorerons dans les parties suivantes.

5.2.2 *Constrained MDPs, Safe RL et guidages implicites*

L'un des principaux défis du RL appliqué à des environnements critiques comme la Cyberdéfense réside dans la capacité à garantir la sûreté des politiques apprises, tout en maintenant l'adaptabilité propre à l'apprentissage connexionniste. En effet, les méthodes classiques d'optimisation de politiques visent à maximiser une récompense cumulée, sans tenir compte de contraintes additionnelles (sécurité, règles organisationnelles, équité, etc.). Pour pallier cette limite, de nombreuses extensions ont été proposées, regroupées sous la bannière du *Safe RL* et des *Constrained MDPs*.

CONSTRAINED MARKOV DECISION PROCESSES (CMDP). Un *CMDP* [189] étend le cadre du MDP classique en intégrant un ensemble de coûts C_i soumis à des bornes maximales d_i. Un *Constrained Markov Decision Process* (CMDP) est défini par un quintuplet $\langle S, A, T, R, \{C_i\}, \gamma \rangle$, où (S, A, T, R, γ) correspond au MDP standard, et où chaque $C_i : S \times A \to \mathbb{R}$ représente une fonction de coût contrainte. L'objectif devient alors :

$$\max_{\pi} \ \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right] \quad \text{s.c.} \quad \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t C_i(s_t, a_t)\right] \leqslant d_i, \ \ \forall i$$

Ce formalisme généralise naturellement les problèmes de planification où certaines propriétés (sûreté, consommation, risque) doivent être respectées en plus de la maximisation de la récompense.

SAFE RL ET OPTIMISATIONS CONTRAINTES. Dans la lignée des CMDP, plusieurs méthodes ont été développées pour résoudre de manière pratique des problèmes contraints. Parmi les plus notables :

- **CPO** [118], qui étend *Trust Region Policy Optimization* (TRPO) en assurant, via une optimisation de type Lagrangien, que les contraintes ne soient pas violées au-delà d'un seuil fixé. Cette méthode fournit des garanties partielles de sûreté pendant l'entraînement.
- ***Deep Constrained Q-Learning*** **(DCQL)** [83], qui adapte le Q-Learning profond aux environnements contraints en intégrant des multiplicateurs de Lagrange dans l'approximation de la fonction de valeur.

GUIDAGES IMPLICITES. Au-delà des CMDP, d'autres travaux proposent d'influencer indirectement le comportement des agents par des mécanismes de guidage souples, sans contraintes explicites :

- **Reward shaping** [181] : modification de la fonction de récompense afin d'inciter certains comportements (ex. : coopération, respect de protocoles).
- **Shielding** [128] : filtrage a priori ou a posteriori des actions dangereuses grâce à un modèle de sûreté, empêchant l'agent d'exécuter des comportements interdits.
- **Feedback humain** [117] : incorporation de corrections ou préférences fournies par un opérateur humain, permettant d'orienter l'apprentissage de manière interactive.

Les CMDP et le Safe RL apportent un premier niveau de garanties théoriques concernant le respect des contraintes, mais celles-ci restent généralement limitées à des contraintes locales (au niveau de l'action ou de la trajectoire) et de nature numérique. Les mécanismes

de guidage implicites offrent une flexibilité appréciable, mais ne fournissent pas de garanties formelles. Par ailleurs, ces approches ne sont pas encore adaptées au contexte multi-agent, qui nécessiterait une extension du formalisme pour intégrer les interactions entre agents. Elles constituent néanmoins des bases importantes pour envisager l'intégration de contraintes organisationnelles plus expressives, que nous chercherons à articuler ultérieurement avec des modèles symboliques tels que $\mathcal{M}$OISE$^+$.

5.2.3 *Le modèle* $\mathcal{M}$OISE$^+$

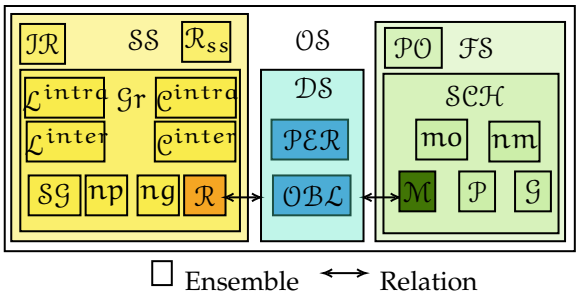


FIGURE 15 : Vue synthétique du modèle $\mathcal{M}$OISE$^+$

Le modèle $\mathcal{M}$OISE$^+$ [172] fournit une description formelle avancée d'une organisation, notamment pour la description formelle des politiques des agents (via les plans). Il prend explicitement en compte les aspects sociaux entre agents, là où AGR se concentre sur l'intégration de normes orientées conception. De plus, il propose une vision suffisamment détaillée de l'organisation pour être comprise selon différents points de vue. Une représentation visuelle des éléments formels de ce modèle est donnée en Figure 15 En nous basant sur le formalisme de $\mathcal{M}$OISE$^+$ [160], nous ne détaillons ici les éléments minimaux principaux.

Spécifications organisationnelles : Les spécifications organisationnelles comprennent l'ensemble des spécifications structurelles, fonctionnelles et déontiques $\mathcal{OS} = \langle \mathcal{SS}, \mathcal{FS}, \mathcal{DS} \rangle$, l'ensemble des spécifications organisationnelles, où $\mathcal{SS}$ sont les **spécifications structurelles**, $\mathcal{FS}$ les **spécifications fonctionnelles**, et $\mathcal{DS}$ les **spécifications déontiques**.

Spécifications Structurelles : Elles définissent la structure de l'organisation en décrivant les rôles, groupes et liens sociaux autorisés, précisant comment les agents peuvent interagir. On les note $\mathcal{SS} = \langle \mathcal{R}, \mathcal{IR}, \mathcal{G} \rangle$, où :

- $\mathcal{R}_{ss}$: l'ensemble des rôles (notés $\rho \in \mathcal{R}$) ;
- $\mathcal{IR} : \mathcal{R} \to \mathcal{R}$: la relation d'héritage entre rôles ($\mathcal{IR}(\rho_1) = \rho_2$ signifie que ρ_1 hérite de ρ_2, noté aussi $\rho_1 \sqsubset \rho_2$) ;
- RG $\subseteq$ GR : l'ensemble des groupes racines, GR $= \langle \mathcal{R}, \mathcal{SG}, \mathcal{L}^{intra}, \mathcal{L}^{inter}, \mathcal{C}^{intra}, \mathcal{C}^{inter}, np, ng \rangle$, l'ensemble des groupes, où :
  - $\mathcal{R} \subseteq \mathcal{R}_{ss}$: l'ensemble des rôles non-abstraits ;
  - $\mathcal{SG} \subseteq \mathcal{GR}$: l'ensemble des sous-groupes ;
  - $\mathcal{L} = \mathcal{R} \times \mathcal{R} \times \mathcal{TL}$: l'ensemble des liens. Un lien est un triplet $(\rho_s, \rho_d, t) \in \mathcal{L}$ (aussi noté $link(\rho_s, \rho_d, t)$), où ρ_s est le rôle source, ρ_d le rôle destination, et $t \in \mathcal{TL}, \mathcal{TL} = \{acq, com, aut\}$ le type de lien :

* $t = acq$ (acquaintance) : les agents jouant ρ_s peuvent identifier les agents jouant ρ_d ;
* $t = com$ (communication) : les agents jouant ρ_s peuvent communiquer avec ceux jouant ρ_d ;
* $t = aut$ (authority) : les agents jouant ρ_s peuvent exercer une autorité sur ceux jouant ρ_d. Ce lien nécessite les liens d'acquaintance et de communication.

– $\mathcal{L}^{intra} \subseteq \mathcal{L}$: ensemble des liens intra-groupe ;
– $\mathcal{L}^{inter} \subseteq \mathcal{L}$: ensemble des liens inter-groupe ;
– $\mathcal{C} = \mathcal{R} \times \mathcal{R}$: l'ensemble des compatibilités. Une compatibilité est un couple $(\rho_a, \rho_b) \in \mathcal{C}$ (noté aussi $\rho_a \bowtie \rho_b$), signifiant qu'un agent jouant ρ_a peut aussi jouer ρ_b ;
– $\mathcal{C}^{intra} \subseteq \mathcal{C}$: ensemble des compatibilités intra-groupe ;
– $\mathcal{C}^{inter} \subseteq \mathcal{C}$: ensemble des compatibilités inter-groupe ;
– $np : \mathcal{R} \to \mathbb{N} \times \mathbb{N}$: relation donnant la cardinalité du nombre d'agents par rôle ;
– $ng : \mathcal{SG} \to \mathbb{N} \times \mathbb{N}$: relation donnant la cardinalité de chaque sous-groupe.

Spécifications Fonctionnelles :Elles décrivent les objectifs collectifs et leur décomposition en sous-objectifs, indiquant celles qui doivent être accomplies et dans quel ordre. On les note $\mathcal{FS} = \langle \mathcal{SCH}, \mathcal{PO} \rangle$, où :

- $\mathcal{SCH} = \langle \mathcal{G}, \mathcal{M}, \mathcal{P}, mo, nm \rangle$: l'ensemble des **schémas sociaux**, où :
  – $\mathcal{G}$: l'ensemble des objectifs globaux ;
  – $\mathcal{M}$: l'ensemble des missions ;
  – $\mathcal{P} = \langle \mathcal{G}, \{\mathcal{G}\}^s, OP, [0,1] \rangle, s \in \mathbb{N}^*$: ensemble des plans qui définissent l'arbre des objectifs. Un plan $p \in \mathcal{P}$ est un 4-uplet $p = (g_f, \{g_i\}_{0 \leqslant i \leqslant s}, op, p)$, où $g_f \in \mathcal{G}$ est un objectif, les $g_i \in \mathcal{G}$ sont des sous-objectifs, $op \in OP = \{sequence, choice, parallel\}$ est un opérateur, et $p \in [0,1]$ est une probabilité de succès :
    * $op = sequence$: les g_i doivent être atteints dans un ordre précis ;
    * $op = choice$: un seul g_i doit être atteint ;
    * $op = parallel$: les g_i peuvent être atteints en parallèle ou séquentiellement.
  – $mo : \mathcal{M} \to \mathbb{P}(\mathcal{G})$: relation liant une mission à un ensemble de objectifs ;
  – $nm : \mathcal{M} \to \mathbb{N} \times \mathbb{N}$: cardinalité du nombre d'agents affectés à une mission.
- $\mathcal{PO} : \mathcal{M} \times \mathcal{M}$: ensemble des **ordres de préférence**. Un ordre de préférence est un couple (m_1, m_2) (noté aussi $m_1 \prec m_2$) signifiant que si un agent peut s'engager à la fois sur m_1 et m_2, il aura une préférence sociale pour m_1.

Spécifications Déontiques : Elles énoncent les règles normatives liant les rôles aux missions. Elles régulent qui doit, peut ou ne peut pas exécuter certaines missions. On les note $\mathcal{DS} = \langle \mathcal{OBL}, \mathcal{PER} \rangle$, l'ensemble des spécifications déontiques, où :

- $\mathcal{TC}$: ensemble des **contraintes temporelles**. Une contrainte $tc \in \mathcal{TC}$ indique les périodes pendant lesquelles une permission ou obligation est valide ($Any \in \mathcal{TC}$ signifie tout le temps) ;
- $\mathcal{OBL} : \mathcal{R} \times \mathcal{M} \times \mathcal{TC}$: ensemble des **obligations**. Une obligation est un triplet (ρ_a, m, tc) (aussi noté $obl(\rho_a, m, tc)$), signifiant qu'un agent jouant le rôle ρ_a est obligé de s'engager dans la mission m pendant la période spécifiée tc ;
- $\mathcal{PER}$: ensemble des **permissions**. Une permission est un triplet (ρ_a, m, tc) (aussi noté $per(\rho_a, m, tc)$), signifiant qu'un agent jouant le rôle ρ_a est autorisé à s'engager dans la mission m pendant tc.

Les spécifications organisationnelles appliquées aux agents sont les rôles et les objectifs (en tant que missions) à travers les permissions ou obligations. En effet, les autres spécifications structurelles comme les compatibilités ou les liens sont inhérentes aux rôles. De même, nous considérons que les objectifs, missions et leur association (mo) permettent de relier les autres spécifications fonctionnelles comme les plans, les cardinalités ou les préférences. Par conséquent, nous considérons qu'il est suffisant de prendre en compte les rôles, les missions (objectifs et correspondance) et les permissions/obligations pour décrire l'essentiel de l'organisation d'un SMA.

5.2.4 *Pourquoi le choix de* $\mathcal{M}$OISE$^+$ *?*

Le choix de $\mathcal{M}$OISE$^+$ repose sur un compromis entre expressivité, opérationalisation et traçabilité. D'autres cadres existent dans la littérature (AGR, méthodologies orientées conception, cadres fortement normatifs), mais ils couvrent moins bien simultanément les besoins de la thèse.

- **Expressivité organisationnelle** : $\mathcal{M}$OISE$^+$ combine structure (rôles/groupes), fonction (missions/plans) et déontique (permissions/obligations), ce qui permet de relier directement les contraintes de conception aux critères C4 (contrôle) et C5 (explicabilité).
- **Compatibilité avec l'optimisation sous contraintes** : les éléments organisationnels peuvent être traduits en guides de contraintes exploitables par le MARL (masquage d'actions, pondération de récompense, objectifs), ce qui soutient C2 (performance) et C3 (adaptation).
- **Traçabilité pour le cycle MAMAD** : la structure explicite facilite l'audit des décisions et le raffinement itératif dans la boucle **TRN–ANL–TRF**, au lieu d'un simple guidage implicite difficile à interpréter.

En contexte Cyberdéfense, ce choix est particulièrement pertinent car il permet de formaliser des responsabilités différenciées (détection, mitigation, supervision), d'imposer des contraintes critiques, et de conserver une base explicative exploitable lors du transfert vers des environnements opérationnels.

5.3 EXPLICABILITÉ ET EXTRACTION ORGANISATIONNELLE (ANL)

L'explicabilité constitue un enjeu central dans la conception de SMAs appris par renforcement, en particulier dans des domaines critiques comme la Cyberdéfense. Les politiques

issues du MARL sont souvent représentées par des réseaux de neurones opaques, rendant difficile leur compréhension, leur validation et leur alignement avec des spécifications organisationnelles. L'objectif de cette section est de présenter les principales notions et techniques mobilisées pour améliorer l'explicabilité et extraire des structures organisationnelles émergentes à partir des comportements appris.

5.3.1 *Notion d'explicabilité*

L'explicabilité peut être définie comme la capacité à fournir une description intelligible des décisions ou comportements d'un système d'apprentissage automatique [119]. Dans le cas du MARL, deux niveaux d'explicabilité sont généralement distingués :

- **Explicabilité locale** : comprendre les décisions d'un agent individuel à un instant donné, par exemple en reliant une action choisie à certaines observations ou caractéristiques de l'environnement.
- **Explicabilité globale** : comprendre les structures collectives qui émergent au sein du SMA, telles que la spécialisation des rôles, la coordination ou la réalisation d'objectifs collectifs.

Alors que l'explicabilité locale est bien couverte par les techniques classiques d'explicabilité en apprentissage automatique (SHAP, *Local Interpretable Model-agnostic Explanations* – LIME, attribution de gradient), l'explicabilité globale est plus récente et reste un défi ouvert [12, 60]. Elle est pourtant essentielle pour analyser, comparer et certifier les organisations implicites formées par apprentissage dans des contextes critiques.

5.3.2 *Méthodes post-hoc*

Les méthodes *post-hoc* visent à expliquer a posteriori des politiques déjà entraînées, sans modifier leur processus d'apprentissage. Elles permettent une meilleure compréhension des réseaux de neurones au prix d'une explicabilité souvent locale ou partielle.

ATTRIBUTION DE CARACTÉRISTIQUES. Des approches comme *Layer-wise Relevance Propagation* (LRP) [133] ou SHAP [122] sont utilisées pour quantifier l'importance de chaque entrée dans la prise de décision d'un agent. Elles permettent de mettre en évidence les observations critiques qui influencent une action donnée.

PATCHING ET INTERVENTIONS. Grupen et al. [56] ont proposé des techniques d'édition de modèles entraînés pour tester la sensibilité des politiques apprises à des concepts spécifiques. De même, l'utilisation d'approches dites de *causal patching* [68] permet de comprendre la relation entre représentations internes et comportements observés.

ATTRIBUTION DE CONCEPTS. Des travaux récents cherchent à relier directement des décisions à des concepts humains interprétables, par exemple via des réseaux neuronaux à base de concepts (*Concept Activation Vector* – CAV) [110] ou par attribution conceptuelle dans le MARL [52]. Ces approches ouvrent la voie à une explicabilité plus riche, en reliant apprentissage connexionniste et représentations symboliques.

5.3.3 *Inférence organisationnelle*

L'explicabilité globale suppose de dépasser l'analyse des décisions locales pour reconstruire des structures organisationnelles implicites (rôles, missions, relations de dépendance). Plusieurs directions ont été explorées dans la littérature.

CLUSTERING DE TRAJECTOIRES. L'analyse non supervisée des trajectoires d'agents permet d'identifier des rôles ou missions émergents à partir de comportements récurrents. Par exemple, Wang et al. [89] proposent l'approche ROMA, où la spécialisation comportementale des agents est favorisée par maximisation d'information mutuelle. Plus largement, les techniques de clustering appliquées aux trajectoires d'action/observation ouvrent la voie à une identification automatique des rôles collectifs.

APPROCHES BAYÉSIENNES ET INFÉRENCE DE RÔLES. Certaines méthodes formalisent l'inférence organisationnelle comme un problème probabiliste. Yusuf et Baber [146] utilisent des modèles bayésiens pour déduire les structures sociales implicites dans des environnements collaboratifs. Des travaux plus anciens comme ceux de Berenji et Vengerov [177] ont exploré l'apprentissage de rôles émergents à partir de la dynamique collective.

VERS UN LIEN AVEC LES MODÈLES SYMBOLIQUES. L'un des défis actuels consiste à relier les structures émergentes ainsi extraites à des modèles organisationnels explicites, comme $\mathcal{M}$OISE$^+$ [161]. Cette perspective ouvre la voie à une hybridation neurosymbolique, où l'on pourrait rétro-inférer des spécifications organisationnelles (rôles, missions, obligations) à partir des trajectoires observées, puis les réinjecter pour guider de nouveaux apprentissages. Des pistes récentes dans ce sens incluent les approches neurosymboliques d'explicabilité en SMAs [32].

Les méthodes actuelles permettent une certaine transparence locale et des inférences partielles sur les structures collectives, mais aucun cadre unifié ne permet encore d'extraire automatiquement des organisations complètes et de les formaliser dans des modèles symboliques. Ce verrou justifie la nécessité de contributions méthodologiques pour développer des outils d'inférence organisationnelle explicables, automatisés et exploitables dans des contextes opérationnels.

5.4 TRANSFERT SIMULATION VERS ENVIRONNEMENT RÉEL ET COHÉRENCE (TRF)

Un des défis majeurs du RL, et en particulier du MARL, réside dans l'écart entre l'environnement simulé et l'environnement réel. Cet *écart de réalité* (*reality gap*) limite la transférabilité des politiques apprises et peut conduire à des comportements inefficaces, voire dangereux, lors du déploiement opérationnel. La littérature propose plusieurs familles de méthodes pour réduire cet écart et maintenir la cohérence entre simulation et réel : (i) l'adaptation de domaine (domain adaptation et *Sim2Real*), (ii) l'apprentissage par renforcement robuste, (iii) l'adaptation en ligne, et (iv) les approches de recalibrage manuel.

5.4.1 *Domain adaptation et Sim2Real*

L'adaptation de domaine (*domain adaptation*) consiste à rapprocher les distributions de données issues de la simulation et du réel, afin de rendre les politiques apprises transférables. Deux grandes approches sont couramment employées :

- **Domain randomization** [126] : l'idée est de randomiser massivement les paramètres de la simulation (textures, latences, topologies, probabilités de transition, etc.) afin que l'agent soit entraîné sur une distribution couvrant le réel comme un cas particulier. Formellement, si $p_{\text{sim}}(s'|s, a, \theta)$ désigne la dynamique simulée avec paramètres θ, on échantillonne $\theta \sim \Theta$ pour maximiser la robustesse de la politique à la variabilité.
- **Domain invariance** [130] : ici, on apprend des représentations latentes $z = f(o)$ des observations o de manière à rendre z invariant au domaine (simulation vs réel). Cela s'écrit souvent comme une minimisation de divergence entre distributions latentes :

$$\min_{f} \; D\big(p_{\text{sim}}(z),\; p_{\text{real}}(z)\big),$$

  où D est une divergence statistique (*Kullback–Leibler divergence* – KL, *Maximum Mean Discrepancy* – MMD, adversariale).

Ces méthodes ont été utilisées avec succès en robotique [113, 126], et commencent à être adaptées en cybersécurité [74]. Elles permettent d'améliorer la fidélité et la transférabilité, mais elles n'intègrent pas toujours de mécanismes d'adaptation en ligne.

5.4.2 *Robust Reinforcement Learning*

Une seconde famille repose sur l'**apprentissage robuste** (*Robust RL*), qui vise à apprendre des politiques stables face à l'incertitude des dynamiques de l'environnement [123]. On suppose que les transitions appartiennent à un ensemble d'incertitude $\mathcal{T}$ autour du modèle nominal T. Le problème est alors formulé comme un jeu min-max :

$$\pi^* = \arg\max_{\pi} \min_{T \in \mathcal{T}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \;\middle|\; T, \pi\right]$$

L'objectif est de maximiser le retour attendu dans le pire cas de dynamique. Cette approche fournit des garanties partielles de robustesse au transfert simulation vers environnement réel, au prix d'une politique souvent plus conservatrice. Elle est utilisée notamment en robotique physique [123] et dans des contextes critiques (énergie, cyber-physique).

5.4.3 *Adaptation en ligne*

Les méthodes d'**adaptation en ligne** cherchent à mettre à jour le modèle simulé ou la politique après déploiement, à partir des retours observés. Elles s'appuient sur des techniques d'*online system identification* et de boucles de rétroaction.

- **Identification de systèmes en ligne** : l'idée est d'estimer dynamiquement les paramètres du modèle T_θ à partir des trajectoires réelles observées, via filtrage bayésien ou apprentissage incrémental [180].

- **PILCO** [148] : algorithme d'optimisation bayésienne basé sur des processus gaussiens, qui met à jour en ligne le modèle probabiliste de la dynamique pour planifier des politiques sûres.
- **World Models incrémentaux** [95] : mise à jour continue des représentations latentes au fur et à mesure des interactions, permettant de maintenir la cohérence du jumeau numérique.

Ces approches assurent une meilleure fidélité du modèle au réel, mais augmentent la complexité computationnelle et nécessitent des mécanismes de sûreté pour éviter des adaptations instables.

5.4.4 *Synchronisation manuelle*

Enfin, une approche plus pragmatique et répandue consiste à **recalibrer manuellement** la simulation à intervalles réguliers à partir de données réelles. C'est le cas de plusieurs simulateurs de Cyberdéfense tels que CybORG [74] ou CyberBattleSim [75], où les scénarios, topologies et vulnérabilités sont mis à jour manuellement par des experts. Cette approche est simple à mettre en œuvre, mais elle est peu adaptée à des environnements très dynamiques, et ne garantit pas la cohérence continue entre simulation et réalité.

5.5 BILAN

En résumé, plusieurs stratégies complémentaires existent pour réduire l'écart entre simulation et réel. Les approches *Sim2Real* (domain randomization, invariance) couvrent le transfert initial, le *Robust RL* fournit des garanties partielles en environnement incertain, l'adaptation en ligne permet de maintenir la cohérence dans la durée, et le recalibrage manuel reste la pratique dominante en cybersécurité. Cependant, aucun cadre unifié n'assure simultanément la mise à jour conjointe du modèle simulé et des politiques déployées, ce qui constitue un verrou scientifique majeur pour le transfert sûr des politiques de SMA dans des environnements critiques.

CONCLUSION

Cette deuxième partie a posé les fondations théoriques et critiques nécessaires à l'élaboration de notre méthode de conception. En s'appuyant sur les enjeux identifiés dans la Partie I, elle a permis de clarifier les concepts mobilisés, d'identifier les verrous qui justifient la nécessité d'une nouvelle approche.

Le Chapitre 5 a introduit les trois piliers conceptuels sur lesquels s'appuie notre démarche : (1) les modèles organisationnels, en particulier $\mathcal{M}$OISE$^+$, qui offrent une structuration explicite du SMA ; (2) Le MARL, qui permet une acquisition autonome de politiques dans des environnements complexes ; et (3) les *World Models*, qui fournissent un moyen de simuler un environnement à partir de données, ouvrant la voie à une exploration sécurisée et accélérée.
Le Chapitre 4 a prolongé cette analyse en examinant les limites de l'état de l'art face aux exigences soulevées par notre question. Chaque hypothèse de recherche (**H-MOD** à **H-TRF**) a été replacée dans son contexte scientifique, discutée à la lumière des travaux existants, et reliée à un verrou spécifique :

- la difficulté à représenter le problème de conception dans un cadre intégrant l'environnement réel (**H-TRF**) ;
- l'absence de *World Models* ou framework de modélisation d'un environnement de Cyberdéfense adapté au contexte multi-agent (**H-MOD**) ;
- le manque d'intégration de contraintes organisationnelles dans l'apprentissage (**H-TRN**) ;
- l'impossibilité d'analyser les comportements appris à l'échelle organisationnelle (**H-ANL**).

Ces constats convergent vers un besoin commun : celui d'une méthode unifiée, capable d'orchestrer l'ensemble du processus de conception (de la modélisation de l'environnement à l'analyse des comportements) en intégrant apprentissage et organisation dans une boucle cohérente. C'est précisément l'objectif de la méthode MAMAD, que nous introduisons dans la partie suivante.

Troisième partie

LA MÉTHODE MAMAD

INTRODUCTION

La partie précédente a mis en lumière les lacunes actuelles dans l'intégration des modèles organisationnels au sein des approches d'apprentissage multi-agent, tant du point de vue du contrôle, de l'explicabilité que de l'automatisation de la conception. Elle a également mis en lumière les lacunes dans la modélisation de l'environnement et son intégration dans le processus d'entraînement notamment sur le manque de cadre permettant d'assurer la cohérence entre l'environnement simulé et réel.

Cette troisième partie présente notre proposition pour répondre à ces lacunes : la méthode MAMAD. Cette méthode repose sur la prémisse que la conception d'un SMA peut être abordée par le prisme d'un problème d'optimisation sous contraintes. La méthode est construite autour de cette vision et s'organise autour de quatre activités :

1. **Modélisation** : modéliser l'environnement réel en un environnement simulé ainsi que les contraintes de conceptions en spécifications organisationnelles ;
2. **Apprentissage** : entraîner les agents dans cet environnement simulé en tenant compte de spécifications organisationnelles comme des rôles durant l'apprentissage ;
3. **Analyse** : extraire des spécifications structurelles et fonctionnelles émergentes à partir des trajectoires des agents entraînés ;
4. **Transfert** : mettre à jour régulièrement les politiques des agents déployés dans l'environnement réel à partir des politiques des agents entraînés en simulation et éventuellement mettre à jour ou améliorer l'environnement simulé.

Ces quatre activités peuvent être vues comme exécutées de façon itérative pour produire des SMAs adaptés à leur environnement, alignés sur des contraintes organisationnelles, explicables et robustes.

Le Chapitre 6 donne une description globale de la méthode concernant les processus proposés. Les quatre chapitres restants détaillent chacune des étapes de cette méthode : Le Chapitre 7 présente l'activité de modélisation. Le Chapitre 8 présente l'activité d'apprentissage contraint par des spécifications organisationnelles. Le Chapitre 9 présente une méthode permettant d'analyser des trajectoires pour inférer des structures organisationnelles émergentes. Le Chapitre 10 décrit l'activité de transfert. La Figure 16 illustre cette organisation de cette partie.

La méthode MAMAD ambitionne de réunir les forces des approches symboliques et connexionnistes pour une conception de SMA à la fois structurée, autonome et explicable.

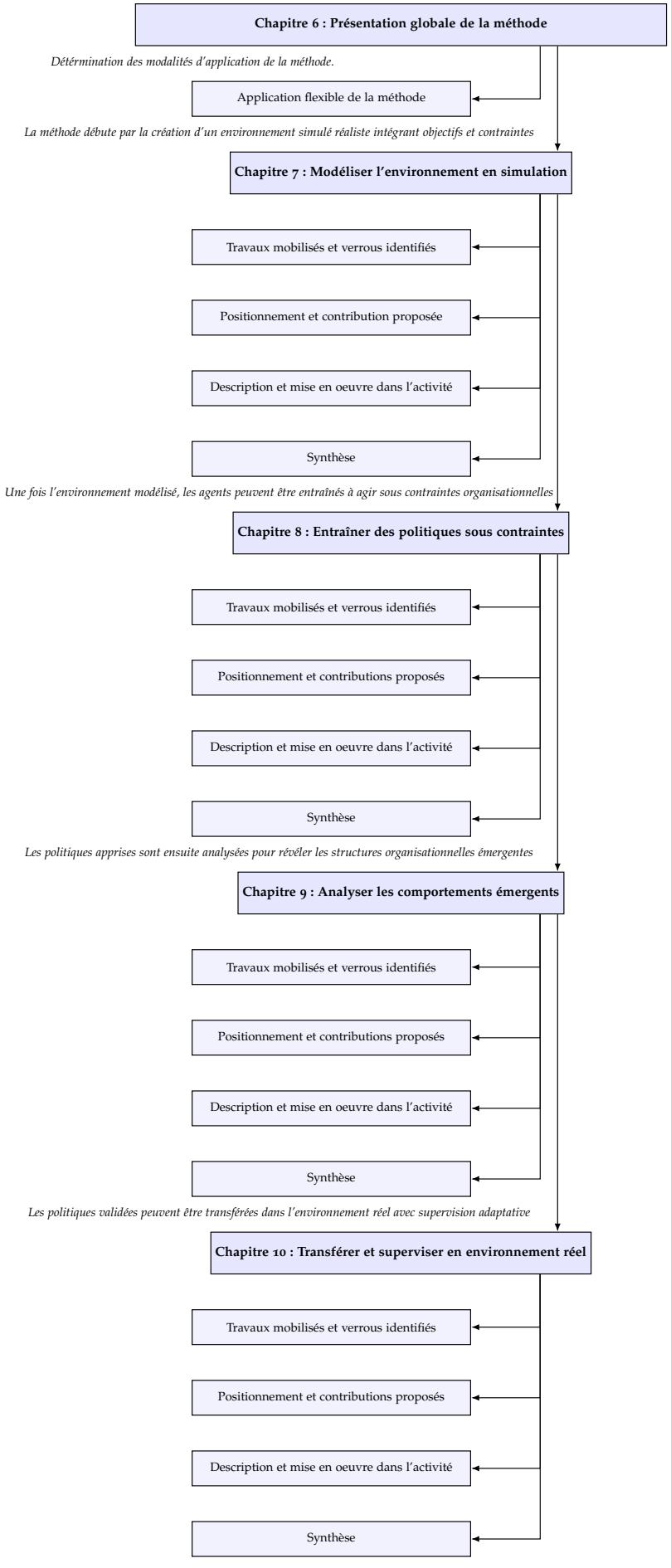


FIGURE 16 : Structure de la Partie III : La méthode MAMAD

6

PRÉSENTATION GLOBALE DE LA MÉTHODE

Guide de lecture du chapitre. Ce chapitre présente la logique globale de MAMAD avant le détail activité par activité. Pour une lecture rapide, le fil directeur est le suivant :

1. comprendre la vue d'ensemble du cycle (voir Figure 17) ;
2. lire la formalisation algorithmique complète (voir Algorithme 1) ;
3. distinguer le rôle du processus *Transferring* et du processus Modéliser-Entraîner-Analyser (MTA) ;
4. retenir les points de contrôle du raffinement ($\texttt{org_fit}_{min}$, $\bar{r}_{min}$, σ_{max}).

La méthode MAMAD [2] repose sur quatre grandes activités : (1) la modélisation de l'environnement, de l'objectif global et des contraintes organisationnelles, (2) l'apprentissage des politiques à l'aide de divers algorithmes MARL, (3) l'analyse des comportements et l'inférence des spécifications organisationnelles à l'aide d'une méthode proposée, et (4) le maintien de la cohérence entre l'environnement simulé et l'environnement réel en déployant les politiques entraînées et en mettant à jour la simulation. Cette approche guide le processus d'apprentissage des agents tout en imposant des contraintes organisationnelles strictes, garantissant ainsi l'efficacité des politiques apprises.

Le cycle de vie d'un SMA conçu avec MAMAD est illustré en Figure 17. Il commence par la modélisation de l'environnement, réalisée à partir d'un ensemble suffisant de trajectoires réelles (issues d'agents déjà transférés ou de toute autre source disponible), ainsi que la définition de l'objectif global et des contraintes de conception sous forme de rôles et d'objectifs. Ensuite, les agents sont entraînés dans cet environnement simulé à l'aide de techniques MARL. Une fois l'entraînement terminé, une analyse post-entraînement permet d'extraire les rôles et objectifs émergents des agents, ce qui conduit à l'amélioration des spécifications organisationnelles appliquées. Enfin, après validation, les politiques apprises sont déployées pour contrôler les actionneurs de l'environnement, générant ainsi de nouvelles traces qui serviront à affiner la modélisation lors des itérations suivantes.

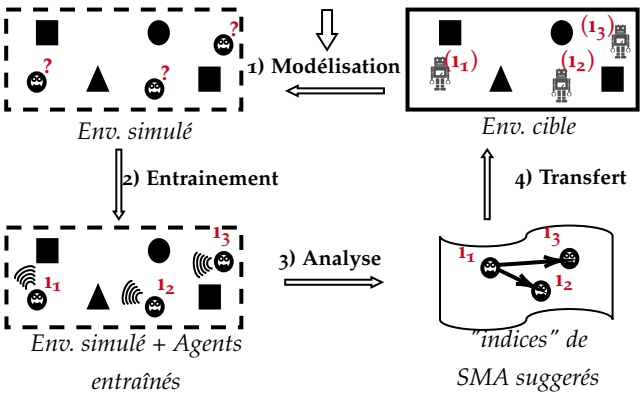


FIGURE 17 : Cycle de vie d'un SMA conçu avec MAMAD

Le cœur de la méthode MAMAD est d'envisager la conception d'un SMA comme un processus itératif d'optimisation sous contraintes. Nous proposons une description formalisée de la méthode MAMAD dans l'Algorithme 1 qui met en perspective les activités évoquées précédemment. Les données en entrée sont :

- $\mathcal{E}_0$: l'environnement initial dans lequel les agents peuvent agir ;
- $d \in D \cup OD$: un environnement fourni par l'utilisateur de façon optionnelle. Ce dernier peut avoir préalablement modélisé manuellement comme un Dec-POMDP (voir Sous-section 5.1.3) ou modélisé automatiquement comme un ODec-POMDP (voir Sous-section 8.2.5) ;
- $\mathcal{G}_{inf}$: une description informelle de l'objectif global recherché ;
- $\mathcal{C}_{inf}$: une spécification informelle des contraintes de conception ;
- $\gamma \in [0, 1]$: le facteur d'actualisation définissant une solution à court ou long terme, généralement fixé empiriquement (par défaut à 1) ;
- A, Ω : respectivement les espaces d'actions et d'observations ;
- $w_{episodes}$: la fenêtre glissante d'épisodes pour la validation de la politique conjointe (par défaut $w_{episodes}$ est fixée à 5) ;
- $\texttt{org_fit}_{min}$, $\bar{r}_{min}$, σ_{max} : respectivement la valeur minimale exigée pour le score d'**adéquation organisationnelle** obtenu sur les $w_{episodes}$ derniers épisodes, la valeur minimale de la moyenne des récompenses cumulées sur les $w_{episodes}$ derniers épisodes, et la valeur maximale de l'écart-type sur les $w_{episodes}$ derniers épisodes. Ces seuils servent à valider une politique conjointe. Ces seuils sont généralement déterminés empiriquement, à partir d'une analyse préliminaire des résultats obtenus lors d'une première série d'épisodes ;
- $\texttt{mode} \in \{\texttt{DIRECT}, \texttt{REMOTE}\}$: le mode de transfert des politiques des agents, soit direct (les agents déployés dans l'environnement embarquent les politiques pour les exécuter eux-mêmes) ou distant (la politique conjointe est exécutée sur le nœud d'exécution de *Cyberdefense Multi-Agent System Development Environment* – CybMASDE et les actions sont communiquées aux agents qui ne font que les appliquer et renvoyer leurs observations respectives à CybMASDE) ;
- n_{refine} : le nombre maximal de cycles de raffinement alternant entre entraînement (pour entraîner les politiques avec spécifications organisationnelles) et analyse (pour déterminer des spécifications organisationnelles plus pertinentes en termes de performance ou explicabilité).

L'**adéquation organisationnelle** est introduite de façon théorique comme un indicateur quantitatif que nous théorisons et qui est compris entre 0 et 1 pour évaluer dans quelle mesure les comportements des agents sont structurés et conformes à des spécifications organisationnelles (qu'elles soient implicites ou explicites). Une valeur proche de 1 indique que les agents adoptent des comportements réguliers, stables et fortement alignés avec une organisation définie. À l'inverse, une valeur proche de 0 signifie que les comportements sont très irréguliers et qu'aucun schéma organisationnel cohérent n'émerge. En résumé, l'adéquation organisationnelle évalue la conformité d'une politique conjointe à une organisation structurée et fonctionnelle. La récompense moyenne $\bar{r}$ et l'écart-type σ

sont des métriques classiques en apprentissage par renforcement, reflétant respectivement la performance globale et la stabilité d'une politique.

La méthode MAMAD propose un cadre méthodologique permettant une conception continue du SMA via la coordination itérative et asynchrone de deux processus distincts : le *processus de Transfert*, qui est connecté à l'environnement réel et gère l'exécution en temps réel et la collecte d'historiques conjoints ; et le *processus MTA*, qui consomme les données stockées pour améliorer itérativement le modèle simulé, la politique conjointe et les spécifications organisationnelles du SMA.

SCHÉMA OPÉRATOIRE (VUE SYNTHÉTIQUE). La lecture de l'Algorithme 1 peut être résumée en quatre transitions de flux :

1. **TRF** déploie π^{j}_{latest} et accumule des trajectoires réelles ;
2. un **déclencheur événementiel** active **MTA** quand le volume de données est suffisant ;
3. **MTA** enchaîne **MOD** $\rightarrow$ **TRN** $\rightarrow$ **ANL** avec mise à jour de $\mathcal{MM}$ et π^{j}_{latest} ;
4. le système revient en **TRF**, ce qui ferme la boucle sim2real.

PROCESSUS DE TRANSFERT : DÉPLOIEMENT DES POLITIQUES ET COLLECTE DE DONNÉES Ce processus qui consiste à maintenir la cohérence entre l'environnement simulé et l'environnement est actif en continu tant que le SMA est en fonctionnement dans l'environnement réel. Il a deux rôles : déployer la politique conjointe la plus récente π^{j}_{latest} auprès des agents déployés dans l'environnement cible, garantissant un comportement à jour sans interruption ; et collecter en continu les trajectoires des agents sous forme d'historiques conjoints H^{j}, stockés par lots. Une fois qu'un nombre suffisant de trajectoires est collecté, le lot est ajouté au dépôt global $\mathcal{D}_{H^{j}}$. Si le processus de mise à jour n'est pas en cours, il déclenche le lancement du processus MTA.

PROCESSUS MTA : OPTIMISATION DES POLITIQUES ET RAFFINEMENT ORGANISATIONNEL Ce processus modélise le problème de conception actuel et améliore la politique conjointe des agents ainsi que ses spécifications organisationnelles. Il commence par construire un modèle de prédiction d'observations conjointes (*Joint-Observation Prediction Model* – JOPM) T^{j} à l'aide de *World Models* [14] étendus, à partir des trajectoires collectées. Les exigences de conception sont formalisées sous forme de spécifications organisationnelles MOISE+MARL $\mathcal{MM}$ et l'objectif est formalisé par une fonction de récompense basée sur l'historique R^{j}_{H}.

Un modèle markovien est ensuite construit à partir des éléments modélisés afin d'entraîner les agents en tenant compte des spécifications organisationnelles, via le framework MOISE+MARL. Une fois l'entraînement terminé, la politique conjointe π^{j} est analysée à l'aide de *Trajectory-based Evaluation in MOISE+MARL* (TEMM) afin d'inférer les spécifications organisationnelles implicites $\mathcal{MM}_{\text{imp}}$ et de calculer un score d'adéquation organisationnelle.

BOUCLE DE RAFFINEMENT VIA LES SPÉCIFICATIONS ORGANISATIONNELLES Si la politique apprise montre des performances insuffisantes ou une grande variabilité (par rapport aux seuils définis), les spécifications organisationnelles implicites inférées sont utilisées pour raffiner les spécifications initiales. Ce processus de raffinement peut impliquer une inspection manuelle des structures inférées pour identifier les facteurs clés

Algorithme 1 : Vue algorithmique de la méthode MAMAD

Input : Environnement initial $\mathcal{E}$, Environnement simulé fourni $d \in OD \cup D$, objectif informel $\mathcal{G}_{inf}$, contraintes de conception informelles $\mathcal{C}_{inf}$, facteur d'actualisation γ, espace d'observation Ω, espace d'action A, fenêtre glissante d'épisodes de validation $w_{episodes}$, valeur minimale d'adéquation organisationnelle $\texttt{org_fit}_{min}$, valeur minimale de récompense cumulée moyenne sur un épisode $\bar{r}_{min}$, valeur maximale de l'écart-type des récompenses σ_{max}, mode de transfert $\texttt{mode} \in \{\texttt{DIRECT}, \texttt{REMOTE}\}$, nombre maximal de cycles de raffinement n_{refine}

Output : Un SMA déployé satisfaisant aux exigences de conception, de performance et d'explicabilité ; ainsi que ses spécifications organisationnelles associées

Initialiser : $\mathcal{D}_{H^j} \leftarrow \emptyset$, $\pi^j_{latest} \leftarrow \pi^j_{init}$, $\texttt{running_MTA} \leftarrow \texttt{False}$, $\texttt{running_refinement} \leftarrow \texttt{True}$, $\mathcal{MM} \leftarrow \emptyset$

while *SMA en cours de conception* **do**

  `// Transfert : récupération des trajectoires & déploiement de la politique`

  $(\mathcal{D}_{H^j}, \texttt{need_update}) \leftarrow \texttt{transfer}(\pi^j_{latest}, \mathcal{E}, \mathcal{D}_{H^j}, \texttt{mode})$ `// appel asynchrone`

  if *need_update et non running_MTA* **then**

    `launch_MTA()` `// appel asynchrone`

Processus (MTA)

$\texttt{running_MTA} \leftarrow \texttt{True}$ `// Variable globale`

`// Si env. simulé non fourni, lancer modélisation`

if $d = \emptyset$ **then**

  `// Modélisation : modéliser l'environnement réel`

  $d := (\mathcal{T}^j, \Omega_0^{\mathcal{T}^j}, R^j_H, S^j_H, \texttt{Render}^j_H) \leftarrow \texttt{model}(\mathcal{E}, \mathcal{D}_{H^j}, \mathcal{G}_{inf}, \gamma, \Omega, A)$

`// Politique insatisfaisante ou nombre max. de raffinement non atteint`

while $i < n_{refine}$ *ou* ($\texttt{org_fit} < \texttt{org_fit}_{min}$ ou $\bar{r} < \bar{r}_{min}$ ou $\sigma > \sigma_{max}$) ou *running_refinement* **do**

  `// Entraînement : politique sous spec. org.`

  $\pi^j \leftarrow \texttt{train}(d, \mathcal{MM}, \mathcal{C}_{inf}, \gamma, \Omega, A)$

  `// Analyse : inférer les nouvelles spec. org.`

  $(\mathcal{MM}_{inferred}, \texttt{org_fit}, \bar{r}, \sigma, \texttt{running_refinement}) \leftarrow \texttt{analyze}(d, \pi^j, \mathcal{MM}, w_{episodes})$

  $\pi^j_{latest} \leftarrow \pi^j$ `// Mise à jour de la politique la plus récente`

  $\mathcal{MM} \leftarrow \mathcal{MM}_{inferred}$

  $i \leftarrow i + 1$

$\texttt{running_MTA} \leftarrow \texttt{False}$ `// Variable globale`

de succès des comportements émergents. Guidé par ces observations, le concepteur peut réviser les spécifications afin d'orienter les prochaines itérations d'apprentissage.

Cette boucle est répétée jusqu'à un maximum de n_{refine} fois, orientant progressivement l'espace des politiques vers des comportements plus structurés et plus performants. La dernière politique validée est alors enregistrée comme π^j_{latest}, prête à être déployée dans l'environnement réel.

La boucle de raffinement est particulièrement utile dans les environnements complexes où la connaissance préalable est limitée ou où la conception manuelle serait trop coûteuse. À chaque itération, elle permet de restreindre l'espace de recherche des politiques en le concentrant sur les régions associées à des régularités organisationnelles émergentes.

A noter que ce processus peut commencer sans aucune spécification organisationnelle initiale, et produire par raffinement successif des contraintes organisationnelles pertinentes, objectives, et indépendantes de toute expertise humaine ou connaissance préalable de l'environnement.

L'interaction entre ces deux processus asynchrones constitue un cycle de conception de SMA complet et fermé. Le système apprend continuellement à partir de l'exécution réelle, met à jour son modèle simulé, réentraîne sous des spécifications évolutives, et déploie des politiques améliorées sans nécessiter d'intervention constante du concepteur. Cette architecture établit un pont entre les principes symboliques de l'ingénierie orientée agents et l'automatisation par apprentissage, assurant conformité, adaptabilité et explicabilité au niveau organisationnel.

Il est à noter que nous proposons d'exploiter un environnement simulé modélisé comme un jumeau numérique (*Digital Twin*) pour l'entraînement ultérieur, tandis que les approches MBRL combinent simultanément modélisation et apprentissage. En effet, nous privilégions une séparation entre modélisation et apprentissage pour les raisons suivantes : i) la réutilisabilité du modèle d'environnement pour d'autres entraînements d'agents, avec des ajustements éventuels ; ii) le besoin d'agents simples n'embarquant pas de modèles coûteux pour planifier ; iii) le besoin d'un environnement simulé de haute fidélité commun à tous les agents.

6.1 APPLICATION FLEXIBLE DE LA MÉTHODE MAMAD

TABLE 13 : Taxonomie de la méthode MAMAD avec activités, sous-activités et acronymes

Activité	Sous-activité	Description	Entrées requises	Sorties produites
MOD	MOD-MAN	Création manuelle du modèle simulé via un framework markovien générique (Dec-POMDP étendu avec MOISE+MARL).	Description informelle de l'environnement, des objectifs et contraintes.	Modèle formel exploitable par MARL.
	MOD-AUT	Génération automatique du modèle simulé à partir de traces collectées (World Models, VAE+RNN, LSTM, etc.).	Traces (actions, observations) collectées dans l'environnement réel.	Modèle simulé approximatif (fonction de transition et d'observation).
TRN	TRN-CON	Apprentissage multi-agent guidé par spécifications organisationnelles MOISE+MARL (rôles, missions, contraintes).	Modèle simulé + spécifications MOISE+MARL.	Politiques conjointes respectant contraintes et objectifs organisationnels.
	TRN-UNC	Apprentissage multi-agent sans contraintes organisationnelles (MOISE+MARL non utilisé ou vide).	Modèle simulé.	Politiques conjointes optimisées uniquement selon la récompense.
ANL	ANL-MAN	Analyse assistée par TEMM suivie d'un ajustement manuel.	Politiques + données de trajectoires.	Rôles et objectifs implicites affinés par l'utilisateur.
	ANL-AUT	Analyse automatique complète via Auto-TEMM (rôles implicites, objectifs intermédiaires, évaluation SOF/FOF).	Politiques + données de trajectoires.	Rapport d'analyse automatisé.
TRF	TRF-MAN	Transfert manuel des politiques dans l'environnement réel et mise à jour manuelle du modèle simulé.	Politiques apprises.	Politiques déployées et environnement ajusté si nécessaire.
	TRF-AUT	Transfert et synchronisation automatique (cadre logiciel automatisant le déploiement et l'actualisation du modèle simulé).	Politiques apprises + framework de déploiement.	Politiques déployées et modèle simulé mis à jour automatiquement.

La méthode MAMAD a été conçue pour être modulable et adaptable selon les besoins de chaque cas d'application. En pratique, toutes les activités décrites dans la taxonomie (Table 13) ne sont pas nécessairement appliquées dans leur intégralité. Chaque activité

peut être utilisée de façon indépendante ou combinée avec d'autres, et chacune dispose de plusieurs sous-activités offrant différents niveaux d'automatisation et de contraintes.

Cette flexibilité permet :

- **Une application partielle** : un cas d'application peut exploiter uniquement certaines activités (par exemple, **MOD** et **TRN** uniquement) tout en omettant l'analyse et le transfert si ces étapes ne sont pas nécessaires comme dans le cas où l'environnement n'est pas dynamique.
- **Un choix ciblé de sous-activités** : pour chaque activité retenue, une sous-activité peut être choisie en fonction des objectifs, des ressources disponibles et du degré d'automatisation souhaité (par exemple, MOD-AUT pour la modélisation automatisée, TRN-CON pour l'entraînement avec des contraintes). Cela permet de gérer le coût de conception en fonction du niveau de complexité de l'environnement et des ressources (financières, temps, expérience) disponibles.
- **Une combinaison adaptative** : certaines activités peuvent être réalisées de manière automatisée tandis que d'autres restent manuelles ou semi-manuelles, afin d'équilibrer précision, contrôle et rapidité.

EXEMPLE ABSTRAIT Considérons un scénario où l'on souhaite concevoir rapidement un SMA pour un environnement complexe, avec un budget limité en ressources humaines, mais un accès étendu à des données d'exécution. Dans ce cas, on pourrait adopter la configuration suivante :

- MOD-AUT : utilisation d'un modèle automatisé basé sur des traces collectées (*World Models*) afin de gagner du temps dans la construction du jumeau numérique.
- TRN-CON : entraînement multi-agent guidé par des spécifications organisationnelles MOISE+MARL pour garantir la conformité des comportements aux rôles et objectifs définis.
- ANL-AUT : analyse entièrement automatisée via *Automatic Trajectory-based Evaluation in MOISE+MARL* (Auto-TEMM) pour extraire rôles et objectifs implicites, et évaluer l'adéquation organisationnelle.
- Pas d'activité **TRF** : les politiques apprises sont utilisées uniquement dans l'environnement simulé pour des études exploratoires, sans déploiement réel.

Cette configuration peut être notée succinctement grâce aux acronymes de la taxonomie :

$$\text{Configuration} = \{\text{MOD-AUT},\ \text{TRN-CON},\ \text{ANL-AUT}\}$$

Ce formalisme facilite la documentation des choix méthodologiques pour chaque expérimentation et permet de comparer rapidement différents cas d'application. Ainsi, la taxonomie proposée constitue un outil de référence pour spécifier précisément le *chemin méthodologique* suivi dans une étude, tout en mettant en évidence les choix d'automatisation et de guidage organisationnel effectués. Les chapitres suivants détaillent chaque activité du cadre MAMAD, identifient les défis spécifiques rencontrés, et décrivent les contributions proposées pour y répondre.

6.2 BILAN

La méthode MAMAD offre un cadre méthodologique structuré pour la conception de SMA, en combinant modélisation, apprentissage, analyse et transfert dans un cycle itératif. Elle permet d'intégrer explicitement des spécifications organisationnelles dans le processus d'apprentissage, afin de contraindre les agents à adopter des comportements conformes aux exigences (par exemple, garantir la sécurité ou restreindre l'espace de recherche des politiques). MAMAD facilite également la découverte, de manière semi-automatisée, de spécifications organisationnelles adaptées à l'atteinte des objectifs du SMA. L'alternance entre entraînement et analyse permet de cibler progressivement les zones les plus pertinentes de l'espace des politiques, réduisant ainsi le besoin de connaissances ou d'expertise préalable sur les interactions agents-environnement.

7

MODÉLISER L'ENVIRONNEMENT EN SIMULATION

L'*activité de modélisation* occupe une place centrale dans la méthode MAMAD. Elle consiste à représenter le problème de conception comme un problème d'optimisation sous contraintes, en produisant une abstraction fidèle de l'environnement dans lequel évolueront les agents. Cette activité joue le rôle de socle pour l'ensemble de la méthode : sans un modèle cohérent et suffisamment riche, les étapes suivantes (entraînement, analyse, transfert) ne peuvent pas être réalisées de manière robuste.

En pratique, la modélisation doit fournir un *jumeau numérique* de l'environnement réel, c'est-à-dire une simulation dans laquelle les agents peuvent interagir, recevoir des observations, exécuter des actions, et accumuler des récompenses en fonction d'objectifs donnés. Ce modèle sert ainsi à la fois de banc d'essai pour optimiser les politiques d'agents et de support formel pour raisonner sur la validité des comportements obtenus.

Objectifs formels

L'objectif de cette activité est de transformer les informations disponibles (traces d'interactions passées, objectifs et contraintes formulés de manière informelle, description partielle de l'environnement) en un formalisme standardisé utilisable par la suite de la méthode.

Concrètement, les **entrées** de cette activité sont :

- la description de l'environnement $\mathcal{E}$;
- les historiques conjoints d'interactions $\mathcal{D}_{H^j}$;
- l'objectif global informel $\mathcal{G}_{\text{inf}}$;
- le facteur d'actualisation γ ;
- l'espace des observations Ω ;
- l'espace des actions A.

Les **sorties attendues** sont :

- un modèle de l'environnement simulé d formalisé soit comme un Dec-POMDP (modélisation manuelle) soit comme un ODec-POMDP (modélisation automatisée) ;

La relation globale peut être exprimée par :

$$\texttt{model} : \mathcal{E} \times \mathcal{D}_{H^j} \times \mathcal{G}_{\text{inf}} \times \gamma \times \Omega \times A \to D \cup OD$$

où d est le modèle de l'environnement simulé. Dans le cas d'une modélisation manuelle, le modèle d est formalisé comme un Dec-POMDP : $d = (S, \{A_i\}, T, R, \{\Omega_i\}, O, \gamma)$. Dans le cas d'une modélisation automatisée, le modèle d est formalisé comme un ODec-POMDP : $d = (\mathcal{T}^j, \Omega_0^{\mathcal{T}^j}, R_H^j, S_H^j, \texttt{Render}_H^j)$ avec :

- un JOPM $\mathcal{T}^j$;
- un ensemble d'états initiaux $\Omega_0^{\mathcal{T}^j}$;

- une fonction de récompense basée sur l'historique R_H^j ;
- une fonction d'arrêt S_H^j déterminant les conditions de terminaison de l'épisode ;
- une fonction de rendue optionnelle $Render_H^j$ permettant de visualiser les trajectoires.

7.1 TRAVAUX MOBILISÉS ET VERROUS IDENTIFIÉS

La modélisation d'un environnement multi-agent pour la Cyberdéfense repose sur plusieurs piliers théoriques. D'une part, le formalisme des **Dec-POMDP** fournit un cadre mathématique rigoureux pour décrire les environnements multi-agents stochastiques et partiellement observables. Dans ce cadre, l'état global du système est caché, les agents ne disposent que d'observations partielles, et doivent prendre des décisions coordonnées pour maximiser une récompense commune.

D'autre part, les **World Models** constituent une approche connexionniste visant à apprendre un simulateur d'environnement à partir de données historiques. Un World Model combine généralement des autoencodeurs pour compresser les observations, un modèle de dynamique récurrent pour prédire l'évolution des représentations latentes, et un décodeur pour reconstruire les observations futures. Ces modèles permettent de générer un environnement simulé de haute fidélité sans nécessiter une description complète a priori.

Enfin, plusieurs **travaux de simulation multi-agents** (ex. environnements PettingZoo, frameworks pour la robotique collective, simulateurs de réseaux) montrent l'importance de disposer d'outils capables de représenter les interactions complexes entre agents. Ces environnements, bien que puissants, sont souvent spécifiques à un domaine et difficilement généralisables à la Cyberdéfense.

En résumé, les approches symboliques (basées sur des modèles explicites comme Dec-POMDP) apportent de l'explicabilité et du contrôle, tandis que les approches connexionnistes (basées sur les World Models) favorisent l'adaptation et la performance. La modélisation doit donc chercher à articuler ces deux dimensions.

Malgré ces apports, plusieurs verrous scientifiques et techniques demeurent :

- **Absence de modèle générique** : il n'existe pas de châssis unifié permettant d'homogénéiser la modélisation des environnements multi-agents de Cyberdéfense. Chaque modèle est souvent ad hoc et difficilement réutilisable.
- **Limites des World Models existants** : les architectures actuelles sont principalement conçues pour des contextes mono-agent. Leur extension directe aux environnements multi-agents se heurte à la croissance combinatoire des observations conjointes et des actions.

Ces verrous motivent le développement de nouvelles contributions, combinant l'élaboration d'un modèle générique pour la simulation manuelle et l'extension des World Models au contexte multi-agent.

7.2 POSITIONNEMENT ET CONTRIBUTIONS PROPOSÉES

Les approches existantes de modélisation se répartissent en deux grandes catégories. D'un côté, les approches **manuelles**, qui consistent à construire un modèle formel de l'environnement à partir d'une description experte (ex. modèles Dec-POMDP adaptés à un cas

particulier de Cyberdéfense). Ces approches présentent l'avantage d'être explicables et contrôlables, mais elles sont chronophages, nécessitent une expertise approfondie du domaine, et conduisent souvent à des modèles hétérogènes difficilement réutilisables ou comparables.

De l'autre côté, les approches **automatisées**, telles que les World Models, qui permettent d'apprendre directement un simulateur à partir de traces d'interactions passées. Ces méthodes offrent une grande capacité d'adaptation et permettent de capturer des dynamiques complexes. Néanmoins, elles souffrent d'un manque d'explicabilité, et leurs extensions au cadre multi-agents restent limitées par la dimensionnalité des observations et la coordination des agents.

Dans le contexte de la Cyberdéfense, ni l'une ni l'autre de ces approches n'est suffisante. La conception d'une méthode générique impose de combiner les avantages des deux :

- proposer un **modèle générique formel** qui serve de châssis commun pour homogénéiser la modélisation manuelle des environnements multi-agents ;
- développer une **extension multi-agent des World Models**, afin d'automatiser la génération de simulations tout en capturant les interactions entre agents.

Ce double positionnement permet de tirer parti à la fois de l'explicabilité et de la réutilisabilité offerts par les modèles formels, et de la capacité d'adaptation offerte par les approches connexionnistes.

7.2.1 *Spécificité Cyberdéfense : éléments génériques vs spécialisés*

Un point important de positionnement est de distinguer clairement ce qui relève d'un cadre **générique** de conception de SMAs et ce qui constitue une **spécialisation Cyberdéfense**.

- **Composants génériques** : la structuration **MOD–TRN–ANL–TRF**, le formalisme Dec-POMDP/ODec-POMDP, la boucle de raffinement, et les mécanismes d'apprentissage/évaluation sont réutilisables dans d'autres domaines multi-agents.
- **Composants spécialisés Cyberdéfense** : le modèle pré-spécialisé orienté réseau (nœuds, propriétés, privilèges), l'intégration des scénarios attaque/défense via MITRE ATT&CK et arbres AD, les métriques de sécurité (détection, violation, robustesse) et les environnements cibles (Company Infrastructure, Kubernetes, Drone Swarm).
- **Impact sur les critères** : la spécialisation sert principalement C2/C3 (performance/adaptation en contexte opérationnel) et C4/C5 (contrôle/explicabilité) en rendant explicites les contraintes de sécurité, tandis que la structure générique de la méthode soutient C1 (autonomie) et la transférabilité globale.

Ainsi, la méthode n'est pas limitée à la Cyberdéfense, mais sa déclinaison présentée dans ce manuscrit intègre explicitement les artefacts du domaine pour assurer un alignement opérationnel réel.

7.2.2 *Les World Models Multi-Agents pour la génération automatique du modèle simulé*

Dans cette approche automatisée, on commence par générer un environnement simulé de haute fidélité en construisant un JOPM $\mathcal{T}^j : H^j \times \Omega^j \times A^j \to \mathcal{H} \times \hat{\Omega}^j$ à partir des

traces d'interactions réelles $\mathcal{D}_{H^j}$ (avec $h^j \in \mathcal{D}_{H^j}$, $h^j = (h^1, h^2 \dots h^{|\mathcal{A}|})$ et pour $i \in \{0 \dots |\mathcal{A}|\}$, $h^i = \langle (o_t, a_t) \rangle_{t \in [\![0, n_{step}]\!]}$). À un pas de temps $t \in [\![0, n_{step}]\!]$, pour un état caché récurrent $\tilde{h}_{t-1} \in \mathcal{H}$ représentant l'historique conjoint jusqu'à $t-1$ (avec $\tilde{h}_{-1} = \mathbf{0}$), l'observation conjointe reçue $o_t^j \in H^j$ et l'action conjointe $a_t^j \in A^j$, le JOPM $\mathcal{T}^j$ renvoie le nouvel état caché $\tilde{h}_t \in \mathcal{H}$ ainsi que la prédiction de la prochaine observation conjointe $\hat{o}^j \in \hat{\Omega}^j$. Cette architecture, illustrée en Figure 18, permet à MAMAD d'apprendre la dynamique des observations de l'environnement pour former une simulation depuis zéro.

Dans les environnements multi-agents, les observations conjointes deviennent rapidement de grande dimension à mesure que le nombre d'agents augmente. Pour pallier cela, des fonctions d'encodage conjoint sont introduites pour les observations et les actions.

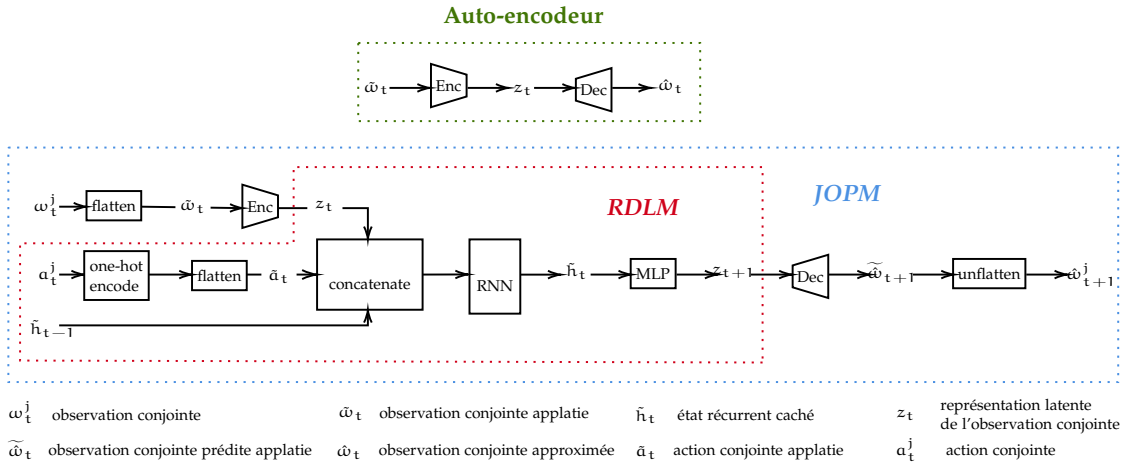


FIGURE 18 : Schéma de l'architecture d'un JOPM incluant le RLDM et l'Auto-encodeur

Plus précisément, chaque observation conjointe $o_t^j = (o_t^1, \dots, o_t^{|\mathcal{A}|}) \in \Omega^j$ est aplatie en un seul vecteur $\tilde{o}_t = vec(o_t^1, \dots, o_t^{|\mathcal{A}|})$, puis passé au travers d'un encodeur $Enc : \tilde{\Omega} \to Z$ pour produire une représentation latente $z_t = Enc(\tilde{o}_t)$. Un décodeur $Dec : Z \to \widetilde{\hat{\Omega}}$ permet la reconstruction de l'observation conjointe applatie $\widetilde{\hat{o}}_t = Dec(z_t)$ avant d'être recomposé en une observation conjointe approximée $\hat{o}_t^j = unvec(\widetilde{\hat{o}}_t) = (\hat{o}_t^1, \dots, \hat{o}_t^{|\mathcal{A}|}) \in \Omega^j$.

On utilise généralement des MLPs ou des architectures à base d'attention pour l'auto-encodeur (ensemble encodeur-décodeur), afin d'agréger les informations multi-agents en vecteurs de caractéristiques de taille fixe, tout en capturant les dépendances critiques entre agents.

Une fois l'encodage effectué sur l'ensemble des observations conjointes $o^j \in \Omega^j$ pour obtenir les représentations latentes $z_t \in Z_t$, le *World Model* multi-agent fonctionne comme en contexte mono-agent, en utilisant les observations encodées z_t dans les historiques transmis au RLDM $\mathcal{T}^z$ on peut également utiliser un RLDM. Pour chacun des épisodes collectés $h^j \in \mathcal{D}_{H^j}$, pour chaque étape $t \in [\![0, n_{step}]\!]$, la représentation latente de l'observation conjointe z_t est concaténée avec le vecteur d'état caché récurrent $\tilde{h}_{t-1}$ ainsi que le vecteur de l'action conjointe aplatie $\tilde{a}_t$. Le vecteur résultant de cette concaténation est passé au travers du RNN (LSTM) pour mettre à jour le vecteur d'état caché récurrent $\tilde{h}_t$ puis est passé au travers un MLP pour déterminer la représentation latente de l'observation conjointe approximée à l'instant suivant z_{t+1}. Ensuite, le décodeur Dec, permet d'obtenir la reconstruction de l'observation conjointe approximée aplatie $\tilde{\hat{o}}_{t+1}^j$. Enfin, cette observation conjointe approximée aplatie est recomposée en une observation conjointe prédite pour l'état suivant $\hat{o}_{t+1}^j$. Dans le cadre de MAMAD, les *World Models* constituent un des

cœurs de la simulation mise en œuvre par l'activité de modélisation, agissant comme des jumeaux numériques de haute fidélité de l'environnement cible.

7.2.3 *Un modèle de simulation pour la génération manuelle du modèle simulé*

Dans l'approche manuelle, une description informelle de l'environnement, de l'objectif global et des contraintes est traduite en un modèle formel générique, extensible à divers contextes [48].

7.2.3.1 *Modélisation générale Dec-POMDP pré-spécialisé*

Nous considérons un environnement réseau composé de nœuds, chacun décrit par un ensemble de propriétés (id, v). Des agents (cyberattaquants/défenseurs) interagissent avec ces nœuds via des actions conditionnées par leurs observations et privilèges. Une action modifie l'état de l'environnement si ses préconditions (sur les propriétés) sont satisfaites, produisant un nouvel état et de nouvelles observations pour l'agent. Les agents agissent séquentiellement (mode *Agent Environment Cycle* – AEC), chaque tour consistant à choisir une action, appliquer la transition, et recevoir l'observation/récompense.

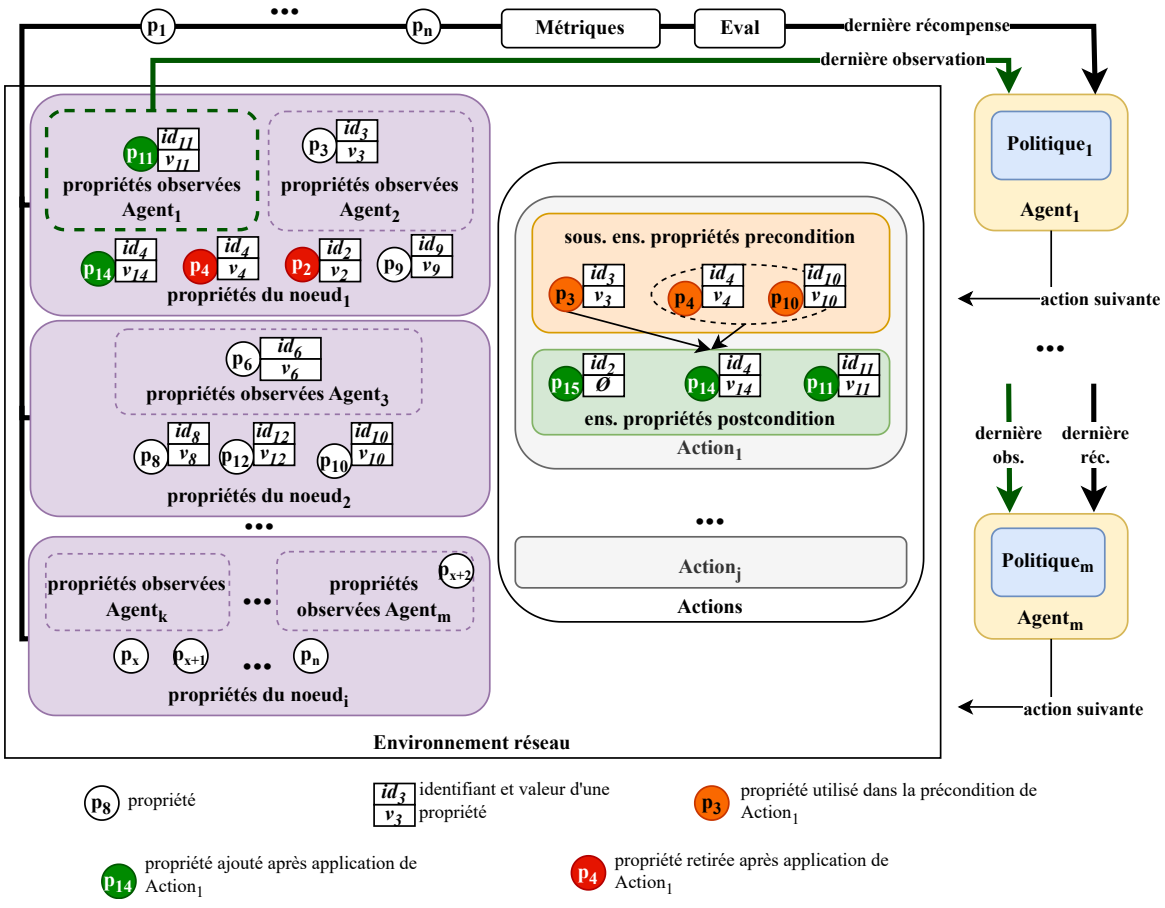


FIGURE 19 : Vue illustrative du modèle de simulation

Le modèle Dec-POMDP exprime l'état comme l'ensemble des propriétés des nœuds. Les actions sont définies par des pré/post-conditions sur ces propriétés. Les transitions et observations sont conditionnées par ces propriétés, et les récompenses sont calculées à partir de métriques sur l'état.

7.2.3.2 *Modélisation formelle Dec-POMDP*

Nous définissons les éléments liés aux propriétés des nœuds, des agents et des actions de l'environnement suivant :

- $Ag = \{ag_1, .., ag_{|Ag|}\}$: ensemble des agents (cyberattaquants et Cyberdéfenseurs).
- Nous appelons le couple $p = (id_j, v_j)$ avec $id_j \in ID$ et $v_j \in V$, une propriété.
  - *Intrusion Detection*(ID) : l'ensemble des identifiants de propriétés indiquant éventuellement comment les propriétés sont organisées dans une structure de données non plate (telle que PC1.processes.agents.agent1). Ces identifiants de propriétés peuvent être utilisés pour un chemin d'accès à un fichier, le type de système d'exploitation utilisé dans un nœud, une ligne de commande utilisée par un agent...
  - V : Ensemble des valeurs de propriétés. Celles-ci peuvent inclure le contenu d'un fichier, une description complète du système d'exploitation, le résultat d'une ligne de commande...
- $P_j = \{p_1, .., p_{|P_j|}\}$: l'ensemble des propriétés p_l (avec $l \in \{1, .., |P_j|\}$) du nœud j ($j \in \mathbb{N}$). Par exemple, ces propriétés peuvent inclure certains identifiants de processus en cours d'exécution, la liste des fichiers d'un dossier, le type de système d'exploitation avec une description, des connaissances spécifiques d'un agent, etc.
  - $P = P_1 \cup P_2 .. \cup P_{|P|}$: Ensemble de toutes les propriétés des nœuds.
- $Obs : \mathcal{P}(P) \times Ag \rightarrow \mathcal{P}(P_{Ag}), P_{Ag} \subset P$: Relation qui associe les propriétés des nœuds et un agent au sous-ensemble de propriétés observées par l'agent.
- $Action : P_{pre} \rightarrow P_{post}$: Relation qui associe un sous-ensemble de propriétés implicite dans une pré-condition booléenne conjonctive équivalente ($P_{pre} \subset \mathcal{P}(P)$) à un sous-ensemble de toutes les propriétés de la post-condition ($P_{post} \in \mathcal{P}(P)$). Par exemple, les propriétés $p_1 = (agent_X_privilege_level, root)$, $p_2 = (agent_X_accessed_text_editor, Vim)$ et $p_3 = (agent_X_bashrc_known_filepath, /home/user/.bashrc)$ peuvent former une pré-condition ($p_1 \wedge p_2 \wedge p_3$) pour associer un nouvel ensemble de propriétés contenant $p4 = (bashrc_file_modified_by_X_agent, \top)$. Deux sous-ensembles de pré-conditions peuvent être associés au même sous-ensemble de post-conditions pour modéliser une disjonction booléenne.
- $Metrics : \mathcal{P}(P) \times A \rightarrow \mathbb{R}^n$: donne des métriques associées à un ensemble de propriétés et à une action conjointe. Par exemple, le nombre de nœuds encore actifs, les mouvements latéraux, etc.

En utilisant la description formelle d'un Dec-POMDP [132], nous proposons le modèle Dec-POMDP pré-spécialisé suivant :

- $S = \{s_1, ..s_{|S|}\}, s_i \subseteq P$ et $1 \leqslant i \leqslant |S|$: L'espace des états en tant qu'ensembles de propriétés possibles.
- $A_i = \{a_i^1, .., a_i^{|A_i|}\}, a_i^j \in Action$ et $1 \leqslant j \leqslant |A_i|$: l'ensemble des actions possibles pour l'agent i.
- T : Ensemble des probabilités de transition conditionnelles entre les états

  - Avec $T(s, a, s') = \mathbb{P}(s'|s, a)$, la relation qui associe la probabilité d'aller de l'état $s \in S$ à l'état $s' \in S$ sachant que nous avons joué $a = (P^a_{pre} \times P^a_{post}) \in A$ avec $P^a_{pre} \subset \mathcal{P}(P)$ et $P^a_{post} \in \mathcal{P}(P)$
  - Avec $\mathbb{P}(s'|s, a) = 0$ si s ne satisfait pas la condition préalable de a (c'est-à-dire $\exists\, P^a_{pre_s} \in P^a_{pre} \,|\, P^a_{pre_s} \notin \mathcal{P}(s)$).
  - Avec $s' = (s - \{p_l = (id_l, v_l) \,|\, p_l \in s \text{ et } id_l \in \{id_k \,|\, (id_k, v_k) \in P^a_{post} \text{ et } v_k \neq \varnothing\}\}) \cup P^a_{post}$

- $R : S \times A \to \mathbb{R}^2 = Eval \circ Metrics$: La fonction de récompense qui prend un état et une action et associe un indicateur de performance (à l'aide des métriques de l'état) pour les attaquants et les défenseurs.
  - Avec $Eval : \mathbb{R}^n \to \mathbb{R}^2$, associe un vecteur métrique à une récompense pour les cyberattaquants et les Cyberdéfenseurs.

- $\Omega_i \subset Range(Obs \,|\, \{(s, ag_i)|s \in S \text{ et } ag_i \in Ag\}) \subset P$: ensemble des propriétés observables pour l'agent ag_i. Par exemple, le contenu d'un fichier, la sortie du journal d'une commande, le résultat d'un scan de port, etc.
  - $\Omega = \Omega_1 \cup \Omega_2 .. \cup \Omega_{|Ag|} = Range(Obs)$: Ensemble de toutes les propriétés observables pour tous les agents.

- O : Ensemble des probabilités d'observation conditionnelles.
  - Avec $O(s', a, o) = \mathbb{P}(o|s', a)$, la relation qui associe la probabilité d'observer une observation $o \subset \Omega$ à partir de l'état $s' \in S$ induit par $a \in A$
  - Avec $\mathbb{P}(o|s', a) = 0$ si l'état $s' \in S$ ne contient pas les propriétés de $o \subset \Omega$ (c'est-à-dire $o \notin \mathcal{P}(s')$). Par exemple, un agent joue l'action $x_reads_a_log_file$, ce qui donne lieu à un nouvel état dont une propriété appartenant à la connaissance de l'agent x est $(log_file_content_known_by_x, abc)$. Cette propriété sera donc incluse dans les observations renvoyées à l'agent x.

7.2.3.3 *Intégration des scénarios d'attaque/défense*

D'un point de vue brut, la modélisation du Dec-POMDP pré-spécialisé proposé s'appuie sur des actions pour simuler la manière dont un système en réseau réel réagirait, y compris les vulnérabilités et les contre-mesures appliquées par les agents cyberattaquants et Cyberdéfenseurs.

Un premier défi consiste à construire un scénario d'attaque/défense représentatif d'un système en réseau comportant des vulnérabilités afin de permettre de rendre une attaque en reliant les seules informations disponibles (telles que les tactiques, techniques et procédures connues de MITRE ATT&CK) et en choisissant des contre-mesures de défense pertinentes (issues des mesures d'atténuation de MITRE ATT&CK) et un environnement de déploiement. Un deuxième défi consiste à établir les actions correspondant au scénario d'attaque/défense. Comme les actions modifient les propriétés de l'environnement, elles ont également un impact sur l'espace des états possibles et les transitions entre ceux-ci. De plus, lorsque l'on considère un faible niveau d'abstraction, de nombreuses actions simples peuvent permettre de décrire avec précision les changements opérés dans le réseau. Cependant, cela augmente le nombre d'actions, et encore plus le nombre d'états, car ceux-ci sont des combinaisons des effets des actions.

Ces défis sont directement liés aux questions étudiées concernant la génération automatisée de graphiques d'attaque à l'aide de bases de données disponibles intégrant éventuellement des techniques d'intelligence artificielle, comme dans [105]. Nous n'avons pas l'intention de nous attarder davantage sur ces questions, car elles dépassent le cadre de ce travail.

Ce modèle, illustré en Figure 19, permet de générer un simulateur multi-agents fidèle, où chaque action et observation est explicitement définie, facilitant l'extension à divers contextes de Cyberdéfense.

Traduction opérationnelle des arbres AD vers le modèle séquentiel. La chaîne de traduction utilisée dans ce manuscrit est la suivante :

- les nœuds et branches AD sont convertis en actions d'attaque/défense avec préconditions et postconditions ;
- ces actions alimentent les composantes du modèle Dec-POMDP $(\mathcal{A}, \mathsf{T}, \mathsf{O}, \mathsf{R})$ via les propriétés des nœuds réseau ;
- les objectifs de défense associés sont ensuite exprimés au niveau organisationnel (missions/rôles) puis injectés dans MOISE+MARL via les guides de contraintes.

Dans l'état actuel, cette traduction reste semi-manuelle et pilotée par le concepteur. L'automatisation complète de la projection AD $\rightarrow$ spécifications $\mathcal{M}$OISE$^+$ est identifiée comme une perspective.

Approche d'intégration MITRE ATT&CK : Nous suggérons une approche manuelle de haut niveau que nous avons utilisée pour intégrer les informations MITRE ATT&CK sous forme d'arborescence AD, car elle formalise les actions à jouer dans un scénario et leurs interactions avec l'environnement. Elle vise à aider à établir les actions d'attaque/défense qui seront finalement intégrées dans le simulateur :

- Pour une menace persistante avancée (APT) donnée, nous avons identifié les tactiques, techniques et procédures pertinentes de MITRE ATT&CK qui semblaient pertinentes pour un système en réseau
- Nous avons produit une description reliant les tactiques identifiées entre elles et les techniques, sous-techniques et procédures associées afin de créer un scénario décrivant comment le groupe APT pourrait attaquer le système en réseau. Cette étape définit la topologie du réseau avec ses principales propriétés (telles qu'un réseau d'entreprise composé de plusieurs serveurs de bases de données dédiés communiquant via FTP et HTTP, etc.)
- Nous avons créé une arborescence AD comme proposé dans [151] avec les tactiques comme objectifs d'action principaux et les techniques, sous-techniques et procédures dans la partie inférieure de l'arborescence. Nous avons veillé à disposer de plusieurs chemins pour atteindre un même objectif d'action principal. Nous avons pris soin de définir chaque action d'attaque avec des conditions préalables et des conditions postérieures basées sur les propriétés de l'environnement.
- Nous avons extrait les techniques/sous-techniques MITRE ATT&CK liées à la détection et aux mesures d'atténuation que nous avons ajoutées dans l'arbre AD afin d'enrichir les nœuds d'attaque. Nous avons veillé à définir chaque action de défense avec des conditions préalables et des conditions postérieures basées sur des propriétés dans l'environnement.

- Nous avons également répertorié et défini les principales actions environnementales spécifiques au déploiement à partir de la description précédente de l'environnement de déploiement ou des actions d'attaque/défense étendues qui sont communes aux Cyberdéfenseurs et aux cyberattaquants. Cette étape permet d'obtenir un environnement plus réaliste, fournissant un nombre représentatif d'actions plausibles qu'un agent peut choisir dans de nombreux systèmes. Ces actions communes pourraient inclure au moins : 1) Lecture et écriture de fichiers ; 2) Créer, supprimer, copier, déplacer, renommer, modifier les propriétés des fichiers/dossiers. ; 3) Accéder à un dossier, accéder au dossier parent ; 4) Sélectionner un fichier/dossier pour y appliquer des actions ultérieures ; 5) Exécuter un fichier binaire ; 6) Utilisation d'un protocole réseau (tel que HTTP, FTP, SSH, etc.). ; 7) Autres interactions avec les lignes de commande de base concernant la surveillance ou le contrôle du système. Ensuite, les propriétés environnementales associées doivent décrire un système de fichiers, une interface de terminal, un port avec des règles, les propriétés des paramètres du système d'exploitation, etc.

7.3 DESCRIPTION ET MISE EN ŒUVRE DANS L'ACTIVITÉ

L'Algorithme 2 décrit le déroulement général de l'activité de modélisation. Chaque étape est explicitée ci-dessous afin d'en préciser les objectifs et le rôle dans la construction du modèle final.

ÉTAPE 1 : FORMALISATION MANUELLE DES FONCTIONS COMPOSANTES. La première étape consiste à dériver manuellement, à partir des descriptions informelles de l'objectif global et des contraintes organisationnelles, trois fonctions fondamentales : la fonction de récompense R^j_H, la fonction d'arrêt S^j_H, et la fonction de rendu optionnelle Render^j_H. Cette étape requiert l'expertise des concepteurs, qui doivent transformer des objectifs de haut niveau (souvent exprimés en langage naturel ou sous forme de règles métiers) en spécifications formelles permettant l'évaluation de trajectoires dans l'environnement simulé.

ÉTAPE 2 : ENTRAÎNEMENT DES AUTO-ENCODEURS POUR LES OBSERVATIONS. Une fois les historiques d'interactions collectés, les observations conjointes Ω^j en sont extraites. Comme leur dimension peut être très élevée dans un contexte multi-agent, elles sont compressées à l'aide d'auto-encodeurs. L'encodeur Enc apprend à transformer les observations en représentations latentes compactes z_t, tandis que le décodeur Dec reconstruit les observations originales à partir de ces latents. L'objectif est de minimiser l'erreur de reconstruction, garantissant ainsi que les latents conservent l'information essentielle.

ÉTAPE 3 : ENCODAGE DES OBSERVATIONS DANS LES HISTORIQUES. Les auto-encodeurs entraînés sont ensuite utilisés pour transformer l'ensemble des historiques $\mathcal{D}_{H^j}$ en séquences d'états latents. Chaque observation ω^j_t est convertie en une représentation z_t, ce qui permet de constituer un nouvel ensemble d'entraînement $\mathcal{B}$ composé de triplets (z_t, a^j_t, z_{t+1}). Cet encodage réduit la complexité des données d'entrée et prépare l'entraînement du modèle de dynamique.

ÉTAPE 4 : ENTRAÎNEMENT DU MODÈLE DE DYNAMIQUE RÉCURRENT (RLDM). À partir de l'ensemble encodé $\mathcal{B}$, un modèle récurrent de dynamique latente (RLDM) est en-

Algorithme 2 : Vue algorithmique de l'activité de modélisation

Input : Environnement initial $\mathcal{E}$, historiques conjoints $\mathcal{D}_{H^j}$; objectif informel $\mathcal{G}_{inf}$; facteur d'actualisation γ ; espace des actions A ; espace des observations Ω

Output : l'environnement simulé $d \in D \cup OD$

`// 1. Formalisation manuelle des fonctions composantes`

$(R^j_H, S^j_H, \texttt{Render}^j_H) \leftarrow \texttt{manual_formalize}(\mathcal{G}_{inf}, \mathcal{E}, A, \Omega)$

if *utilisateur souhaite modélisation manuelle* **then**

  return $(S, \{A_i\}, T, R, \{\Omega_i\}, O, \gamma) \leftarrow \texttt{manual_model}(\mathcal{G}_{inf}, \mathcal{E}, A, \Omega)$

`// 2. Entraîner les auto-encodeurs pour les observations`

Extraire les observations $\Omega^j = \{\omega^j_t\}$ à partir des historiques $\mathcal{D}_{H^j}$

Entraîner un auto-encodeur (Enc, Dec) sur Ω^j en minimisant l'erreur de reconstruction

`// 3. Encoder les observations dans les historiques`

Pour chaque historique $h^j = (\omega^j_t, a^j_t) \in \mathcal{D}_{H^j}$, encoder chaque observation conjointe $z_t = Enc(\omega^j_t)$ pour constituer l'ensemble d'entraînement $\mathcal{B} = \{\{(z_t, a^j_t, z_{t+1})\} = h^j_z, h^j_z \in \mathcal{D}_{H^j}\}$

`// 4. Entraîner le RLDM`

Initialiser le RLDM $\mathcal{T}^z = f(g)$

for $h^j_z \in \mathcal{B}$ **do**

  for $(z_t, a^j_t, z_{t+1}) \in h^j$ **do**

    Entraîner le RLDM $\mathcal{T}^z$ en minimisant l'erreur quadratique moyenne (*Mean Squared Error* – MSE) entre la prédiction $\hat{z}_{t+1}$ et la valeur réelle z_{t+1}.

`// 5. Sauvegarder les observations initiales et former le JOPM`

$\Omega^{\mathcal{T}^j}_0 \leftarrow \{\omega^j_0\}$ extraites des historiques $\mathcal{D}_{H^j}$

$\mathcal{T}^j(h_{t-1}, \omega_t, a_t) = \langle f(h_{t-1}, Enc(\omega^j_t), a^j_t), Dec(\mathcal{T}^z(h_{t-1}, Enc(\omega^j_t), a^j_t))\rangle$

`// 6. Retourner les éléments modélisés`

return $(\mathcal{T}^j, \Omega^{\mathcal{T}^j}_0, R^j_H, S^j_H, \mathit{Render}^j_H)$

traîné. Ce modèle, noté $\mathcal{T}^z$, apprend à prédire l'évolution des états latents en fonction de l'historique caché $\tilde{h}_{t-1}$, de l'état encodé courant z_t, et de l'action conjointe a^j_t. L'entraînement se fait en minimisant l'erreur quadratique moyenne entre la prédiction $\hat{z}_{t+1}$ et le latent réel z_{t+1}. Ce mécanisme d'apprentissage permet de capturer la dynamique de l'environnement sans accès direct à l'état global.

ÉTAPE 5 : CONSTRUCTION DU JOPM. Une fois le RLDM entraîné, les observations initiales $\Omega^{\mathcal{T}^j}_0$ sont extraites des historiques et utilisées pour initialiser le simulateur. Le JOPM $\mathcal{T}^j$ est alors défini en combinant le modèle récurrent f, l'encodeur Enc, et le décodeur Dec. Ainsi, pour toute observation et action donnée, $\mathcal{T}^j$ met à jour l'état caché et génère une observation prédite, constituant ainsi un simulateur complet des interactions multi-agents.

ÉTAPE 6 : SORTIES DE L'ACTIVITÉ. Enfin, l'activité retourne l'ensemble des éléments modélisés : le JOPM $\mathcal{T}^j$, l'ensemble des observations initiales $\Omega^{\mathcal{T}^j}_0$, la fonction de récompense R^j_H, la fonction d'arrêt S^j_H, et la fonction de rendu éventuelle $Render^j_H$. Ces com-

posants constituent le cœur du jumeau numérique qui sera exploité par les activités d'entraînement, d'analyse et de transfert de la méthode MAMAD.

7.4 BILAN

En synthèse, l'activité de modélisation vise à fournir un environnement simulé fidèle et exploitable pour l'entraînement multi-agent, en combinant formalisme explicite (Dec-POMDP générique) et génération automatique par World Models multi-agents. Les sorties produites (le JOPM, l'ensemble des observations conjointes initiales, la fonction de récompense, la fonction d'arrêt et optionnellement la fonction de rendue) constituent le socle du jumeau numérique utilisé dans les étapes suivantes. Cette approche permet d'assurer l'adaptation (via l'apprentissage sur données réelles), l'explicabilité (par la structure formelle du modèle) et la réutilisabilité (grâce au châssis générique). Toutefois, la fidélité du modèle dépend de la qualité et de la diversité des historiques collectés, et le coût computationnel de l'entraînement des auto-encodeurs et du RLDM peut être élevé pour des environnements complexes. Enfin, la granularité des actions et la couverture des dynamiques réelles restent des limites inhérentes à toute simulation. L'activité d'entraînement exploitera ce modèle simulé pour optimiser les politiques sous contraintes organisationnelles, amorçant ainsi le cycle itératif de la méthode MAMAD.

8

ENTRAÎNER DES POLITIQUES SOUS CONTRAINTES

L'*activité d'entraînement* consiste à optimiser les politiques conjointes des agents dans l'environnement simulé, en tenant compte des contraintes organisationnelles. Elle correspond à la phase de résolution du problème de conception, en exploitant les modèles produits par l'activité de modélisation.

Cette activité est cruciale car elle relie les critères définis dans Section 1.4 concernant la performance (C2), l'adaptation (C3),l'explicabilité (C5) et le contrôle (C4), via l'intégration de contraintes explicites dans l'apprentissage multi-agent.

OBJECTIFS FORMELS

Les **entrées** de l'activité d'entraînement sont :

- l'environnement simulé $d \in D \cup OD$ produit par l'activité de modélisation ;
- les spécifications organisationnelles $\mathcal{MM}$ possiblement issues des spécifications organisationnelles d'un cycle de raffinement précédent ;
- les spécifications organisationnelles informelles $\mathcal{C}_{inf}$ notamment si premier cycle de raffinement.

La **sortie attendue** est une politique conjointe entraînée $\pi^j = \{\pi^j_0, \pi^j_1, \ldots, \pi^j_{|\mathcal{A}|}\}$.

La relation globale peut s'exprimer par :

$$\texttt{train} : (D \cup OD) \times \mathcal{MM} \times \mathcal{C}_{inf} \to \pi^j$$

Si $d \in OD, d = (\Omega, A, \mathcal{T}^j, \Omega_0^{\mathcal{T}^j}, R^j_H, S^j_H, \texttt{Render}^j_H, \gamma)$ et si $d \in D, d = (S, \{A_i\}, T, R, \{\Omega_i\}, O, \gamma)$.

8.1 TRAVAUX MOBILISÉS ET VERROUS IDENTIFIÉS

L'activité d'entraînement des politiques sous contraintes s'appuie sur plusieurs familles de travaux issus du domaine du MARL et de l'intégration de contraintes organisationnelles.

Du côté du MARL, les méthodes classiques telles que l'apprentissage indépendant, l'apprentissage centralisé avec exécution décentralisée (CTDE), ou encore les algorithmes de type Q-learning, Policy Gradient et leurs variantes multi-agents, constituent la base pour optimiser des politiques conjointes dans des environnements simulés. Ces approches sont efficaces pour maximiser la performance collective, mais elles n'intègrent pas nativement de contraintes organisationnelles explicites.

Pour pallier ce manque, plusieurs travaux issus du Safe RL et des CMDP ont été mobilisés. Les méthodes comme CPO ou DCQL permettent d'intégrer des contraintes numériques (sûreté, consommation, risque) dans le processus d'apprentissage, mais leur expressivité reste limitée à des contraintes locales et numériques, sans prise en compte de structures organisationnelles complexes.

Les approches de reward shaping, shielding, ou feedback humain offrent des mécanismes de guidage souple, permettant d'influencer indirectement les politiques apprises.

Cependant, elles ne garantissent pas le respect formel de contraintes organisationnelles et restent difficiles à interpréter.

Enfin, les travaux sur l'intégration de modèles organisationnels symboliques, tels que MOISE$^+$, proposent une formalisation riche des rôles, missions et relations collectives. Toutefois, leur intégration directe dans le processus d'apprentissage MARL reste un verrou majeur, en raison de la difficulté à traduire ces spécifications en contraintes opérationnelles exploitables par les algorithmes d'apprentissage.

En synthèse, les principaux verrous identifiés sont :

- l'absence de cadre unifié permettant d'intégrer des contraintes organisationnelles symboliques dans l'apprentissage MARL ;
- la difficulté à garantir le respect de ces contraintes tout en maintenant la performance et l'adaptabilité des politiques ;
- le manque d'explicabilité et de contrôle sur les politiques apprises dans des environnements complexes et dynamiques.

8.2 POSITIONNEMENT ET CONTRIBUTIONS PROPOSÉES

Pour lever ces verrous, notre approche propose d'hybrider les forces des cadres symboliques et connexionnistes en introduisant le framework MOISE+MARL [13]. Ce cadre permet d'intégrer explicitement des spécifications organisationnelles (rôles, missions, permissions, obligations) dans le processus d'apprentissage multi-agent, en les traduisant sous forme de guides de contraintes (action masking, shaping de récompense, guides d'objectifs) injectés dans les algorithmes MARL.

Notre contribution principale consiste à :

- formaliser l'intégration des spécifications organisationnelles MOISE$^+$ dans le MARL via des guides de contraintes, permettant de restreindre ou d'orienter l'espace des politiques apprises ;
- proposer un nouveau formalisme, l'ODec-POMDP, compatible avec les environnements simulés appris (World Models), pour permettre l'entraînement à partir de données observables uniquement ;
- développer un algorithme d'entraînement générique (voir Algorithme 3) qui articule ces guides de contraintes avec les méthodes MARL existantes, assurant ainsi la compatibilité entre apprentissage connexionniste et respect des contraintes organisationnelles ;
- offrir un certain degré d'explicabilité et de contrôle sur les politiques apprises, grâce à la traçabilité des guides de contraintes et à l'analyse post-hoc des comportements émergents.

Ce positionnement permet de concilier performance, adaptation, contrôle et explicabilité dans l'entraînement des politiques multi-agents sous contraintes, ouvrant la voie à une conception plus robuste et transparente des SMA pour des environnements critiques comme la Cyberdéfense.

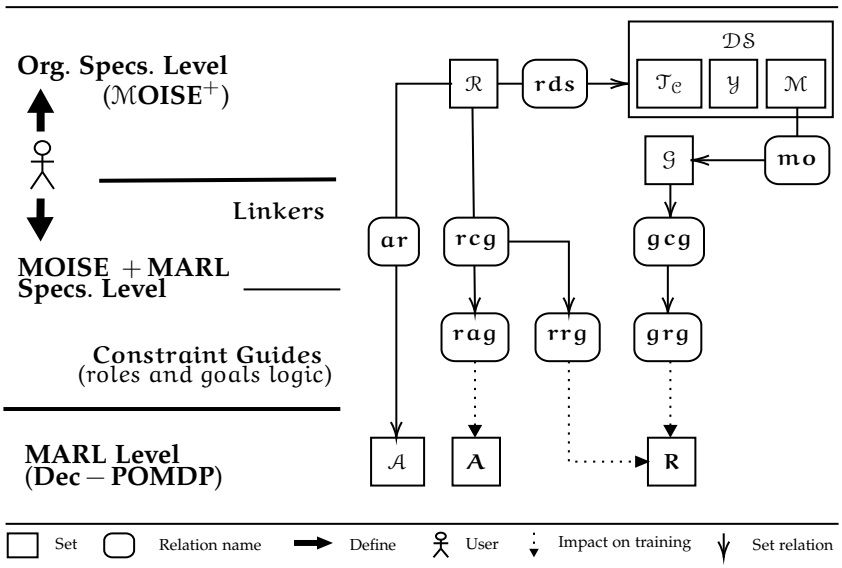


FIGURE 20 : Vue minimale du framework MOISE+MARL : Les utilisateurs définissent d'abord les spécifications $\mathcal{M}$OISE$^+$, qui incluent les rôles ($\mathcal{R}$) et les missions ($\mathcal{M}$), tous deux associés via rds. Ils créent ensuite les spécifications MOISE+MARL en définissant d'abord des guides de contraintes tels que rag et rrg pour spécifier la logique des rôles, et grg pour la logique des objectifs. Des linkers sont ensuite utilisés pour connecter les agents aux rôles via ar et pour lier la logique des guides de contraintes aux spécifications $\mathcal{M}$OISE$^+$ définies. Une fois ces éléments configurés, les rôles peuvent être attribués aux agents, et le framework MARL est mis à jour en conséquence pendant l'apprentissage.

8.2.1 *MOISE+MARL pour lier $\mathcal{M}$OISE$^+$ avec le MARL*

MOISE+MARL introduit des moyens de contrôler ou de guider l'apprentissage des agents en MARL. Sa principale contribution réside dans les **Guides de contraintes**, qui sont trois nouvelles relations introduites pour décrire la logique des rôles et des objectifs dans le formalisme Dec-POMDP :

- **Guide d'action des rôles** $\mathrm{rag} : H \times \Omega \to \mathcal{P}(A \times \mathbb{R})$, relation modélisant un rôle comme un ensemble de règles qui, pour chaque couple constitué d'un historique $h \in H$ et d'une observation reçue par l'agent $\omega \in \Omega$, associe des actions attendues $A \in \mathcal{P}(A)$ chacune associée à une contrainte de dureté $ch \in [0, 1]$ ($ch = 1$ par défaut). En restreignant le choix de l'action suivante parmi celles autorisées, l'agent est contraint d'adhérer au comportement attendu du rôle

- **Guide de récompense des rôles** $\mathrm{rrg} : H \times \Omega \times A \to \mathbb{R} = \{r_m$ if $a \notin A_\omega$, $\mathrm{rag}(h, \omega) = A_\omega \times \mathbb{R}$, $h \in H$; else $0\}$, la relation qui modélise un rôle en ajoutant une pénalité r_m à la récompense globale si la dernière action choisie par l'agent $a \in A$ n'est pas autorisée. Ceci vise à encourager l'agent à adhérer au comportement attendu.

- **Guide de récompense d'objectif** $\mathrm{grg} : H \to \mathbb{R}$, la relation qui modélise un objectif comme une contrainte souple ajoutant un bonus de récompense $r_b \in \mathbb{R}$ si l'historique $h \in H$ de l'agent contient une sous-séquence caractéristique d'un objectif $h_g \in H_g$, encourageant l'agent à l'atteindre.

Enfin, nous introduisons les **Linkers** pour lier les spécifications organisationnelles $\mathcal{M}$OISE$^+$ aux guides de contraintes et aux agents :

- **Agent vers Rôle** $ar : \mathcal{A} \to \mathcal{R}$, la relation bijective reliant un agent à un rôle ;
- **Guide Rôle vers Contrainte** $rcg : \mathcal{R} \to rag \cup rrg$, la relation associant chaque rôle $\mathcal{M}$OISE$^+$ à une relation rag ou rrg, forçant/encourageant l'agent à suivre les actions attendues pour le rôle $\rho \in \mathcal{R}$;
- **Guide Objectif vers Contrainte** $gcg : \mathcal{G} \to grg$, la relation reliant les objectifs aux relations grg, représentant les objectifs comme des récompenses dans MARL.

Résolution du Dec-POMDP avec MOISE+MARL

Un modèle MOISE+MARL est défini par $\mathcal{MM} = \langle \mathcal{OS}, ar, rcg, gcg, rag, rrg, grg \rangle$. La résolution d'un Dec-POMDP avec $mm \in \mathcal{MM}$ consiste à trouver une politique conjointe $\pi^j = \pi^j_0, \pi^j_1, \ldots, \pi^j_{|\mathcal{A}|}$ qui maximise la récompense cumulative espérée (ou satisfait un seuil minimal), représentée par la fonction état-valeur V^{π^j}. Cette valeur reflète le rendement d'un état initial $s \in S$ lors de l'application d'actions conjointes successives $a^j \in A^{|\mathcal{A}|}$ sous les contraintes organisationnelles supplémentaires. La définition de V^{π^j} suit le schéma d'exécution d'agent séquentiel et cyclique (mode AEC) et est formalisée dans Définition 1, intégrant des adaptations basées sur les rôles (en rouge) et les missions (en bleu) qui influencent à la fois l'espace d'action et la récompense. Figure 20 illustre comment les spécifications $\mathcal{M}$OISE$^+$ sont intégrées à la résolution Dec-POMDP via le cadre MOISE+MARL.

Definition 1 **Fonction État-Valeur adaptée aux guides de contraintes en AEC :**

$$V^{\pi^j}(s_t) = \sum_{\substack{a_t \in A \text{ if } rn() < ch_t, \\ a_t \in A_t \text{ else}}} \pi_i(a_t|\omega_t) \sum_{s_{t+1} \in S} T(s_{t+1}|s_t, a_t) \Big[R(s_t, a_t, s_{t+1}) + \sum_{m \in \mathcal{M}_i} \nu_m(t) \frac{grg_m(h_{t+1})}{1 - p + \epsilon} + (1 - ch_t) \times rrg(\omega_t, a_{t+1}) + V^{\pi^j_{i+1 \bmod n}}(s_{t+1}) \Big]$$

Avec $rag(h_t, \omega_t) = A_t \times \mathbb{R}$, $\langle a_t, ch_t \rangle \in A_t \times \mathbb{R}$; $rn : \emptyset \to [0, 1[$, une fonction aléatoire uniforme

Avec $\omega_t = O(\omega_t|s_t, a_t)$; $h_t = \{h_0 = \langle\rangle, h_{t+1} = \langle h_t, \langle \omega_{t+1}, a_{t+1} \rangle\rangle\}$; $\epsilon \in \mathbb{R}_{>0}$; $grg_m(h) = \sum_{(grg_i, w_i) \in mo(m)} w_i \times grg_i(h)$; $\nu_m(t) = \{1 \text{ if } t \in t_c \text{ ; else } 0\}$; $\mathcal{M}_i = \{m_j | \langle ar(i), m_j, t_c, p \rangle \in \mathcal{M}\}$

À chaque pas de temps $t \in \mathbb{N}$ (à partir de $t = 0$), l'agent $i = t \bmod n$ se voit attribuer le rôle $\rho_i = ar(i)$. Pour chaque spécification déontique temporellement valide $d_i = rds(\rho_i) = \langle tc_i, y_i, m_i \rangle$, l'agent est autorisé ($y_i = 0$) ou obligé ($y_i = 1$) à s'engager dans la mission $m_i \in \mathcal{M}$, avec un objectif $\mathcal{G}_{m_i} = mo(m_i)$ et $n \in \mathbb{N}$ agents. En observant ω_t, l'agent sélectionne une action parmi A_t (actions attendues par le rôle) avec une probabilité ch_t, ou parmi A sinon. Si $ch_t = 1$, l'agent est strictement contraint par son rôle. L'action sélectionnée fait passer le système de s_t à s_{t+1}, génère l'observation ω_{t+1} et renvoie une récompense composée de : i) des bonus pour les objectifs atteints dans les missions valides (via les guides de récompenses d'objectifs), pondérés par $\frac{1}{1-p+\epsilon}$; ii) des pénalités du guide de récompenses de rôle, échelonnées par ch_t. Le processus se poursuit dans l'état s_{t+1} avec l'agent $(i + 1) \bmod n$.

8.2.2 *Détails algorithmiques et passage à l'échelle*

Dans l'implémentation expérimentale, plusieurs familles d'algorithmes MARL sont mobilisées (notamment *Multi-Agent Proximal Policy Optimization* (MAPPO), *Multi-Agent Deep Deterministic Policy Gradient* (MADDPG), QMIX, *Counterfactual Multi-Agent Policy Gradients*

(COMA)), avec une prédominance de MAPPO comme compromis stabilité/performance sur les cas étudiés.

Le coût d'entraînement dépend principalement :

- du nombre d'agents $|\mathcal{A}|$, de la longueur d'épisode T, du nombre d'itérations de mise à jour et de la taille des réseaux ;
- de la dimension des observations/actions conjointes, qui croît rapidement avec la taille du système ;
- de la non-stationnarité induite par l'apprentissage simultané de plusieurs politiques.

De manière indicative, le coût suit un ordre de grandeur de type

$$O\big(|\mathcal{A}| \times T \times \text{taille_modèle} \times \text{updates}\big),$$

auquel s'ajoute le coût d'application des guides de contraintes

$$O\big(|\mathcal{A}| \times T \times \text{nombre_de_règles}\big).$$

Les atténuations déjà intégrées dans la thèse sont : (i) compression latente des observations via JOPM/auto-encodeurs, (ii) contraintes organisationnelles qui réduisent l'espace d'actions effectif (*action masking* + *reward guides*), (iii) protocole d'évaluation reproductible multi-runs (seeds, baselines, ablations) pour contrôler la variance, et (iv) boucle de raffinement MAMAD qui permet des itérations courtes et validées par seuils ($\texttt{org_fit}_{min}$, $\bar{r}_{min}$, σ_{max}).

Cette discussion complète les résultats sans introduire de nouvelles expérimentations : elle explicite les compromis de scalabilité observés et justifie les choix algorithmiques effectués.

8.2.3 *Portée des contraintes : multi-objectif, shielding et contraintes temporelles*

Dans cette thèse, la dimension **multi-objectif** est traitée de façon opérationnelle dans la fonction de récompense via les guides $\texttt{grg}$ associés aux missions, et non comme une optimisation de Pareto dédiée avec front explicite. Autrement dit, plusieurs objectifs peuvent être poursuivis simultanément, mais leur arbitrage passe par des pondérations et des priorités définies dans les spécifications organisationnelles et les guides. Cette formulation est cohérente avec l'objectif de contrôle (C4) et d'explicabilité (C5), tout en conservant des algorithmes MARL standards.

Le **shielding** est implémenté selon deux niveaux complémentaires déjà présents dans MOISE+MARL :

- un niveau **dur** via le *action masking* ($\texttt{rag}$, $\texttt{ch}_t = 1$), qui interdit explicitement certaines actions ;
- un niveau **souple** via le *reward shaping* ($\texttt{rrg}$, $\texttt{grg}$, $0 \leqslant \texttt{ch}_t < 1$), qui pénalise ou récompense des trajectoires sans blocage strict.

Cette combinaison explicite le compromis performance/contrôle : plus le masquage est strict, plus le respect des contraintes est assuré, au prix d'une exploration réduite.

Enfin, les **contraintes temporelles et de précédence** sont prises en compte à deux niveaux :

- au niveau déontique, via $\mathrm{rds}(\rho) = \langle tc, y, m \rangle$ et $\nu_m(t)$, qui activent obligations/permissions selon des fenêtres temporelles ;
- au niveau de la modélisation de scénario, via des préconditions/postconditions et des enchaînements d'actions issus des arbres AD, ensuite traduits en missions et guides.

Cette prise en compte est déjà opérationnelle dans le pipeline, mais son automatisation complète (notamment pour la génération systématique des dépendances de précédence) reste un axe d'amélioration identifié.

8.2.4 *Faciliter l'implémentation des* ***Guides de Contrainte***

Puisque les rôles, objectifs et missions sont de simples étiquettes, leur définition est implicite. Cependant, implémenter une relation rag, rrg ou grg nécessite de définir de nombreux historiques, souvent redondants, rendant une définition extensionnelle fastidieuse et peu évolutive. De plus, la logique de chaque **Guide de Contrainte** repose sur l'analyse de trajectoires d'agents : pour chaque historique observé, il faut décider s'il appartient à un ensemble d'historiques attendus et quelles conséquences en tirer (masquage d'actions, ajout de pénalités ou de bonus de récompense). Par exemple, rag restreint les actions disponibles selon l'appartenance de la trajectoire courante à un ensemble H_g et la nouvelle observation.

Une première approche consiste à laisser l'utilisateur définir ces guides par une logique procédurale (scripts Python ou règles spécifiques). Dans ce cas, la relation $b_g : H \to \{0, 1\}$ formalise la décision d'appartenance d'un historique à un ensemble H_g. Cette solution est flexible, car elle peut exploiter la totalité du contexte disponible (positions spatiales, états internes, séquences passées). Toutefois, elle reste coûteuse à écrire et à maintenir, et conduit souvent à des définitions très verbeuses.

Pour dépasser cette limite, nous introduisons les *Trajectory-based Patterns* (TPs), inspirés du Traitement Automatique du Langage. L'idée est de fournir un formalisme déclaratif compact permettant d'exprimer des comportements attendus comme des motifs temporels. Un TP $p \in P$ correspond ainsi à un patron qui capture un ensemble d'historiques de manière intentionnelle. Chaque observation ou action est associée à une étiquette $l \in L$ (via $l : \Omega \cup A \to L$), ce qui rend la manipulation pratique et indépendante des détails bas-niveau de l'environnement. Un TP p peut être :

- une **séquence feuille** $s_l = \langle h, \{c_{min}, c_{max}\} \rangle$, où $h \in H$ désigne une sous-séquence observation/action et $\{c_{min}, c_{max}\}$ est la cardinalité ;
- une **séquence nœud** $s_n = \langle \langle s_{l_1}, s_{l_2}, \dots \rangle, \{c_{min}, c_{max}\} \rangle$, combinant plusieurs séquences en un motif hiérarchique.

Par exemple, le pattern $p =$ "$[o_1, a_1, [o_2, a_2]\langle 0, 2 \rangle]\langle 1, * \rangle$" se lit comme suit : un historique valide doit contenir au moins une occurrence de la paire $\langle o_1, a_1 \rangle$, suivie de zéro à deux occurrences de $\langle o_2, a_2 \rangle$. Ce TP capture donc une famille entière de comportements, sans avoir à lister tous les historiques. L'intérêt des TPs est double : i) Ils permettent de coder de manière **compacte** des comportements étendus dans le temps, difficiles à exprimer autrement. ; ii) Ils facilitent la **réutilisation**, puisque les mêmes motifs peuvent être partagés entre plusieurs rôles ou objectifs.

EXEMPLE CONCRET. Considérons l'environnement *Overcooked-AI* [79], où des agents cuisiniers doivent collaborer en se déplaçant dans un monde en grille et interagir avec les ingrédients (oignons) et les instruments (pots, bols) pour faire de la soupe qui est expédiée en interagissant avec la zone d'expédition. Ici, l'on souhaite reconnaître le comportement suivant : "un agent qui détient un oignon, observe un pot et interagit avec lui pour le remplir". Ce comportement peut être exprimé par le TP suivant :

$$p = [[\#any](*),\ has_onion,\ [\#any](*),\ see_pot,\ interact\ ](1,1)$$

Ce TP peut être exploité dans les guides de contrainte comme montré dans la Table 14 pour inciter l'agent à interagir à nouveau pour récupérer la soupe par exemple.

TABLE 14 : Exemple de guides appliqués au TP "remplir un pot avec un oignon".

Guide	Exemple de règle
RAG (*Role Action Guide*)	Si le TP est satisfait, restreindre les actions possibles : $rag(h, \omega) = \{\texttt{interact} \mapsto 1.0,\ \texttt{nothing} \mapsto 0.2\}$ L'action `interact` est fortement favorisée.
RRG (*Role Reward Guide*)	Ajouter un bonus de rôle lorsque l'action attendue est réalisée : $rrg(h, \omega, a) = \begin{cases} +3 & \text{si } a = \texttt{interact} \\ 0 & \text{sinon} \end{cases}$
GRG (*Goal Reward Guide*)	Accorder une récompense si le TP est détecté / atteint : $grg(h) = \begin{cases} +5 & \text{si le pot est rempli (TP reconnu)} \\ 0 & \text{sinon} \end{cases}$

Ainsi, au lieu de définir extensionnellement de vastes ensembles H_g, l'utilisateur décrit quelques motifs symboliques qui capturent l'essence des comportements recherchés. MOISE+MARL applique alors automatiquement ces motifs aux guides (rag, rrg, grg), rendant leur mise en œuvre plus modulaire, scalable et interprétable.

8.2.5 *Extension de MOISE+MARL aux World Models Multi-Agents*

Dans des environnements réalistes, on ne dispose que des transitions issues des historiques d'actions et d'observations reçues. Pour mieux représenter ce contexte, nous introduisons un nouveau formalisme appelé **Dec-POMDP basé sur les observations** (ODec-POMDP). Un ODec-POMDP $d \in OD$ (avec OD, l'ensemble des ODec-POMDPs) est défini comme un quintuplet : $d = (\mathcal{T}^j, \Omega_0^{\mathcal{T}^j}, R_H^j, S_H^j, \texttt{Render}_H^j)$ où :

- Ω, l'espace d'observations ;
- A, l'espace d'actions ;
- $\mathcal{T}^j$, le JOPM estimant la prochaine observation conjointe ω' à partir de l'historique $\tilde{h} \in \mathcal{H}$, de l'observation conjointe actuelle ω et de l'action conjointe a. Le modèle renvoie également l'état caché récurrent mis à jour $\tilde{h}'$;

- $\Omega_0^{\mathcal{T}^j}$, l'ensemble des observations initiales conjointes ;
- $R_H^j : H \times \Omega \times A \times \Omega \to \mathbb{R}$, la fonction de récompense basée sur l'historique, calculant la récompense depuis l'historique précédent, l'observation et action courante et l'observation suivante ;
- $S_H^j : H \to \{0, 1\}$ la fonction d'arrêt basée historique qui indique si l'agent a atteint la fin de son épisode ou réussi son objectif ;
- $\texttt{Render}_H^j : H \to \emptyset$, une fonction de rendue visuelle optionnelle basée historique ;
- $\gamma \in [0, 1]$, le facteur d'actualisation.

Cette formulation permet aux agents MARL d'opérer uniquement à partir de données observables, rendant la méthode compatible avec les environnements simulés appris.

Résolution d'un ODec-POMDP avec MOISE+MARL

Résoudre un ODec-POMDP avec des contraintes $mm \in \mathcal{MM}$ consiste à trouver une politique conjointe $\pi^j = \{\pi_0^j, \pi_1^j, \ldots, \pi_{|\mathcal{A}|}^j\}$ qui maximise la récompense cumulée espérée (ou qui satisfait un seuil minimal), via la fonction de valeur basée sur les observations $V_{\mathcal{T}^j}^{\pi^j}$. Cette fonction représente le retour attendu depuis une observation conjointe initiale $\omega^j \in \Omega_0^{\mathcal{T}^j}$, un historique h^j et un état caché $\tilde{h}$, en appliquant des actions conjointes $a^j \in A^{|\mathcal{A}|}$ sous contraintes organisationnelles $\mathcal{MM}$, et en utilisant $\mathcal{T}^j$ pour approximer les transitions.

La définition complète de $V_{\mathcal{T}^j}^{\pi^j}$ est donnée dans Définition 2, et intègre les adaptations basées sur les rôles (en rouge) et sur les missions (en bleu), qui influencent à la fois l'espace d'actions conjointes et la récompense. La Figure 20 illustre comment les spécifications $\mathcal{M}OISE^+$ sont injectées dans la résolution d'un ODec-POMDP à l'aide du cadre MOISE+MARL.

Definition 2 **Fonction Observation-Valeur adaptée aux guides de contraintes en mode parallèle :**

$$V^{\pi^j}(\tilde{h}_{t-1}, h_{t-1}^j, \hat{\omega}_t^j) = \sum_{\substack{a_t^j \in A^j \text{ if } rn() < ch_t, (\tilde{h}_t, \hat{\omega}_{t+1}^j) \in \mathcal{H} \times \hat{\Omega}^j \\ a_t^j \in A_t^j \text{ else}}} \pi_i(a_t^j | \hat{\omega}_t^j) \sum \mathcal{T}^j(\langle \tilde{h}_t, \hat{\omega}_{t+1}^j \rangle | \tilde{h}_{t-1}, \hat{\omega}_t, a_t^j) \Big[R_H^j(h_{t-1}^j, \hat{\omega}_t^j, a_t^j, \hat{\omega}_{t+1}^j) + grg_m^j(h_t^j) + (1 - ch_t) \times rrg^j(\hat{\omega}_t^j, a_{t+1}^j) + V^{\pi^j}(\tilde{h}_t, h_t^j, \hat{\omega}_{t+1}^j) \Big]$$

Avec $\tilde{h}_{-1} = \mathbf{o}$ and $\tilde{\omega}_0^j \in \Omega_0^{\mathcal{T}^j}$; $a_t^j = \langle a_{t,0}, a_{t,1} \ldots a_{t,|\mathcal{A}|} \rangle$; $\omega_t^j = \langle \omega_{t,0}, \omega_{t,1} \ldots \omega_{t,|\mathcal{A}|} \rangle$;

$h_t^j = \langle h_{t,0}, h_{t,1} \ldots h_{t,|\mathcal{A}|} \rangle = \langle \langle h_{t-1,i}, \omega_{t,i}, a_{t,i} \rangle \rangle_{i \in \mathcal{A}}$

Avec $\langle rag_i, rrg_i \rangle = rcg(ar(i))$; $rn : \emptyset \to [0, 1[$, une fonction aléatoire uniformef

$$A_t^j \times \mathbf{R}^{|\mathcal{A}|} = rag^j(h_t^j, \tilde{\omega}_t^j) = \langle rag_i(h_{t,i}, \omega_{t,i}) \rangle_{i \in \mathcal{A}} ; rrg^j(h_t^j, \tilde{\omega}_t^j, a_t^j) = \sum_{i \in \mathcal{A}} rrg_i(h_{t,i}, \omega_{t,i}, a_{t,i})$$

$$grg_m(h) = \sum_{(grg_i, w_i) \in mo(m)} w_i \times grg_i(h) ; grg_m^j(h_t^j) = \sum_{i \in \mathcal{A}} \sum_{m \in \mathcal{M}_i} \nu_m(t) \frac{grg_m(h_{t,i})}{1 - p + \epsilon} ; \epsilon \in \mathbb{R}_{>0} ;$$

$$\nu_m(t) = \{1 \text{ if } t \in t_c \text{ ; else } 0\} ; \mathcal{M}_i = \{m_j | \langle ar(i), m_j, t_c, p \rangle \in \mathcal{M}\}$$

En mode parallèle, à chaque pas de temps $t \in \mathbb{N}$ (en commençant à $t = 0$), chaque agent $i \in \mathcal{A}$ est assigné à un rôle $\rho_i = ar(i)$. Pour chaque spécification déontique temporellement

valide $d_i = \mathrm{rds}(\rho_i) = \langle tc_i, y_i, m_i \rangle$, l'agent est soit autorisé ($y_i = 0$), soit obligé ($y_i = 1$) de s'engager dans la mission $m_i \in \mathcal{M}$, avec ensemble d'objectifs $\mathcal{G}_{m_i} = \mathrm{mo}(m_i)$.

Lorsque les agents observent $\tilde{\omega}^j_t$, ils sélectionnent leurs actions dans $A_{i,t}$ (dérivées via les guides de récompense de rôle) avec une probabilité ch_t, ou dans A_t sinon. Si $ch_t = 1$, les agents sont strictement contraints par leur rôle.

Les transitions d'observation et d'état sont approximées via le JOPM $\mathcal{T}^j$ à partir de l'état caché précédent $\tilde{h}_{t-1}$, de l'observation conjointe ω^j_t et de l'action conjointe a^j_t. La fonction de récompense R^j_H utilise ces mêmes informations, ainsi que l'observation suivante, pour produire la récompense. Des bonus ou malus sont ensuite ajoutés selon : i) l'atteinte d'objectifs valides (via les guides de récompense des objectifs, pondérés par $\frac{1}{1-p+\epsilon}$), ii) la conformité au rôle (via les guides de récompense de rôle, pondérés par $1 - ch_t$).

Bien qu'il n'est pas indiqué, en pratique la fonction d'arrêt S^j_H est utilisée à la place du facteur d'actualisation afin de mettre un terme à la boucle des étapes. Le plus souvent cette fonction d'arrêt consiste à renvoyer vrai quand un nombre seuil d'étapes a été franchi. De même, si la fonction de rendue basée historique `Render` est fournie, celle-ci est utilisée pour donner une visualisation des observations de chaque agent. Cette fonction est utilisée surtout à des fins d'explicabilité et de supervision.

8.3 DESCRIPTION ET MISE EN ŒUVRE DANS L'ACTIVITÉ

L'Algorithme 3 décrit le déroulement général de l'activité d'entraînement. Chaque étape est détaillée ci-après.

ÉTAPE 1 : FORMALISATION DES CONTRAINTES. Si les spécifications $\mathcal{MM}$ ne sont pas fournies, elles sont dérivées manuellement à partir des contraintes informelles $\mathcal{C}_{\text{inf}}$.

ÉTAPE 2 : INITIALISATION. Initialiser les paramètres de la politique conjointe π^j et un buffer d'expérience $\mathcal{B}$.

ÉTAPE 3 : EXÉCUTION D'ÉPISODES SIMULÉS. Pour chaque épisode, échantillonner une observation initiale $\omega^j_0 \in \Omega^{\mathcal{T}^j}_0$, initialiser l'historique h^j_{-1} et l'état caché $\tilde{h}_{-1}$.

ÉTAPE 4 : SÉLECTION DES ACTIONS SOUS CONTRAINTES. À chaque étape t, calculer l'ensemble des actions autorisées $A^j_t = \mathrm{rag}^j(h^j_t, \omega^j_t)$. L'agent sélectionne son action a^j_t parmi A^j_t avec probabilité ch_t, sinon dans A.

ÉTAPE 5 : TRANSITION ET MISE À JOUR DU JOPM. La transition $(\tilde{h}_t, \omega^j_{t+1})$ est générée par $\mathcal{T}^j(\tilde{h}_{t-1}, \omega^j_t, a^j_t)$.

ÉTAPE 6 : CALCUL DES RÉCOMPENSES. La récompense r_t combine :

- la récompense de base R^j_H,
- un bonus $\mathrm{grg}^j(h^j_t)$ si des objectifs sont atteints,
- un bonus/malus $(1 - ch_t) \times \mathrm{rrg}^j(h^j_t, \omega^j_t, a^j_t)$ lié au respect du rôle.

ÉTAPE 7 : MISE À JOUR DE L'EXPÉRIENCE ET APPRENTISSAGE. Les transitions sont stockées dans $\mathcal{B}$, et la politique π^j est optimisée par tout algorithme MARL compatible (Q-learning, Policy Gradient, CTDE, etc.).

ÉTAPE 8 : SORTIE. À la fin des épisodes, l'activité retourne la politique conjointe entraînée π^j.

Algorithme 3 : Vue algorithmique de l'activité d'entraînement

Input : Environment simulé comme Dec-POMDP ou ODec-POMDP $d \in D \cup OD$; Spécifications organisationnelles $\mathcal{MM}$; Contraintes de conception informelles $\mathcal{C}_{inf}$;

Output : π^j : Politique conjointe entraînée

if $\mathcal{MM} = \emptyset$ **then**
  $\mathcal{MM} \leftarrow$ `manual_formalize`$(\mathcal{C}_{inf})$ `// Formalisation manuelle specs. orgs.`

`// Si env. simulé manuel, entraînement classique`

if $d \in D$ **then**
  entrainer une politique conjointe π^j via MARL contraint selon Définition 1

else
  Initialiser les paramètres de la politique π^j et du buffer de replay $\mathcal{B}$
  foreach *épisode* $e = 1 \dots N$ **do**
    Échantillonner $\omega_0^j \sim \Omega_0^{\mathcal{T}^j}$, initialiser $\tilde{h}_{-1} \leftarrow \mathbf{0}$
    Initialiser l'historique $h_{-1}^j \leftarrow \emptyset$
    foreach *étape* $t = 0 \dots T$ **do**
      Calculer $A_t^j = rag^j(h_t^j, \omega_t^j)$ via les guides de récompense de rôle dans $\mathcal{MM}$
      if $rn() < ch_t$ **then**
        Sélectionner $a_t^j \sim \pi^j(\cdot|\omega_t^j)$ dans l'ensemble A_t^j (contraint)
      else
        Sélectionner $a_t^j \sim \pi^j(\cdot|\omega_t^j)$ dans l'ensemble A_t
      $(\tilde{h}_t, \omega_{t+1}^j) \leftarrow \mathcal{T}^j(\tilde{h}_{t-1}, \omega_t^j, a_t^j)$ `// Prédiction JOPM`
      $r_t \leftarrow \gamma^t \times R_H^j(h_{t-1}^j, \omega_t^j, a_t^j, \omega_{t+1}^j)$ `// Récompense de base`
      $r_t \leftarrow r_t + grg^j(h_t^j)$ `// Bonus via guides d'objectifs`
      $r_t \leftarrow r_t + (1 - ch_t) \times rrg^j(h_t^j, \omega_t^j, a_t^j)$ `// Bonus/malus via guides de rôle`
      Ajouter $(\omega_t^j, a_t^j, r_t, \omega_{t+1}^j)$ à $\mathcal{B}$
      Mettre à jour $h_t^j \leftarrow \langle\langle h_{t-1,i}, \omega_{t,i}, a_{t,i}\rangle\rangle_{i \in \mathcal{A}}$
      Entraîner π^j avec des mini-lots tirés de $\mathcal{B}$ en utilisant toute méthode MARL

return π^j

8.4 BILAN

En synthèse, l'activité d'entraînement permet de produire des politiques conjointes optimisées, intégrant à la fois les contraintes organisationnelles explicites et les dynamiques apprises via les World Models multi-agents. Cette approche concilie performance et adaptation tout en offrant un certain degré de contrôle et d'explicabilité.

Les principales limites identifiées concernent :

- la scalabilité du MARL contraint à un grand nombre d'agents ;
- la dépendance aux données disponibles pour apprendre un JOPM fidèle ;
- le compromis entre respect strict des contraintes et performance optimale.

Ces éléments préparent l'**activité d'analyse**, qui vise à évaluer et expliciter les politiques entraînées.

9

ANALYSER LES COMPORTEMENTS ÉMERGENTS

L'*activité d'analyse* vise à évaluer et expliciter les politiques conjointes apprises. Elle fournit une explication des comportements observés en termes de rôles, objectifs et missions. Cette activité joue un rôle central pour l'explicabilité en reliant les dynamiques apprises aux structures organisationnelles interprétables.

OBJECTIFS FORMELS

Les **entrées** de l'activité d'analyse sont :

- un ODec-POMDP d représentant l'environnement simulé appris ;
- une politique conjointe entraînée π^j ;
- une spécification organisationnelle initiale $\mathcal{MM}$ (optionnelle) ;
- la fenêtre d'épisodes de test w_{episodes}.

Les **sorties attendues** sont :

- l'ensemble des spécifications implicites inférées $\mathcal{MM}_{\text{inferred}}$;
- le score d'adéquation organisationnelle `org_fit` sur les w_{episodes} derniers épisodes ;
- la moyenne des récompenses $\bar{r}$ sur les w_{episodes} derniers épisodes ;
- l'écart-type des récompenses σ sur les w_{episodes} derniers épisodes ;
- le booléen qui indique si l'utilisateur souhaite poursuivre les cycles de raffinement `running_refinement`.

La relation globale peut être exprimée par :

$$(\mathcal{MM}_{\text{inferred}}, \texttt{org_fit}, \bar{r}, \sigma, \texttt{running_refinement}) \leftarrow \texttt{analyze}(\mathrm{d}, \pi^j, \mathcal{MM}, w_{\text{episodes}})$$

9.1 TRAVAUX MOBILISÉS ET VERROUS IDENTIFIÉS

L'activité d'analyse des comportements émergents s'appuie sur trois axes principaux : l'explicabilité post-hoc en apprentissage automatique, l'analyse des politiques multi-agents, et l'inférence organisationnelle à partir de trajectoires.

Les méthodes d'explicabilité locales (SHAP, LIME, CAV) expliquent les décisions individuelles, mais restent limitées pour l'analyse globale des dynamiques collectives. Les modèles interprétables (arbres de décision, extraction de concepts) offrent une certaine lisibilité, mais peinent à capturer la complexité organisationnelle à grande échelle.

Des approches de clustering de trajectoires ou d'inférence de rôles permettent d'identifier des spécialisations ou missions collectives, mais nécessitent souvent un paramétrage manuel et restent déconnectées des modèles organisationnels symboliques.

À ce jour, aucune méthode n'extrait automatiquement des spécifications organisationnelles complètes (rôles, missions, permissions, obligations) à partir des trajectoires, ni ne relie systématiquement les comportements émergents à un cadre symbolique tel que MOISE$^+$.

Les principaux verrous sont :

- l'absence de cadre pour évaluer quantitativement l'explicabilité organisationnelle ;
- le manque d'automatisation dans l'inférence des structures organisationnelles ;
- l'absence de méthode pour relier les structures extraites à des modèles symboliques existants.

Ces limites motivent le développement de la méthode TEMM et d'Auto-TEMM, pour automatiser l'inférence organisationnelle, quantifier l'adéquation organisationnelle, et relier les comportements émergents à des spécifications formelles exploitables dans la boucle de conception.

9.2 POSITIONNEMENT ET CONTRIBUTIONS PROPOSÉES

L'activité d'analyse est centrale dans la méthode MAMAD, car elle relie les dynamiques apprises à des structures organisationnelles interprétables et évalue quantitativement leur alignement.

Définition : L'*adéquation organisationnelle* est théorisée comme la mesure de la conformité entre les comportements collectifs observés et les spécifications organisationnelles attendues (rôles, objectifs, missions, permissions, obligations). Elle est quantifiée par un indicateur global (*Organizational Fit* – OF), combinant cohérence structurelle (*Structural Organizational Fit* – SOF) et fonctionnelle (*Functional Organizational Fit* – FOF) extraites des trajectoires.

Notre approche propose :

- une mesure quantitative globale de l'adéquation organisationnelle OF ;
- l'extraction automatique de spécifications organisationnelles implicites à partir des trajectoires ;
- une méthode flexible, utilisable en mode manuel (TEMM) pour l'analyse experte, ou automatisée (Auto-TEMM).

TEMM permet un contrôle expert et une analyse qualitative, tandis que **Auto-TEMM** automatise l'analyse et optimise les hyperparamètres pour une utilisation à grande échelle.

Les contributions principales sont :

- l'introduction d'un indicateur robuste d'adéquation organisationnelle ;
- la formalisation d'une méthode d'inférence organisationnelle adaptée au multi-agent ;
- l'automatisation de l'analyse via Auto-TEMM ;
- l'intégration de cette analyse dans la boucle de conception MAMAD.

En synthèse, l'adéquation organisationnelle permet d'évaluer, comparer et raffiner objectivement les comportements émergents, garantissant que les SMA produits sont performants, explicables et alignés sur les exigences organisationnelles.

9.2.1 *La méthode TEMM*

La méthode TEMM fait partie du composant d'explicabilité du cadre MOISE+MARL. Elle repose sur l'hypothèse que les comportements des agents, malgré une variabilité apparente, présentent des régularités lorsqu'ils atteignent des récompenses cumulées comparables. Ainsi, des comportements différents peuvent être interprétés comme des variantes bruitées d'un nombre limité de stratégies latentes. D'après la loi des grands nombres, une moyenne sur un ensemble suffisant d'historiques conjoints réussis permet de filtrer le bruit et de révéler des stratégies typiques.

La méthode exploite des techniques d'apprentissage non supervisé pour inférer des spécifications organisationnelles à partir des trajectoires observées des agents, et pour calculer l'**adéquation organisationnelle** (OF) entre les comportements émergents et les rôles, objectifs et missions attendus. Elle se décline en trois volets.

1) RÔLES ET HÉRITAGE DE RÔLES Les trajectoires $(\omega, a) \in \Omega \times A$ sont regroupées en clusters à l'aide de métriques de distance (par exemple *Longest Common Sequence* – LCS, Smith-Waterman [193]), éventuellement après encodage one-hot des actions. Dans ce cadre, un **rôle** ρ est défini comme une politique dont les agents partagent une Séquence Parente (SP) dans leurs historiques. Nous définissons une **Séquence Parente** comme la *séquence de consensus locale* obtenue à partir de l'*alignement local optimal* des séquences de transitions considérées, au sens de l'algorithme de Smith–Waterman [193]. Cette séquence capture le "motif le plus représentatif" partagé entre deux (ou plusieurs) séquences d'agents, en ignorant le bruit et les divergences en début et fin de séquence. Elle n'est pas nécessairement la plus longue sous-séquence commune. La notion de *séquence de consensus locale* est ici entendue dans le sens classique de la bio-informatique, à savoir une séquence représentant les symboles majoritaires dans une région alignée [4, 86, 191].

Un rôle ρ_2 hérite de ρ_1 si $\mathrm{SP}(\rho_2) \subseteq \mathrm{SP}(\rho_1)$. Le clustering hiérarchique permet d'extraire ces SP et de construire une hiérarchie des rôles. Pour chaque cluster, un centroïde de transitions moyennes par pas de temps est calculé. Une procédure de sélection retient les transitions les plus représentatives, interprétées comme des **règles comportementales** associées à un rôle. Une faible représentativité conduit à inclure toutes les transitions, au risque de sur-apprentissage. Le **SOF** (structural organizational fit) est calculé comme l'inverse normalisé de la variance globale dans les clusters de transitions : une faible variance indique une forte cohérence structurelle.

2) OBJECTIFS, PLANS ET MISSIONS Les trajectoires d'observations sont regroupées en clusters à l'aide de métriques de distance ou via des méthodes vectorielles (par ex. K-means sur des embeddings de trajectoires). Pour chaque cluster, une trajectoire centroïde est calculée, associant chaque pas de temps à une observation moyenne. La **représentativité** est définie comme l'inverse normalisé de la variance locale par pas de temps. Un seuil minimal de représentativité est appliqué pour sélectionner les observations saillantes, interprétées comme des **objectifs intermédiaires** – jalons importants vers l'objectif global. Si la représentativité minimale est élevée, seules les observations très fréquentes sont retenues, assurant robustesse et pertinence. Les **plans** sont inférés comme des sous-séquences de transitions menant systématiquement à ces objectifs. Une **mission** regroupe un ou plusieurs objectifs poursuivis collectivement par un ou plusieurs agents. Le **FOF** (functional organizational fit) évalue la cohérence fonctionnelle des agents dans l'atteinte de ces objectifs intermédiaires, calculé comme l'inverse normalisé de la variance dans les clusters d'observations.

3) PERMISSIONS ET OBLIGATIONS Les permissions et obligations sont dérivées en analysant si les agents remplissant un rôle accomplissent systématiquement (ou exclusivement) certaines missions dans des contraintes temporelles données. Une obligation implique une exclusivité, tandis qu'une permission implique une optionnalité.

AGRÉGATION ET INTERPRÉTATION L'**adéquation organisationnelle globale** est obtenue en agrégeant les scores structurel et fonctionnel. Un score élevé indique que les spécifications inférées (rôles, objectifs, missions) sont représentatives des comportements réellement appris. Un score faible suggère une faible structuration ou des comportements peu cohérents. Bien que certains hyperparamètres de clustering puissent nécessiter un ajustement manuel pour garantir la robustesse de l'extraction des rôles et objectifs, TEMM fournit une approche méthodique pour analyser les comportements organisationnels émergents et affiner les spécifications en conséquence.

9.2.2 *Auto-TEMM : la méthode TEMM étendue avec optimisation des hyperparamètres*

Un problème majeur rencontré avec TEMM est la nécessité de choisir manuellement plusieurs hyperparamètres (métriques de distance, seuils de clustering, seuils de représentativité), ce qui ralentit le processus d'analyse et limite son automatisation. Une représentativité trop faible conduit à du sur-apprentissage, tandis qu'une représentativité trop élevée limite les contraintes et ralentit la convergence.

Pour surmonter cette difficulté, nous proposons un processus d'**optimisation d'hyperparamètres** (*Hyper-Parameter Optimization* – HPO) consistant en une recherche par grille (grid search) sur les combinaisons possibles, visant à maximiser le adéquation organisationnel et minimiser le nombre de clusters :

- (i) Pour les observations et les transitions, appliquer une recherche conjointe sur les métriques de distance et les seuils de clustering afin de minimiser la variance intra-cluster et le nombre de clusters ;
- (ii) Déterminer les représentativités minimales (structurelle et fonctionnelle) pour obtenir des objectifs et des rôles concis et pertinents. Diminuer cette représentativité augmente la couverture, mais réduit la robustesse des contraintes organisationnelles. Une valeur élevée limite la généralisation, tandis qu'une valeur trop faible entraîne un sur-apprentissage. Cela est illustré dans la Figure 21, où le temps de convergence normalisé est tracé en fonction de la représentativité minimale. Une convergence rapide indique des contraintes organisationnelles fortes et cohérentes, tandis qu'une convergence lente suggère des contraintes faibles ou incohérentes.

Nous adoptons un compromis basé sur le **point de coude** du graphique convergence/temps (voir Figure 21), en choisissant la plus grande représentativité assurant une convergence normalisée de 3.5%. Cette stratégie permet d'obtenir des spécifications utiles, interprétables et généralisables, sans complexité excessive.

9.3 DESCRIPTION ET MISE EN ŒUVRE DANS L'ACTIVITÉ

L'Algorithme 4 formalise le déroulement général de l'activité d'analyse. Chaque étape est détaillée ci-dessous afin d'expliciter les mécanismes qui permettent d'inférer une spécification organisationnelle implicite et de calculer l'adéquation organisationnelle.

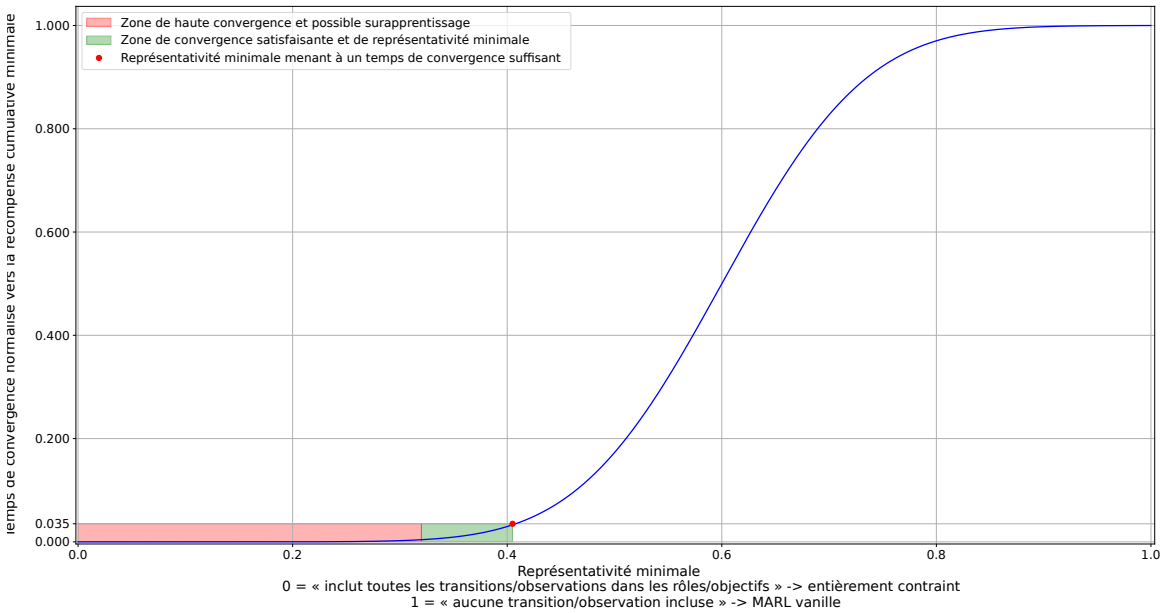


FIGURE 21 : Temps de convergence normalisé en fonction de la représentativité minimale

ÉTAPE 1 : COLLECTE DES TRAJECTOIRES. La première étape consiste à exécuter la politique conjointe π^j dans l'environnement d afin de générer des historiques complets de transitions (ω, a, ω') sur les w_{episodes} de la fenêtre de validation. Cela permet de calculer la moyenne $\bar{r}$ et l'écart-type σ des récompenses sur les épisodes collectés. De plus, deux ensembles de données sont extraits de ces épisodes :

- $\mathcal{D}_{\text{trans}}$ contenant les séquences de transitions $(\omega_t, a_t, \omega_{t+1})$, utilisées pour l'inférence des rôles ;
- $\mathcal{D}_{\text{obs}}$ contenant uniquement les séquences d'observations (ω_t), utilisées pour l'inférence des objectifs et missions.

ÉTAPE 2 : OPTIMISATION DES DISTANCES ET SEUILS DE CLUSTERING. Pour réduire la variabilité entre trajectoires et identifier des structures récurrentes, les ensembles $\mathcal{D}_{\text{obs}}$ et $\mathcal{D}_{\text{trans}}$ sont soumis à un processus de clustering. On explore plusieurs métriques de distance δ_t (ex. LCS, Smith-Waterman, distances vectorielles) et seuils de regroupement τ_t. Chaque combinaison (δ_t, τ_t) est évaluée selon un score pondérant :

$$\text{Score} = \alpha(\sigma_{\text{obs}} + \sigma_{\text{trans}}) + \beta N_{\text{clusters}}$$

où σ désigne la variance intra-cluster et N_{clusters} le nombre total de clusters. La combinaison minimisant ce score est retenue, garantissant un compromis entre cohérence interne et compacité des clusters.

ÉTAPE 3 : APPLICATION DU CLUSTERING OPTIMAL. Une fois les hyperparamètres optimaux (δ^*, τ^*) déterminés, les trajectoires sont regroupées :

- les clusters de transitions C_{trans} permettent d'inférer les **rôles** par extraction des séquences communes (*Classification token* – CLS) et des règles comportementales associées ;
- les clusters d'observations C_{obs} servent à identifier les **objectifs intermédiaires** et les plans associés.

Algorithme 4 : Vue algorithmique de l'activité d'analyse (Auto-TEMM)

Input : Politique conjointe entraînée π^j ; ODec-POMDP d ; Spécification initiale $\mathcal{MM}$; Seuil de convergence normalisé (défaut : 3.5%) η

Output : Spécification organisationnelle inférée $\mathcal{MM}_{\text{inferred}}$; Score de adéquation organisationnel OF

`// 1. Collecte des trajectoires`

Générer les historiques individuels $\mathcal{D}_{\text{trans}}$ depuis d sous π^j

$\mathcal{D}_{\text{obs}} \leftarrow$ trajectoires d'observations individuelles issues de $\mathcal{D}_{\text{full}}$

`// 2. HPO sur distance et seuil de clustering`

for $t \in \{obs, trans\}$ **do**

  foreach *métrique de distance* δ_t **do**

    foreach *seuil minimal de cluster* τ_t **do**

      Appliquer clustering avec (δ_t, τ_t)

      Calculer $\sigma_{\text{obs}}, \sigma_{\text{trans}}, N_{\text{clusters}}$

      Score $\leftarrow \alpha(\sigma_{\text{obs}} + \sigma_{\text{trans}}) + \beta N_{\text{clusters}}$ `// par défaut :` $\alpha = 0.4,\ \beta = 0.6$

      Retenir (δ_t^*, τ_t^*) avec Score minimal

`// 3. Application du clustering avec HPO optimal`

Clustering des observations : $\mathcal{D}_{\text{obs}} \rightarrow C_{obs}$ via $(\delta^*_{obs}, \tau^*_{obs})$

Clustering des transitions : $\mathcal{D}_{\text{trans}} \rightarrow C_{trans}$ via $(\delta^*_{trans}, \tau^*_{trans})$

`// 4. HPO sur la représentativité (convergence)`

for $t \in \{obs, trans\}$ **do**

  foreach *représentativité* ρ_t **do**

    Inférer $\mathcal{MM}_{\rho_t}$ à partir des clusters

    Initialiser une politique $\pi^j_{\rho_t}$

    Entraîner $\pi^j_{\rho_t}$ sur $(d, \mathcal{MM}_{\rho_t})$ jusqu'à atteindre $R_{\min}$

    Enregistrer le temps de convergence c_{ρ_t} tel que $ct_t(\rho_t) = c_{\rho_t}$

  `// Sélectionner le point de coude`

  $\rho_t^* \leftarrow max(\{\rho_t \mid ct_t(\rho_t) < \eta\})$ `// par défaut` $\eta = 3.5\%$

`// 5. Inférence des rôles et objectifs`

Inférer les rôles à partir de $\mathcal{D}_{\text{trans}}, \delta^*, \tau^*, \rho^*$

Inférer les objectifs à partir de $\mathcal{D}_{\text{obs}}, \delta^*, \tau^*, \rho^*$

`// 6. Calcul du adéquation organisationnel`

Calculer SOF et FOF à partir des variances intra-cluster

$OF \leftarrow \frac{1}{2}(SOF + FOF)$

`// 6.5. Raffinement manuel (optionnel)`

$(\mathcal{MM}_{\text{inferred}} \times 0, 1) \leftarrow$ `manual_refine`$(\mathcal{MM}_{\text{inferred}})$

return $\mathcal{MM}_{inferred}$, *org_fit*, $\overline{r}, \sigma$, *running_refinement*

ÉTAPE 4 : RECHERCHE DE REPRÉSENTATIVITÉ OPTIMALE. Le degré de représentativité ρ_t fixe le seuil minimal pour qu'une transition ou observation soit retenue comme caractéristique d'un rôle ou objectif. Une recherche par grille est effectuée sur différentes valeurs de ρ_t. Pour chaque ρ_t, une spécification $\mathcal{MM}_{\rho_t}$ est inférée, puis une nouvelle politique $\pi^j_{\rho_t}$ est réentraînée dans d. On enregistre alors le temps de convergence c_{ρ_t} pour atteindre une performance minimale $R_{\min}$. Le paramètre optimal ρ_t^* est choisi comme la plus grande représentativité garantissant une convergence inférieure au seuil η (par défaut 3.5%).

ÉTAPE 5 : INFÉRENCE DES RÔLES ET OBJECTIFS. Avec $(\delta^*, \tau^*, \rho^*)$, on extrait les rôles $\mathcal{R}$, les objectifs intermédiaires $\mathcal{G}$ et leurs relations hiérarchiques (missions, héritage de rôles). Les permissions et obligations sont déduites en observant la systématicité (obligations) ou la variabilité (permissions) des associations rôle–mission dans les trajectoires.

ÉTAPE 6 : CALCUL DE L'ADÉQUATION ORGANISATIONNELLE. Comme pour TEMM, deux indicateurs partiels sont calculés :

- le **SOF** (structural organizational fit), mesurant la cohérence des rôles via la variance des transitions intra-cluster ;
- le **FOF** (functional organizational fit), mesurant la cohérence fonctionnelle dans l'atteinte des objectifs intermédiaires.

L'indicateur global est obtenu par agrégation : $\mathrm{OF} = \frac{1}{2}(\mathrm{SOF} + \mathrm{FOF})$

ÉTAPE 6.5 : COMPRÉHENSION ET RAFFINEMENT DES SPÉCIFICATIONS. À cette étape, les spécifications organisationnelles implicites $\mathcal{MM}_{\text{inferred}}$ sont présentées sous forme de guides de contraintes, qui s'apparentent à des règles reliant des observations à des actions (RAG) ou à des ensembles d'observations (GRG). L'utilisateur peut, s'il le souhaite, examiner et interpréter ces guides de contraintes afin de proposer de nouvelles spécifications organisationnelles plus explicites et compréhensibles. Cette étape est facultative, mais elle permet d'améliorer la robustesse et l'interprétabilité des spécifications organisationnelles. Si l'utilisateur ne modifie pas les spécifications inférées, celles-ci sont conservées et réutilisées lors du cycle de raffinement suivant pour restreindre l'espace de recherche. Si l'utilisateur considère que les résultats sont satisfaisants, il peut mettre fin aux cycles de raffinement en positionnant le booléen `running_refinement` à 0.

Nous englobons cette étape optionnelle dans la relation `manual_refine` : $\mathcal{MM}_{\text{inferred}} \to \mathcal{MM}_{\text{explicit}} \times 0, 1$, qui prend en entrée les spécifications implicites et retourne des spécifications explicites (ou les mêmes si l'utilisateur ne souhaite pas les modifier) ainsi qu'un booléen indiquant s'il souhaite poursuivre les cycles de raffinement.

ÉTAPE 7 : SORTIES DE L'ACTIVITÉ. L'activité retourne :

- une spécification implicite $\mathcal{MM}_{\text{inferred}}$ décrivant rôles, missions, permissions et obligations inférés automatiquement ;
- le score d'adéquation organisationnelle `org_fit` permettant de quantifier la qualité organisationnelle des comportements émergents ;
- la moyenne $\bar{r}$ et l'écart-type σ des récompenses sur les épisodes analysés ;
- un booléen `running_refinement` indiquant si l'utilisateur souhaite poursuivre les cycles de raffinement.

9.4 BILAN

En synthèse, l'activité d'analyse permet d'établir un lien objectif entre les comportements émergents des agents et les structures organisationnelles attendues. Grâce à la méthode TEMM et à son extension Auto-TEMM, il devient possible d'inférer automatiquement des

rôles, objectifs et missions à partir des trajectoires, et de quantifier leur adéquation organisationnelle par un indicateur robuste. Cette démarche favorise l'explicabilité, la traçabilité et le raffinement itératif des spécifications organisationnelles, tout en fournissant des outils d'évaluation pour comparer différentes politiques ou configurations. Toutefois, la qualité de l'analyse dépend de la diversité des trajectoires collectées et du choix des hyperparamètres de clustering, même si l'automatisation par optimisation conjointe permet de limiter l'intervention humaine. De plus, bien que permissions et obligations soient inférées, leur évaluation reste aujourd'hui agrégée dans l'indicateur global : une mesure déontique dédiée constitue un axe d'amélioration direct pour renforcer C4 et C5. Cette activité constitue ainsi un pivot essentiel pour la boucle de conception MAMAD, en préparant le transfert et l'amélioration continue des politiques dans l'environnement réel.

10 TRANSFÉRER ET SUPERVISER EN ENVIRONNEMENT RÉEL

L'*activité de transfert* correspond à la mise en production et au suivi des politiques conjointes dans l'environnement réel. Elle joue un double rôle : (i) assurer l'exécution continue de la politique la plus récente π^j_{latest} dans $\mathcal{E}$, garantissant l'action efficace des agents, et (ii) collecter de nouvelles trajectoires réelles $(\omega^j_t, a^j_t, \omega^j_{t+1})$ pour enrichir la base de données $\mathcal{D}_{H^j}$, permettant la mise à jour du modèle simulé et des spécifications organisationnelles.

OBJECTIFS FORMELS

Les **entrées** de l'activité de transfert sont :

- la politique conjointe la plus récente π^j_{latest} ;
- l'environnement réel $\mathcal{E}$;
- la base de trajectoires accumulées $\mathcal{D}_{H^j}$.

Les **sorties attendues** sont :

- une base enrichie de trajectoires $\mathcal{D}_{H^j}$;
- un signal `need_update` déclenchant la reprise du cycle de conception.

La relation globale peut être exprimée par :

$$(\mathcal{D}_{H^j}, \texttt{need_update}) \leftarrow \texttt{transfer}(\pi^j_{\text{latest}}, \mathcal{E}, \mathcal{D}_{H^j}, \texttt{mode})$$

10.1 TRAVAUX MOBILISÉS ET VERROUS IDENTIFIÉS

L'activité de transfert et de supervision en environnement réel s'appuie sur plusieurs axes : le transfert de politiques (policy transfer), l'adaptation de domaine (Sim2Real), la calibration dynamique des modèles (online model calibration), et la supervision continue des SMAs.

Les approches de *Robust Reinforcement Learning* [123] visent à rendre les politiques résistantes aux écarts simulation/réalité, mais n'intègrent pas la mise à jour du modèle simulé après déploiement. Les méthodes d'adaptation de domaine et *Sim2Real* [126, 130] réduisent l'écart simulation/réel via la randomisation ou l'apprentissage de représentations invariantes, mais leur adaptation en ligne reste limitée. Les techniques de calibration dynamique [148] mettent à jour le modèle simulé à partir des retours du réel, sans prise en compte explicite de l'adaptation des politiques multi-agents. Enfin, la synchronisation manuelle reste courante, mais peu adaptée aux environnements dynamiques.

Les principaux verrous sont :

- l'absence de cadre unifié pour la mise à jour conjointe du modèle simulé et des politiques déployées ;
- la difficulté à détecter et corriger automatiquement les écarts simulation/réalité ;

- le manque de mécanismes intégrés pour garantir robustesse et sécurité lors du transfert ;
- la nécessité d'une supervision continue et automatisée.

Ces limites motivent le développement d'un cadre méthodologique assurant l'adaptation conjointe du jumeau numérique et des politiques multi-agents, avec supervision automatisée du transfert.

10.2 POSITIONNEMENT ET CONTRIBUTIONS PROPOSÉES

L'approche proposée introduit un **cadre de transfert asynchrone et événementiel** comprenant :

- une **Boucle de transfert** (`Transfer Loop`) qui assure l'exécution en continu de la politique et la collecte des trajectoires dans un tampon temporaire $\mathcal{B}$;
- un **Déclencheur de mise à jour** (`Update Trigger`) qui ajoute les trajectoires à la base $\mathcal{D}_{\text{H}^j}$ et active, de façon asynchrone, les activités de modélisation et d'entraînement dès qu'un seuil `batch_size` est atteint.

Ce double mécanisme assure la continuité du fonctionnement des agents, tout en maintenant la boucle de conception synchronisée avec les données réelles.

10.3 DESCRIPTION ET MISE EN ŒUVRE DE L'ACTIVITÉ

L'Algorithme 5 formalise le fonctionnement de l'activité de transfert. Chaque élément est décrit ci-dessous afin de préciser les mécanismes et leur rôle dans la boucle de conception.

Schéma opératoire du transfert

Pour clarifier la lecture de l'Algorithme 5, le flux de contrôle est le suivant :

1. exécuter la politique courante en mode `DIRECT` ou `REMOTE` ;
2. stocker les transitions dans un tampon temporaire $\mathcal{B}$;
3. tester le seuil `batch_size` ;
4. en cas de seuil atteint, fusionner $\mathcal{B}$ dans $\mathcal{D}_{\text{H}^j}$ puis déclencher une mise à jour asynchrone ;
5. reprendre immédiatement la boucle de transfert avec la dernière politique disponible.

Ce schéma distingue explicitement le plan **d'exécution en ligne** (déploiement) et le plan **de mise à jour hors-ligne** (relance MTA), ce qui réduit l'ambiguïté sur la synchronisation.

ENTRÉES ET SORTIES. L'activité reçoit en entrée :

- la politique conjointe la plus récente π^{j}_{latest}, produite lors de l'entraînement ;
- l'environnement réel $\mathcal{E}$, représentant le domaine opérationnel où le SMA agit ;

Algorithme 5 : Vue algorithmique de l'activité de transfert

Input : Politique actuelle π^j_{latest}, environnement réel $\mathcal{E}$, base de trajectoires $\mathcal{D}_{H^j}$, mode de transfert `mode` $\in \{$`DIRECT`, `REMOTE`$\}$

Output : Base de trajectoires mise à jour $\mathcal{D}_{H^j}$, signal de mise à jour `need_update`

Procédure BoucleDeTransfert
  while *le SMA est actif dans l'environnement* $\mathcal{E}$ **do**
    if *mode* $=$ *DIRECT* **then**
      `// Chaque agent applique localement la politique`
      Les agents observent localement ω^j_t
      Calcul local $a^j_t \leftarrow \pi^j_{\text{latest}}(\omega^j_t)$
      Application locale des actions et mise à jour de l'état ω^j_{t+1}
      Stockage local $(\omega^j_t, a^j_t, \omega^j_{t+1})$ dans un tampon interne
      Périodiquement : envoi des tampons locaux à CybMASDE pour mise à jour de $\mathcal{D}_{H^j}$
    else if *mode* $=$ *REMOTE* **then**
      `// La politique est exécutée par le processus Transferring`
      $\omega^j_t \leftarrow$ `observe`$(\mathcal{E})$
      $a^j_t \leftarrow \pi^j_{\text{latest}}(\omega^j_t)$
      $\omega^j_{t+1} \leftarrow$ `apply`$(\mathcal{E}, a^j_t)$
      Ajouter $(\omega^j_t, a^j_t, \omega^j_{t+1})$ au tampon temporaire $\mathcal{B}$
    `// Vérification du déclenchement de la mise à jour`
    if $|\mathcal{B}| \geqslant$ *batch_size* **then**
      Ajouter $\mathcal{B}$ à $\mathcal{D}_{H^j}$ et vider $\mathcal{B}$
      `need_update` $\leftarrow$ `True`
      if *not running_update* $=$ *True* **then**
        `launch_update()` `// Appel asynchrone du processus MTA`

- la base courante de trajectoires $\mathcal{D}_{H^j}$, enrichie au fil des déploiements.

En sortie, elle retourne :

- une base de trajectoires mise à jour $\mathcal{D}_{H^j}$;
- un signal booléen `need_update` indiquant si les activités de modélisation et d'entraînement doivent être relancées.

BOUCLE DE TRANSFERT. La boucle de transfert s'exécute tant que le SMA est actif dans $\mathcal{E}$. À chaque pas de temps t :

1. une observation ω^j_t est collectée via la fonction `observe`$(\mathcal{E})$;
2. l'action a^j_t est choisie en appliquant la politique π^j_{latest} à l'observation courante ;
3. cette action est exécutée dans l'environnement via `apply`$(\mathcal{E}, a^j_t)$, produisant la nouvelle observation ω^j_{t+1} ;
4. la transition $(\omega^j_t, a^j_t, \omega^j_{t+1})$ est stockée dans un tampon temporaire $\mathcal{B}$.

DÉCLENCHEUR DE MISE À JOUR. Lorsque la taille du tampon $\mathcal{B}$ dépasse un seuil `batch_size`, le contenu est ajouté à la base de trajectoires $\mathcal{D}_{\text{H}^j}$ puis le tampon est vidé. Le signal `need_update` est alors activé. Si aucun processus de mise à jour n'est en cours (`not running_update`), la procédure `launch_update()` est déclenchée de manière asynchrone pour relancer les activités de modélisation et d'entraînement.

FONCTIONNEMENT GLOBAL. Ce schéma assure trois propriétés essentielles :

- la **continuité d'exécution** : les agents opèrent toujours avec la dernière politique disponible ;
- la **réactivité** : les données réelles sont intégrées dès qu'un volume suffisant est collecté ;
- la **automatisation** : les mises à jour se déclenchent sans intervention humaine, tout en évitant les conflits entre processus parallèles.

ÉLÉMENTS FORMELS.

- $\mathcal{B}$ désigne le tampon temporaire (voir Algorithme 5),
- `batch_size` fixe la granularité de déclenchement des mises à jour,
- `launch_update`() assure la synchronisation avec les autres activités de la méthode.

10.4 BILAN

En synthèse, l'activité de transfert assure l'exécution continue de la politique la plus récente et la collecte automatisée de trajectoires réelles pour raffiner le modèle et réentraîner les politiques. Ses atouts sont :

- automatisation du déploiement et de la collecte en environnement réel ;
- synchronisation robuste avec les autres activités de MAMAD ;
- adaptation continue des agents à l'environnement.

Ses limites portent sur le coût de supervision, la gestion des environnements critiques, et la fréquence optimale des mises à jour. Cette activité propose une méthode **autoadaptative**, alternant apprentissage simulé et déploiement réel pour garantir la robustesse des SMA en contexte dynamique.

CONCLUSION

Cette troisième partie a introduit la méthode **MAMAD** comme une réponse concrète aux limites identifiées dans les approches actuelles de conception de SMA. Reposant sur un cycle itératif structuré en quatre activités (*Modélisation*, *Apprentissage*, *Analyse*, *Transfert*), MAMAD articule de manière cohérente des outils symboliques (spécifications organisationnelles) et MARL pour guider la conception, l'entraînement et l'adaptation d'agents intelligents dans des environnements complexes.

La méthode s'appuie notamment :

- sur une modélisation fidèle des environnements à partir de données empiriques,
- sur un entraînement contraint par des spécifications organisationnelles intégrées au sein du cadre *MOISE+MARL*,
- sur une analyse des trajectoires pour inférer les structures émergentes de l'organisation apprise,
- et enfin sur un transfert contrôlé permettant l'amélioration itérative du SMA.

Dans la partie suivante, nous proposons de valider expérimentalement cette méthode à travers son implémentation concrète dans un outil dédié, CybMASDE, et son application à plusieurs environnements représentatifs. L'objectif est de démontrer la capacité de MAMAD à produire automatiquement des SMAs performants, explicables et conformes à des exigences organisationnelles dans des contextes variés.

Nous évaluons notamment la méthode selon des critères d'efficacité, d'automatisation, de conformité aux contraintes et d'explicabilité, tout en comparant ses résultats à ceux d'approches classiques non guidées par des modèles organisationnels.

La Partie IV met donc à l'épreuve la méthode MAMAD, en analysant ses performances et sa pertinence au regard des verrous identifiés sur différents scénarios.

Quatrième partie

VALIDATION EXPÉRIMENTALE DE LA MÉTHODE

INTRODUCTION

Cette partie vise à montrer l'applicabilité et la pertinence de la méthode MAMAD dans l'ensemble de ses activités ou une partie d'entre elles dans différents contextes de conception de SMAs. Pour cela, nous avons développé une plateforme dédiée, CybMASDE, qui implémente l'ensemble du pipeline proposé (modélisation, apprentissage, analyse, transfert) de manière modulaire et reproductible.

Dans un premier temps, nous décrivons en détail l'environnement expérimental, les outils logiciels et matériels mobilisés, ainsi que les environnements de test retenus. Nous présentons également les spécifications organisationnelles associées à chaque environnement, ainsi que les métriques d'évaluation permettant de valider les performances de la méthode. La Figure 22 illustre l'organisation de cette partie.

Dans un second temps, nous analysons les résultats obtenus afin de répondre aux objectifs de recherche identifiés dans la partie précédente. Cela inclut une évaluation de l'efficacité de la méthode, de sa capacité d'automatisation, de l'adéquation des politiques apprises avec les contraintes organisationnelles, ainsi que de leur explicabilité.

Cette étude expérimentale nous permettra de mieux cerner les atouts et les limites de la méthode MAMAD, et de dégager des perspectives d'amélioration pour une automatisation encore plus poussée de la conception organisationnelle en MARL.

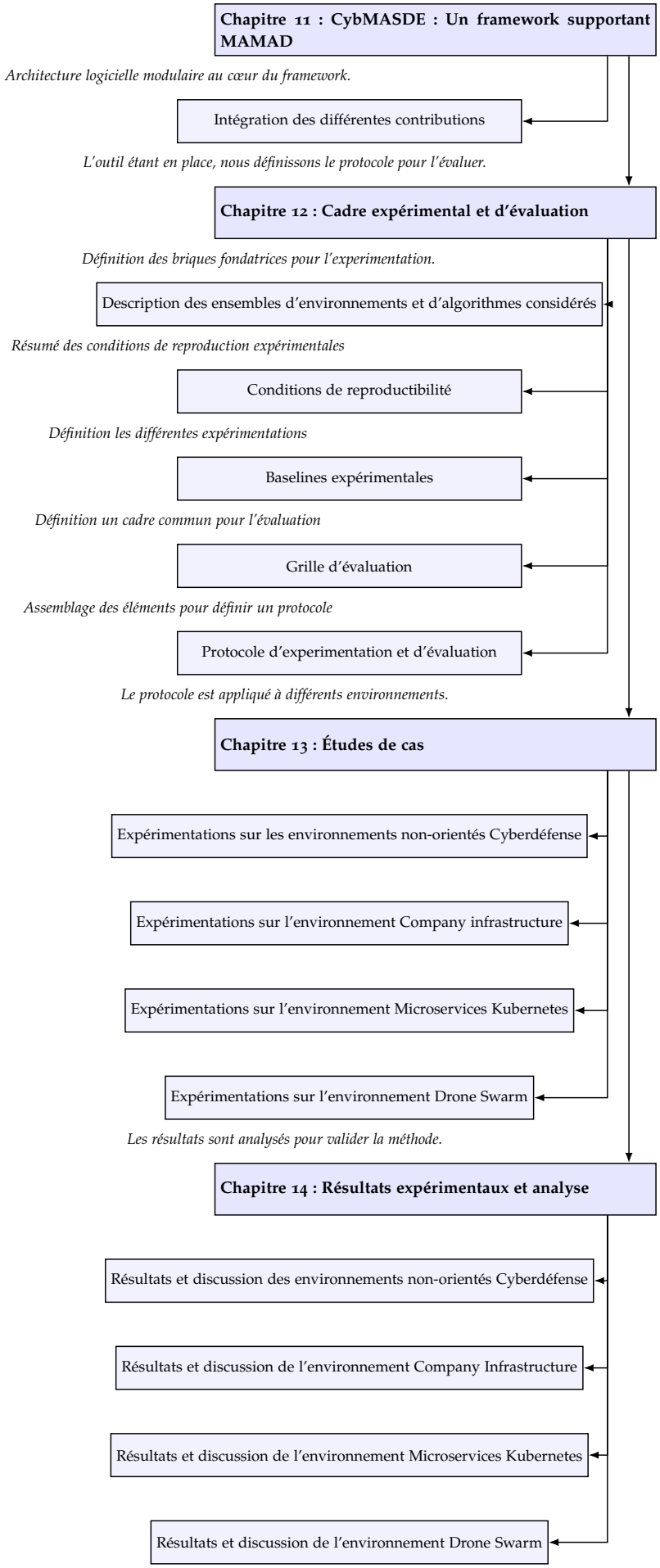


Figure 22 : Structure de la Partie IV : Cadre expérimental et analyse des résultats

11 CYBMASDE : UN FRAMEWORK SUPPORTANT MAMAD

CybMASDE[1] est une plateforme modulaire et extensible que nous proposons pour supporter la méthode MAMAD. Elle intègre l'ensemble des implémentations des contributions proposées dans la méthode, afin de répondre aux différentes attentes du concepteur. Elle lui permet de modéliser l'environnement cible, de déterminer des politiques efficaces, puis de les analyser pour inférer des spécifications organisationnelles compréhensibles en alternant entre entraînement et analyse. En parallèle, CybMASDE permet de maintenir la cohérence entre l'environnement simulé et l'environnement réel, en mettant à jour à la fois le modèle simulé et les politiques déployées. CybMASDE est utilisable principalement en mode *Command Line Interface* (CLI) avec la commande "cybmasde" (voir le manuel en Annexe B.3), mais propose également une interface graphique pour effectuer les principales tâches de configuration.

11.1 FONCTIONNALITÉS PROPOSÉES PAR `cybmasde`

Cette section décrit le parcours type d'un utilisateur de CybMASDE, depuis la modélisation de l'environnement jusqu'à l'obtention d'une politique conjointe satisfaisante accompagnée de mesures de performance, stabilité et spécifications organisationnelles explicatives. Chaque étape indique à la fois les actions de l'utilisateur et les processus internes de la plateforme.

En général, le **Concepteur** peut activer n'importe quelle fonctionnalité tant que les dépendances entre les fonctionnalités sont respectées. Par exemple, il est possible de lancer l'entraînement si l'environnement a été préalablement modélisé manuellement.

Bien que l'**Concepteur** représente le rôle "général", il joue en réalité des rôles différents lorsqu'il interagit avec CybMASDE :

- **Modélisateur de l'environnement**, chargé de définir ou compléter la description de l'environnement simulé, soit automatiquement à l'aide d'un **World Model**, soit manuellement par le biais du modèle *Multi-Cyberdefense Agent Simulator* (MCAS) ;
- **"Entraîneur" d'agents**, qui conduit les phases d'apprentissage des politiques multi-agents en tenant compte des spécifications organisationnelles ;
- **Analyste organisationnel**, qui exploite la méthode TEMM ou Auto-TEMM afin d'inférer des rôles et objectifs organisationnels à partir des trajectoires produites ;
- **Raffineur de spécifications**, qui combine les deux rôles précédents pour réaliser des cycles itératifs d'entraînement et d'analyse jusqu'à l'obtention d'une politique satisfaisante ;
- **Opérateur de déploiement**, qui supervise l'intégration de la politique conjointe finale dans l'environnement réel, soit en mode *DIRECT* (politique exécutée localement par les agents), soit en mode *REMOTE* (politique exécutée par le processus *Transferring*).

En pratique, CybMASDE est conçu comme un outil principalement utilisable en mode CLI : chaque étape du cycle MAMAD (Modelling, Training, Analyzing, Transferring) est exposée à travers une commande explicite. Après avoir créé un nouveau projet, CybMASDE génère automatiquement une arborescence de dossiers structurée autour des quatre activités principales de la méthode MAMAD (modélisation, entraînement, analyse, transfert), ainsi qu'un fichier central de configuration `project_configuration.json`. Un exemple détaillé de ce fichier pour l'environnement Overcooked-AI est donné en Table 15.

TABLE 15 : Résumé du fichier "project_configuration.json" avec exemples sur Overcooked-AI

Clé / Élément	**Rôle dans CybMASDE**	**Exemple (Overcooked-AI)**
"common . project_name"	Nom du projet	""Overcooked_coop""
"common . project_description"	Brève description du projet	""Test coopération à 2 agents""
"common . label_manager"	Fichier Python mappant actions/observations en étiquettes	Associer "0→move_north", "1→pickup_onion"
"common . project_path"	Chemin absolu du projet	"/home/user/Overcooked_coop"
"modelling . simulated_environment . environment_path"	Environnement manuel (MCAS) si utilisé	Vide si World Model uniquement
"modelling . generated_environment . world_model . jopm . autoencoder"	Encodeur d'observations (VAE)	VAE compressant la cuisine en vecteur latent (dim=32)
"modelling . generated_environment . world_model . jopm . rdlm"	Modèle dynamique (RNN+MLP)	Prédit transition après action "pickup_onion"
"modelling . generated_environment . world_model . initial_joint_observation"	Observations initiales	Positions des 2 cuisiniers et des ingrédients
"modelling . generated_environment . component_functions_path"	Fonctions "reward()", "stop()", "render()"	"reward=+1" si soupe servie ; "stop" après 400 pas
"modelling . organizational_specifications"	Spécifications MOISE+MARL	Rôles "Chef1", "Chef2", missions : ramasser, servir
"training . hyperparameters"	Paramètres MARL	"lr=0 . 0003", "gamma=0 . 95", batch=128
"training . statistics"	Résultats d'entraînement	Récompense moyenne par épisode
"training . joint_policy"	Dernier checkpoint de la politique conjointe	"policy_epoch200 . pth"
"analyzing . hyperparameters"	Paramètres Auto-TEMM	Distance DTW, seuil représentativité=0 . 3
"analyzing . statistics"	Résultats quantitatifs	Variance des récompenses = 0 . 12
"analyzing . figures_path"	Graphiques produits	Dendrogrammes de rôles, courbes de convergence
"analyzing . post_training_trajectories_path"	Trajectoires utilisées pour l'analyse	Séquence d'actions collectives pour une soupe
"analyzing . inferred_organizational_specifications"	Spécifications MOISE+MARL inférées	Agent A spécialisé en ramassage, Agent B en service
"transferring . last_checkpoint"	Politique finale à déployer	"policy_final . pth"
"transferring . configuration"	Paramètres de transfert (mode, API)	""mode" : "REMOTE", "api_url" : "http ://localhost :8000""

Le principe général de CybMASDE est que l'utilisateur doit ainsi investir un effort initial important pour configurer l'ensemble des paramètres de son projet, centralisés dans le fichier `project_configuration.json`. Le concepteur renseigne alors les informations nécessaires pour chaque activité, principalement sous forme de code Python, JSON ou en indiquant les chemins vers des fichiers existants. Par exemple, pour la modélisation, l'utilisateur peut fournir le chemin d'une simulation existante, ou choisir d'en créer une nouvelle en définissant l'espace des observations, des actions, les fonctions de récompense, d'arrêt et éventuellement de rendu basées sur l'historique, ainsi que les hyperparamètres pour l'entraînement du World Model. Une fois ce travail préparatoire réalisé, l'ensemble

du pipeline devient automatisable et ne nécessite plus nécessairement d'interventions manuelles excepté pour les cycles de raffinement.

Le parcours type des lignes de commande entrées est généralement le suivant :

1. **`init`** : crée l'arborescence du projet et un fichier de configuration vierge. L'utilisateur complète ensuite ce fichier en renseignant les espaces d'observations et d'actions, les fonctions de récompense et d'arrêt, les spécifications organisationnelles et les hyperparamètres d'entraînement. Une étape requise pour la configuration est d'implémenter une **API Environnementale** en implémentant l'interface "environment_api" qui va permettre à CybMASDE de communiquer avec l'environnement cible (voir Annexe B.2).

2. **`validate`** : vérifie la complétude et la cohérence du projet. En cas d'erreurs (fichier manquant, fonction non implémentée, API invalide), l'exécution est stoppée avec un message explicite.

3. **`model`** : modélise un environnement simulé à partir de l'environnement réel. Deux variantes existent : `model -auto`, qui déclenche la génération d'un World Model à partir de traces, et `model -manual`, qui charge une instance MCAS renseignée par l'utilisateur.

4. **`train -algo <alg>`** : entraîne des politiques multi-agents à l'aide des algorithmes disponibles dans MARLlib (MAPPO, MADDPG, QMix, etc.). L'entraînement applique automatiquement les contraintes organisationnelles définies dans le projet.

5. **`analyze -auto-temm`** : applique la méthode Auto-TEMM pour extraire des spécifications organisationnelles (rôles, objectifs) et des métriques d'adéquation organisationnelle. Les résultats sont sauvegardés dans le dossier `analyzing/`.

6. **`refine -max <N>`** : lance une boucle itérative combinant entraînement et analyse, jusqu'à obtention d'une politique conjointe satisfaisante (seuils de récompense et de stabilité atteints, ou arrêt manuel par l'utilisateur).

7. **`deploy`** : déploie la politique conjointe validée dans l'environnement réel, soit en mode `-direct` (agents exécutent la politique en autonomie), soit en mode `-remote` (le processus *Transferring* exécute la politique et envoie les actions).

8. **Commandes utilitaires** : `run -full-auto` (exécution complète du pipeline sans interruption), `run -manual` (exécution étape par étape), `status` (suivi d'un projet en cours), `export -format <fmt>` (export des résultats et métriques), `clean -all` (réinitialisation de l'environnement de travail).

EXEMPLE TYPE D'UTILISATION EN MODE AUTOMATIQUE. Un scénario fréquent d'utilisation de CybMASDE consiste à exécuter l'ensemble du pipeline **MTA+T** (*Modelling*, *Training*, *Analyzing*, *Transferring*) en mode entièrement automatisé, par exemple lors d'expérimentations reproductibles sur un cluster de calcul. Dans ce cas, une seule commande en ligne suffit à orchestrer toutes les étapes, sans interaction humaine intermédiaire, comme illustrée ci-dessous :

```
cybmasde run \
  --full-auto \                          # pipeline complet (MTA+T) sans
    interaction
```

```
--project /home/john/Documents/new_test \ # chemin vers le projet
--config /home/john/Documents/new_test/project_configuration.json \ # fichier
  de config
--max-refine 10 \                         # nombre maximal d'iterations de
  raffinement
--reward-threshold 3.5 \                  # seuil de performance
--std-threshold 0.05 \                    # seuil de stabilite (ecart-type)
--accept-inferred \                       # accepter specs org. inferees
  automatiquement
--skip-model \                            # eviter de relancer la modelisation
--skip-analyze                            # sauter l'analyse si la recompense
  est suffisante
```

Listing 1 : Exécution complète de CybMASDE en mode full-auto

Cet exemple correspond à une configuration classique : l'utilisateur fixe un *seuil de récompense* pour valider les politiques conjointes, un *nombre maximal d'itérations de raffinement* pour améliorer la stabilité, et active l'option `-accept-inferred` afin d'intégrer automatiquement les spécifications organisationnelles déduites par Auto-TEMM. En pratique, cette commande illustre le mode **batch/HPC**, utilisé pour les campagnes expérimentales de grande ampleur, car elle garantit une exécution continue du pipeline, du traitement des traces initiales jusqu'au déploiement d'une politique conjointe finalisée.

À noter que CybMASDE fournit également une interface graphique moins configurable, mais permettant de configurer le projet de façon accessible et également d'exécuter les différentes activités. Cette interface est décrite plus en détail dans Annexe B.1.

11.2 CYCLE DE VIE DE CYBMASDE

Nous détaillons ci-dessous des ensembles d'échanges entre les différentes entités constituant des étapes qui incluent également les processus internes et les interactions avec l'utilisateur (voir Figure 23).

1. CONFIGURATION INITIALE ENTRE L'UTILISATEUR, L'ENVIRONNEMENT RÉEL ET CYBMASDE L'utilisateur commence par créer un nouveau projet avec la commande `cybmasde init <project_name>`, qui génère l'arborescence de dossiers et les gabarits de fichiers nécessaires. Dans ce nouveau projet, l'utilisateur doit compléter le fichier de configuration `project_configuration.json` en renseignant les espaces d'actions et d'observations, les fonctions `reward()` et `stop()`, les spécifications organisationnelles (rôles, objectifs, missions), ainsi que les hyperparamètres pour l'entraînement du World Model et des politiques multi-agents. Il doit également implémenter une API environnementale pour permettre à CybMASDE de communiquer avec l'environnement cible. Une fois, la configuration terminée, l'utilisateur peut valider le projet avec la commande `cybmasde validate`, qui vérifie la complétude et la cohérence des fichiers. Si des erreurs sont détectées, l'exécution est interrompue avec un message explicite.

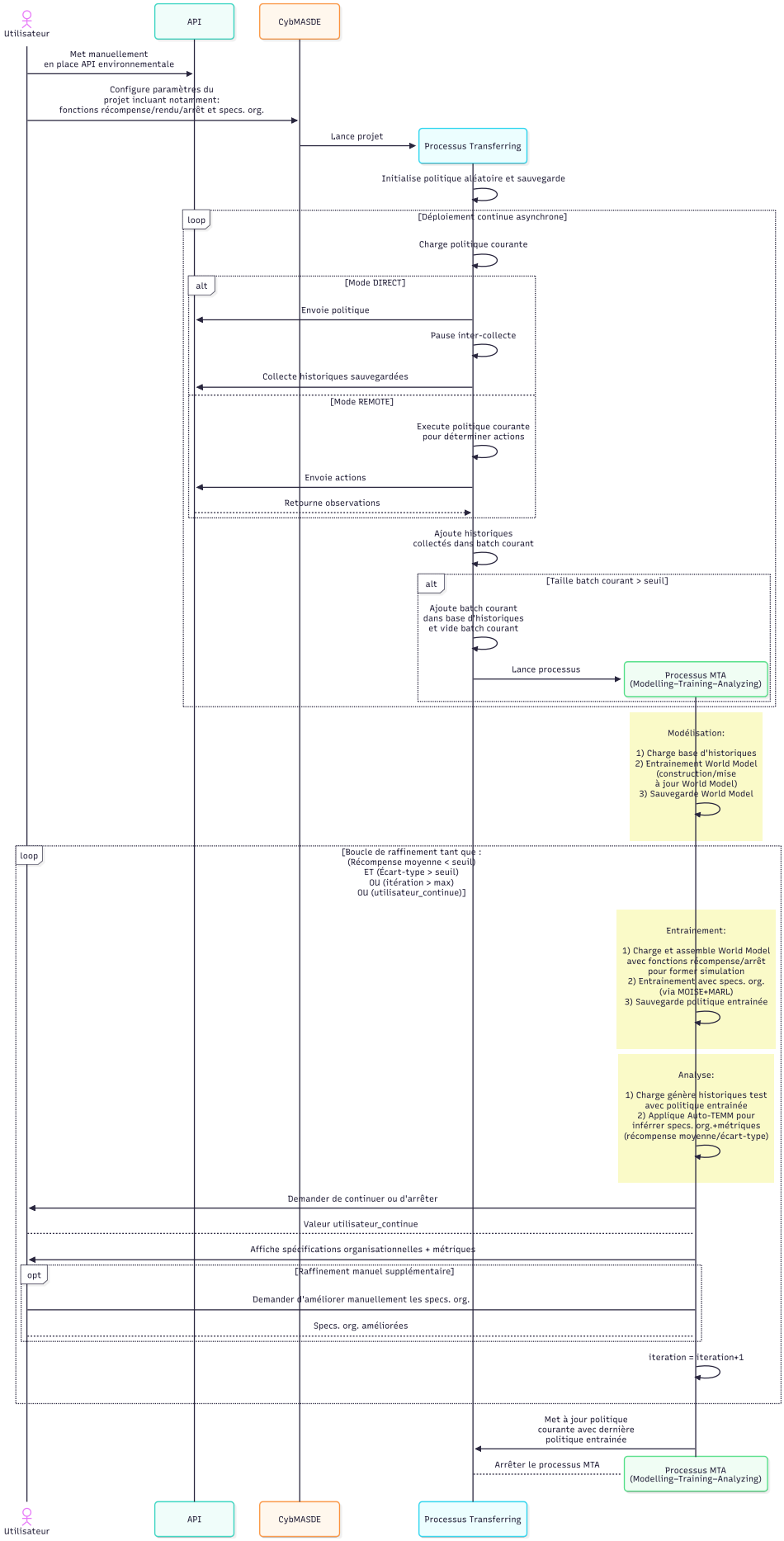


FIGURE 23 : Diagramme de séquence pour une utilisation de CybMASDE

2. PROCESSUS *transferring* Si la configuration du projet est correcte, la pipeline complète est lancée en mode automatique avec la commande `cybmasde run -full-auto`. Cela a pour effet, de créer et d'exécuter le processus *Transferring* qui est un processus exécuté en continu et indépendamment des autres. Pour la première exécution, ce processus initialise une politique aléatoire et la sauvegarde localement dans le dossier du projet ainsi qu'un batch d'historiques conjoints vide.

Ensuite, l'exécution de *Transferring* entre dans une boucle exécutée de façon asynchrone qui permet la mise à jour des politiques déployées ainsi que le modèle de simulation. Dans cette boucle, *Transferring* charge la dernière politique entraînée et la déploie dans l'environnement réel en utilisant l'API environnementale selon deux modes possibles :

- en mode *DIRECT*, la politique est envoyée directement aux agents qui l'exécutent localement stockant leurs propres historiques qui sont récupérés et concaténés dans le batch courant à interval de temps régulier via l'API environnementale ;
- en mode *REMOTE*, la politique est exécutée côté CybMASDE, les actions sont envoyées via l'API environnementale, les observations sont reçues et de nouveaux historiques sont ainsi formés côté CybMASDE et stockés dans le batch courant.

Toujours dans cette boucle, si le batch courant des historiques conjoints stockés dépasse un seuil en nombre d'historiques conjoints (défini dans la configuration), alors il est sauvegardé dans la base d'historiques (qui est un simple dossier contenant les historiques conjoints au format JSON) et vidé. Cette sauvegarde déclenche automatiquement le processus *MTA*.

3. PROCESSUS MTA : MODELLING–TRAINING–ANALYZING Le processus *MTA* est donc déclenché ponctuellement à chaque fois qu'un batch de nouveaux historiques conjoints est ajouté afin de prendre en compte les évolutions de l'environnement afin de générer des politiques adaptées. Ce processus est indépendant de *Transferring* et exécute séquentiellement les trois activités principales : *Modelling*, *Training* et *Analyzing*. Chaque étape utilise les données et configurations fournies par l'utilisateur pour accomplir sa tâche spécifique.

D'abord, dans l'activité de modélisation, si un environnement simulé *PettingZoo* a déjà été fourni dans les paramètres du fichier de configuration, il est chargé. Sinon, un environnement est généré en entraînant un **World Model** à partir des dernières données enregistrées. Dans ce dernier cas, le dernier **World Model** enregistré est chargé (s'il n'existe pas encore, alors un nouveau **World Model** est créé). Ensuite, l'ensemble des historiques conjoints enregistrés dans la base est chargé pour entraîner le **World Model** en utilisant les hyperparamètres fournis dans le fichier de configuration. Une fois, l'entraînement (ou mise à jour) terminé, le **World Model** est ensuite sauvegardé localement. Un environnement simulé *PettingZoo* est alors créé à partir du **World Model** entraîné et des informations fournies manuellement par l'utilisateur dans la configuration : les observations conjointes initiales, l'espace des actions/observations, les fonctions de récompense, de rendu (optionnel) et d'arrêt.

Boucle de raffinement

Ensuite, le processus *MTA* rentre dans une boucle de raffinement qui ne s'arrête que lorsque la politique conjointe apprise atteint un certain niveau de performance et de stabilité, ou lorsque le nombre maximal d'itérations est atteint, ou lorsque l'utilisateur décide d'arrêter le processus. Dans cette boucle, la politique conjointe est améliorée à chaque itération en alternant entre les activités d'entraînement et d'analyse.

Dans l'activité d'entraînement, l'environnement simulé mis en place précédemment est chargé pour entraîner les politiques multi-agents en utilisant l'implémentation de MOISE+MARL (voir Section 11.4) pour prendre en compte les spécifications organisationnelles. En suivant les hyperparamètres définis dans la configuration, l'entraînement est effectué avec un algorithme choisi (MAPPO, MADDPG, QMix, etc.) et selon un nombre d'épisodes maximal donné. Une fois l'entraînement terminé, la politique conjointe apprise est sauvegardée localement dans le dossier du projet. Une fois l'entraînement terminé, l'activité d'analyse est lancée.

Dans l'activité d'analyse, la politique conjointe apprise est chargée et évaluée en exécutant plusieurs épisodes dans l'environnement simulé (selon la valeur de la fenêtre glissante sur les derniers épisodes). Les trajectoires résultantes sont collectées et analysées à l'aide de la méthode Auto-TEMM ou TEMM en fonction des paramètres donnés dans le fichier de configuration. Des spécifications organisationnelles implicites (rôles, objectifs) sont obtenues sous la forme de guide de contrainte (RAG et GRG principalement) ainsi que des métriques d'adéquation organisationnelle (SOF, FOF, OF). La stabilité de la politique est également évaluée en calculant la variance des récompenses obtenues sur ces épisodes ainsi que la récompense moyenne sur ces mêmes épisodes. Les résultats de l'analyse sont sauvegardés localement dans le dossier du projet. À ce niveau-là, l'utilisateur est invité à consulter les résultats de l'analyse pour décider de continuer, d'arrêter ou de raffiner manuellement les spécifications organisationnelles (par exemple, conserver les règles observation-action les plus pertinentes). Pour cela, il peut s'aider d'une série de figures (des visualisations *Principal Component Analysis* – PCA et dendrogrammes des trajectoires comprenant les centroides calculés par Auto-TEMM/TEMM).

La boucle de raffinement continue tant que la récompense moyenne n'atteint pas le seuil fixé, que l'écart-type reste supérieur au seuil de stabilité, que le nombre maximal d'itérations n'est pas atteint, ou que l'utilisateur choisit de poursuivre. Dans le cas où les cycles de raffinement sont terminés, comme pour toutes les politiques entraînées, la dernière politique conjointe apprise est également enregistrée pour être chargée par le processus *Transferring* qui continue son exécution parallèle. Le processus *MTA* s'arrête alors jusqu'à la prochaine fois qu'un nouveau batch d'historiques conjoints est ajouté à la base d'historiques conjoints.

En synthèse, CybMASDE articule deux processus parallèles : le *Transferring*, responsable du déploiement continu et de la collecte de données, et le processus *MTA*, déclenché périodiquement pour modéliser, entraîner, analyser et raffiner les politiques. Les interactions avec l'utilisateur surviennent principalement lors de la configuration initiale, du contrôle des boucles de raffinement et du déploiement final.

11.3 CYCLE DE VIE IMPLÉMENTÉ SUR OVERCOOKED-AI

La Figure 24 présente le cycle d'utilisation de CybMASDE. Afin de rendre ce processus plus concret, nous proposons ici un tutoriel détaillé pas-à-pas en prenant l'exemple de l'environnement **Overcooked-AI** [79], qui simule une cuisine coopérative où deux agents doivent préparer et servir des soupes. La Table 15 décrit la configuration de ce projet.

11.3.1 *Configuration initiale entre l'utilisateur, l'environnement réel et CybMASDE*

L'utilisateur commence par installer Overcooked-AI et exécuter une instance locale accessible via une API REST, de façon à permettre à "simuler un environnement réel" (et permettre à CybMASDE d'interagir directement avec l'environnement via l'API environnementale qui se trouve être une API REST dans un autre processus – voir Annexe B.2). Cette étape de mise en place consiste à préparer un démon qui gère les agents, l'état de la cuisine et les règles du jeu. Du côté de CybMASDE, aucun processus n'est encore activé : la plateforme attend simplement la création d'un projet et l'initialisation de sa configuration pour pouvoir enclencher le cycle complet.

Création d'un projet Une fois l'environnement prêt, l'utilisateur crée un nouveau projet avec la commande `cybmasde init -n overcooked_coop -template worldmodel`. Cette commande génère automatiquement l'arborescence de travail, avec les dossiers `modelling`, `training`, `analyzing` et `transferring`. Un fichier central `project_configuration.json` est également produit, servant de point d'entrée pour la description du projet. En parallèle, CybMASDE crée des fichiers gabarits comme le `label_manager.py` et des squelettes de fonctions de récompense ou d'arrêt, afin de guider l'utilisateur dans la complétion des éléments essentiels.

Fourniture des éléments initiaux Dans cette phase, l'utilisateur renseigne le fichier `project_configuration.json`. Il y décrit les espaces d'actions et d'observations nécessaires pour représenter l'environnement de la cuisine : déplacements des agents, ramassage et dépôt d'ingrédients, actions de cuisson et de service. Il définit ensuite une fonction `reward()` qui attribue par exemple un point lorsqu'une soupe est servie, et éventuellement une pénalité en cas de collision entre les deux agents. Un critère d'arrêt est fixé, comme une limite de 30 pas de jeu. L'utilisateur peut laisser vide le fichier `handcrafted_environment.py`, car dans ce cas, aucun modèle MCAS n'est utilisé. Une fois ces éléments renseignés, il exécute la commande `cybmasde validate` qui permet à CybMASDE de vérifier la cohérence et la complétude de la configuration. En cas d'erreur, l'exécution est interrompue avec un message explicite ; sinon, le projet est prêt à être exécuté.

11.3.2 *Processus Transferring*

Une fois une politique conjointe satisfaisante obtenue, elle est déployée dans l'environnement réel avec la commande `cybmasde deploy -remote -api http://localhost:5000/api`. Dans ce mode, la politique est exécutée par CybMASDE qui envoie les actions aux agents du jeu et reçoit leurs observations. Le mode `-direct`, qui intègre directement la politique dans les agents, reste également possible, mais n'a pas été envisagé ici. Pendant toute la durée du déploiement et du processus *MTA*, le processus *Transferring* continue de collecter de nouvelles traces qui pourront alimenter ultérieurement de nouveaux cycles de raffinement.

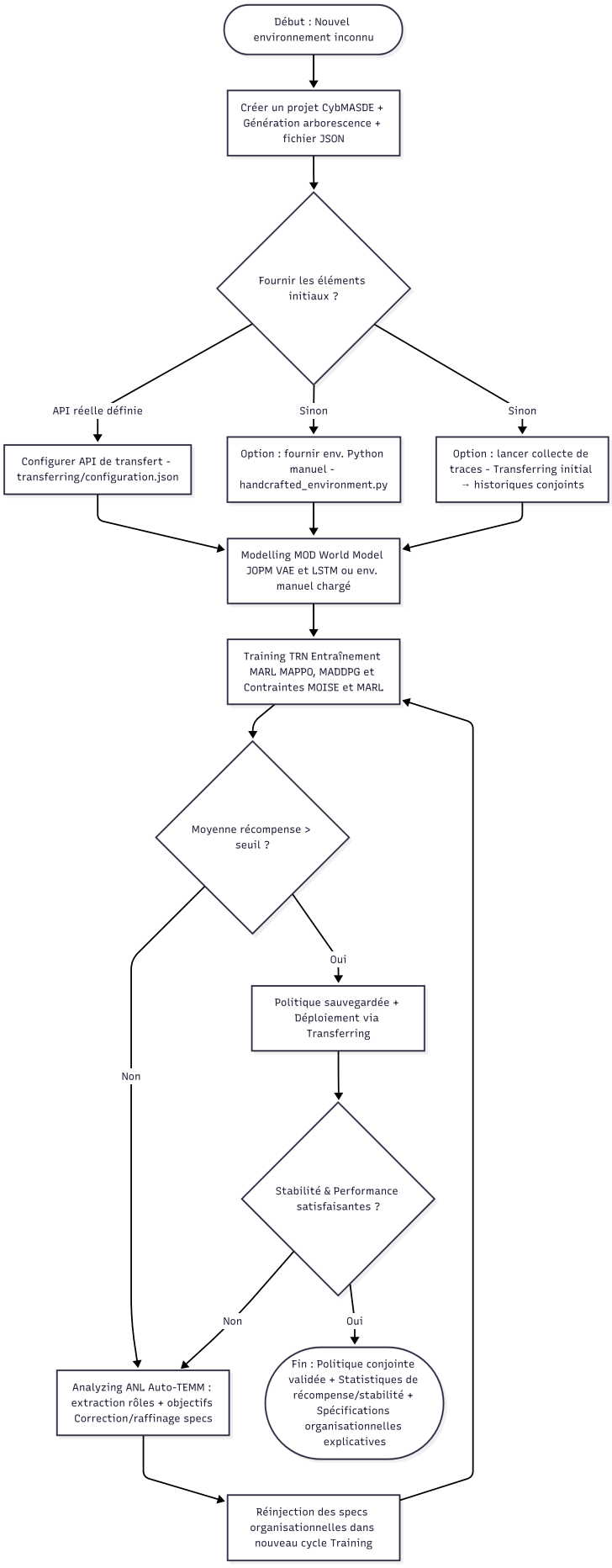


FIGURE 24 : Cycle d'utilisation de CybMASDE

11.3.3 *Processus MTA : Modelling–Training–Analyzing*

Modélisation La commande `cybmasde model -auto`, CybMASDE est lancée et procède à une première collecte de trajectoires en exécutant une politique aléatoire dans Overcooked-AI. Les deux agents se déplacent alors sans but, ramassant parfois un oignon ou occupant les mêmes cases de la cuisine. Ces données brutes servent à entraîner un **World Model** de type JOPM. Ce modèle est constitué d'un encodeur variationnel (VAE) qui compresse les observations conjointes des agents en vecteurs latents et d'un réseau RNN+MLP (RLDM) qui prédit la prochaine observation conjointe à partir de l'action exécutée. Le modèle entraîné est ensuite sauvegardé dans le dossier `generated_environment/` et constitue la base d'un environnement simulé équivalent à celui d'Overcooked-AI, utilisable sans nécessiter l'API réelle.

Entraînement Une fois le modèle simulé de l'environnement généré, les agents sont entraînés automatiquement avec la commande `cybmasde train -algo MAPPO`. CybMASDE mobilise alors `MARLlib` et `Ray RLlib` pour exécuter l'apprentissage multi-agent. Les rôles organisationnels définis par MOISE+MARL guident le processus : un agent est spécialisé dans le ramassage et la préparation des oignons (cuisinier), l'autre dans le service des soupes (serveur), et des rôles polyvalents sont également considérés (voir Listing 2). À mesure que l'entraînement progresse, des checkpoints sont sauvegardés ainsi que des courbes de récompenses qui permettent de suivre la performance au fil des épisodes.

```
organizational_model(
structural_specifications(
    roles={
        "role_server": role_logic(label_manager=oa_label_mngr).registrer_script_rule(
primary_fun),
        "role_polyvalent": role_logic(label_manager=oa_label_mngr).
registrer_script_rule(secondary_fun)},
    role_inheritance_relations={}, root_groups={}),
functional_specifications=functional_specifications(
    goals={}, social_scheme={}, mission_preferences=[]),
deontic_specifications=deontic_specifications(permissions=[], obligations=[
    deontic_specification(
        "role_server", ["agent_0"], [], time_constraint_type.ANY),
    deontic_specification(
        "role_polyvalent", ["agent_1"], [], time_constraint_type.ANY)
]))
```

Listing 2 : Extrait du fichier de configuration organisationnelle pour Overcooked-AI

Analyse Dans l'étape de configuration, l'utilisateur a fixé un seuil de performance, par exemple une récompense moyenne de 3.5 par épisode. Après l'entraînement, l'activité d'analyse CybMASDE calcule la moyenne des récompenses obtenues par les agents et la compare à ce seuil. Dans le cas d'Overcooked-AI, il n'est pas rare que les agents atteignent une moyenne inférieure après les premières itérations.

La commande `cybmasde analyze -auto-temm` applique la méthode Auto-TEMM sur les trajectoires collectées (voir Listing 3). Cette analyse consiste à regrouper les trajectoires en clusters selon leur similarité, à inférer les rôles implicites joués par les agents et à calculer des métriques de stabilité et d'adéquation organisationnelle. Dans l'environnement Overcooked-AI, on peut par exemple observer que l'agent 0 adopte systématiquement un comportement de serveur tandis que l'agent 1 assume celui de polyvalent. Les résultats de cette analyse sont sauvegardés dans le dossier `analyzing/` et peuvent être consultés par l'utilisateur (voir Listing 4).

```
Running TEMM analysis...
1. Loading trajectories...
2. Clustering trajectories...
3. Generating visualizations...
4. Computing centroids...
5. Selecting near-centroid trajectories...
6. Extracting roles and goals...
7. Summarizing roles and goals...
8. Computing organizational fit scores...
Structural Fit (SOF): 1.000
Functional Fit (FOF): 1.000
Overall Organizational Fit (OF): 1.000
Finished TEMM analysis
```

Listing 3 : Extrait de la sortie de log après entrainement et application de TEMM

```
{
    "role_2": {
        "rules": [
            {
                "observation": [
                    0.6,
                    0.4,
                    0.0,
                    0.0,
                    ...],
                "action": 3,
                "weight": 0.15
            },
            {
                "observation": [...],
                "action": [
                    0.0,
                    0.0,
                    -0.2,
                    -0.2,
                    ...],
                "weight": 0.15
            },
            {
                "observation": [...],
                "action": 5,
                "weight": 0.15
            }
        ],
        "support": 8
    },
    "role_1": {
        "rules": [
            {
                "observation": [
                    0.6,
                    0.4,
                    0.0,
                    0.0,
                    ...],
                "action": 0,
                "weight": 0.19
            },
        ],
        "support": 8
    },
    "1": {
        "rules": [
            {
                "observation": [
                    0.6,
                    0.4,
                    0.0,
                    0.0,
                    ...],
                "action": 2,
                "weight": 0.15
            },
        ],
        "support": 8
    }
}
```

Listing 4 : Extrait de spécifications organisationnelles inférées à analyser puis raffiner manuellement

Boucle de raffinement Si la performance reste insuffisante, l'utilisateur peut décider de relancer une série de cycles avec la commande `cybmasde refine -max 5 -interactive`. Dans ce mode, le système alterne automatiquement entre **entraînement** et **analyse**, en réinjectant à chaque itération les spécifications organisationnelles corrigées manuellement ou garder celles inférées par TEMM/Auto-TEMM. Dans Overcooked-AI, cela peut signifier renforcer le rôle de cuisinier pour l'un des agents, ou spécifier plusieurs rôles polyvalent pour diversifier les stratégies. Le cycle se répète jusqu'à ce que le seuil fixé soit atteint ou que le nombre maximal d'itérations soit dépassé.

Pour conclure sur cet exemple basé sur l'environnement Overcooked-AI, CybMASDE montre comment, en quelques commandes, il est possible de passer d'un environnement inconnu comme Overcooked-AI à un SMA en réduisant les interventions manuelles.

11.4 SOCLE TECHNOLOGIQUE (DÉVELOPPEMENT)

Le développement de CybMASDE repose sur un ensemble de technologies choisies pour répondre aux contraintes de performance, de modularité et d'accessibilité. Nous décrivons ci-dessous le rôle de chacune d'elles, les raisons de leur choix, ainsi que leur utilisation concrète dans le pipeline.

PYTHON (BACKEND). Le cœur de CybMASDE est écrit en **Python**, qui offre un écosystème riche en bibliothèques d'apprentissage automatique et d'orchestration scientifique. Python a été choisi car :

- il permet d'intégrer aisément des frameworks d'apprentissage multi-agents existants (MARLlib, Ray RLlib) ;
- il est adapté au prototypage rapide de modèles complexes (ex. World Models ou fonctions de récompense personnalisées) ;
- il est largement utilisé dans la communauté recherche et industrielle, ce qui favorise la reproductibilité.

Tous les modules du pipeline MTA+T (Modelling, Training, Analyzing, Transferring) sont implémentés en Python.

PYTORCH. Le choix de **PyTorch** répond à deux besoins :

1. entraîner les modèles de simulation (autoencodeurs, LSTM, VAE) pour construire les *World Models* ;
2. servir de backend commun pour les algorithmes de renforcement multi-agents.

Son API flexible permet d'implémenter des architectures de réseaux spécifiques, et son intégration avec CUDA assure un entraînement accéléré sur GPU.

MARLLIB ET RAY RLLIB. Pour l'entraînement multi-agent, CybMASDE utilise **MARLlib** adossé à **Ray RLlib**.

- **MARLlib** fournit une collection représentative d'algorithmes de référence (MAPPO, MADDPG, QMix, COMA, etc.) déjà configurés pour le multi-agent.

- **Ray RLlib** assure la scalabilité : il permet d'exécuter des entraînements distribués sur plusieurs GPU ou nœuds HPC, ce qui est indispensable pour nos expériences en cyberdéfense.

Ce couple a été choisi pour éviter de réimplémenter chaque algorithme tout en garantissant de bonnes performances sur cluster.

OPTUNA. L'optimisation des hyperparamètres est déléguée à **Optuna**, une librairie flexible qui :

- explore automatiquement l'espace des hyperparamètres (learning rate, discount factor, taille des réseaux, coefficients de clipping, etc.) ;
- intègre des stratégies d'arrêt anticipé (pruning) pour économiser du temps de calcul ;
- permet de fixer des plages d'exploration dans le fichier de configuration du projet.

L'usage d'Optuna garantit que chaque expérimentation exploite efficacement les ressources de calcul disponibles.

ANGULAR (FRONTEND). Si l'usage principal de CybMASDE reste la ligne de commande, une **interface graphique Angular** est proposée en complément. Elle vise à :

- offrir une vue unifiée de la configuration du projet (plutôt que de modifier directement l'arborescence de fichiers) ;
- faciliter le suivi des activités par des onglets dédiés (modélisation, entraînement, analyse, transfert) ;
- permettre l'injection et la visualisation de traces d'exécution ou de métriques sans coder.

Angular a été choisi car il permet de développer rapidement une interface web modulaire, moderne, et facilement extensible.

API REST ET CLI. Le pivot entre ces composants est une **API REST** interne :

- toutes les commandes CLI (`cybmasde init`, `cybmasde train`, etc.) appellent cette API ;
- l'interface Angular communique elle aussi via cette API, ce qui garantit une cohérence entre mode graphique et mode console.

Ainsi, CLI et frontend ne font qu'exposer deux modes d'accès différents au même backend.

En résumé, le socle technologique associe **Python/PyTorch** pour le calcul, **MARLlib + RLlib** pour l'apprentissage multi-agent, **Optuna** pour l'optimisation, et **Angular + REST + CLI** pour l'accessibilité.

ARCHITECTURE LOGICIELLE ET INTERACTIONS CLASSIQUES Le diagramme de déploiement des composants C4 (voir Figure 25) illustre les modules de CybMASDE et leur enchaînement lors d'une utilisation typique.

1. L'**utilisateur** interagit soit via la **CLI**, soit via l'interface **Angular**.

2. Ces deux points d'accès appellent la même **API REST** du backend.
3. L'API REST déclenche le **processus Transferring**, qui :
   - communique avec l'**API environnementale** de l'environnement cible (réel ou simulé) ;
   - génère et stocke en continu des historiques dans la base locale du projet.
4. Dès qu'un batch d'historiques est complet, le processus **MTA** est activé : il instancie successivement les modules **Modelling**, **Training** et **Analyzing**.
5. Les résultats (politiques entraînées, métriques, spécifications organisationnelles) sont stockés et rendus accessibles à l'utilisateur (via CLI ou interface).

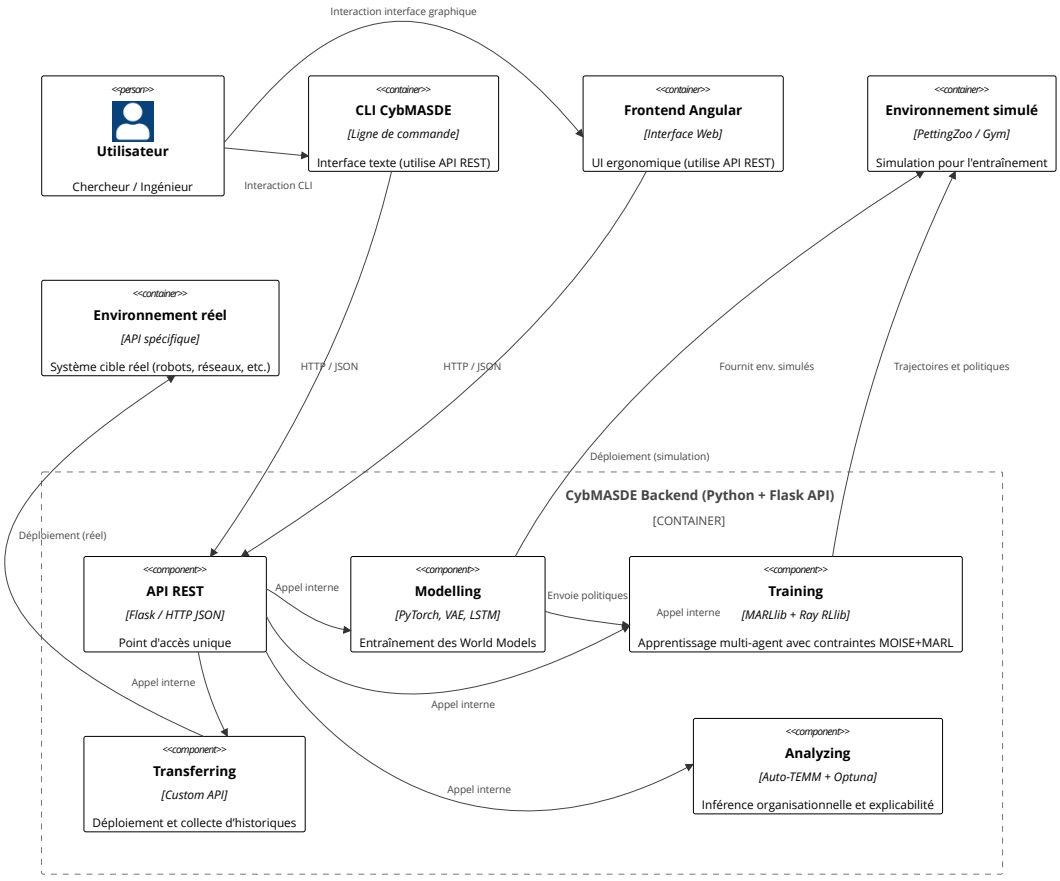


FIGURE 25 : Diagramme de composants C4 illustrant l'architecture logicielle de CybMASDE

11.5 INTÉGRATION DES DIFFÉRENTES CONTRIBUTIONS

11.5.1 *Implémentation du modèle Dec-POMDP pré-spécialisé pour la Cyberdéfense*

En complément de la modélisation automatique fondée sur des *World Models*, nous proposons une approche de modélisation manuelle guidée par un modèle pré-spécialisé pour les environnements de Cyberdéfense. Le développement du formalisme Dec-POMDP appliqué à ce domaine a conduit à un modèle que nous appelons MCAS (*Multi-Cyberdefense*

Agent Simulator). Celui-ci constitue une instance particulière de Dec-POMDP adaptée à la description explicite de scénarios de Cyberdéfense.

POURQUOI UN MODÈLE MCAS ? Dans les environnements de Cyberdéfense, il est parfois difficile de collecter suffisamment d'historiques pour alimenter un *World Model*. De plus, certains comportements défensifs sont mieux spécifiés de manière experte. MCAS permet donc :

- de donner à l'utilisateur un point d'entrée « clé en main » pour modéliser un environnement spécifique ;
- de structurer la description en respectant le formalisme Dec-POMDP (observations, actions, transitions, récompenses, arrêt) ;
- d'intégrer directement ce modèle au cycle de CybMASDE.

COMMENT EST-IL FOURNI ? Dès la création d'un projet, CybMASDE génère un fichier gabarit dans "modelling/simulated_environment/handcrafted_environment.py".

Ce fichier est une interface Python à trous que l'utilisateur doit compléter en définissant :

- l'espace d'observation (ex. : état d'un serveur ou d'un nœud du réseau),
- l'espace d'action (ex. : bloquer un trafic, scanner un port, relancer un service),
- la dynamique de transition (comment l'environnement évolue après une action),
- la fonction de récompense (bonus/malus selon la défense ou l'attaque réussie),
- la fonction d'arrêt (conditions de fin d'épisode).

EXÉCUTION ET INTÉGRATION. Une fois complété, MCAS est automatiquement reconnu par CybMASDE :

- il peut être exécuté en mode simulation pour générer des trajectoires d'entraînement multi-agent ;
- il est compatible avec le mode « tour à tour » : l'utilisateur peut visualiser en temps réel l'évolution de l'environnement, représenter la topologie sous forme de graphe et consulter les métriques (récompenses cumulées, taux de succès, etc.) ;
- il peut également être sérialisé en format JSON, afin de partager ou réutiliser la description de l'environnement.

Ainsi, MCAS offre une alternative manuelle et structurée à la modélisation automatique, garantissant que les environnements de simulation restent adaptés et compréhensibles dans des scénarios de Cyberdéfense tout en s'intégrant naturellement dans la chaîne MTA+T de CybMASDE.

11.5.2 *Implémentation du framework MOISE+MARL*

L'un des apports essentiels de CybMASDE[1] est d'intégrer le cadre organisationnel MOISE+ aux méthodes d'apprentissage multi-agents. Pour cela, nous avons développé une implémentation Python appelée *MOISE+MARL API* (MMA)[2].

POURQUOI MMA ? Le cadre MOISE+ permet de formaliser des organisations (rôles, missions, permissions), mais son usage direct peut être fastidieux. MMA vise à :

- encapsuler les relations MOISE+ sous forme de classes Python orientées objet,
- minimiser les interventions manuelles en proposant une API claire et prête à l'emploi,
- interfacer facilement avec des environnements simulés (Dec-POMDP, PettingZoo).

STRUCTURE INTERNE. L'API de MMA définit :

- une classe racine `Moise`, contenant les rôles, objectifs et permissions ;
- des classes spécialisées pour chaque type de guide de contrainte (par exemple `rag`, `rrg`, `grg`) ;
- un dictionnaire pour mapper les labels d'actions/observations à des comportements organisationnels.

Ces guides peuvent être instanciés soit par code Python, soit par règles JSON.

INTÉGRATION AVEC PETTINGZOO. Pour interagir avec un environnement multi-agent, MMA encapsule celui-ci dans un *wrapper* PettingZoo :

- les masques d'action sont appliqués automatiquement pour empêcher les agents de violer des contraintes organisationnelles,
- les récompenses sont ajustées en fonction des objectifs organisationnels (shaping),
- la cohérence entre organisation prescrite et comportements émergents est assurée pendant l'entraînement.

LIEN AVEC L'ENTRAÎNEMENT. MMA s'intègre naturellement avec **MARLlib** et **Ray RLlib** :

- l'utilisateur choisit un algorithme (MAPPO, QMix, MADDPG…) ;
- le wrapper MMA applique les contraintes MOISE+ au-dessus de l'algorithme choisi ;
- les résultats d'entraînement tiennent donc compte non seulement de la performance, mais aussi du respect des spécifications organisationnelles.

2 L'implémentation MMA, les hyperparamètres et spécifications utilisés sont disponibles à https://github.com/julien6/MOISE-MARL. Une vidéo de démonstration est également accessible à https://www.youtube.com/watch?v=b3wqFpfXZi0.

ANALYSE POST-ENTRAÎNEMENT. Après l'entraînement, CybMASDE applique la méthode TEMM (ou Auto-TEMM) pour :

- identifier automatiquement des rôles et missions implicites,
- représenter les comportements par clustering hiérarchique et K-means,
- générer des sorties visuelles (dendrogrammes de rôles, graphes de transitions),
- exporter les spécifications inférées en format JSON pour réutilisation.

En résumé, MCAS offre une modélisation manuelle spécialisée pour la Cyberdéfense, tandis que MMA fournit un cadre organisationnel opérationnel intégré au cycle d'apprentissage. Leur combinaison dans CybMASDE permet d'allier rigueur organisationnelle et flexibilité d'entraînement multi-agent.

11.6 BILAN

Ce chapitre a présenté l'implémentation de MAMAD au travers de la plateforme CybMASDE. L'approche modulaire, articulée autour des activités de modélisation, d'entraînement, d'analyse et de transfert, s'est adaptée à des scénarios variés, des environnements-jouets aux cas réels en Cyberdéfense et microservices. L'intégration de MOISE+MARL dans le pipeline d'apprentissage multi-agent a permis d'obtenir des politiques conformes aux objectifs organisationnels. CybMASDE est disponible en open-source[1].

Malgré ces apports, des limites subsistent : dépendance à la qualité des traces, besoins computationnels élevés, difficulté de généralisation à des systèmes distribués contraints, et explicabilité encore perfectible. Ce chapitre pose ainsi les bases pour une automatisation avancée de la conception organisationnelle en MARL, ouvrant la voie à des optimisations futures et à des applications sur des SMAs réels.

1 L'implémentation "CybMASDE" en plus de rassembler nos contributions, intègre un nombre important de packages conduisant à un logiciel relativement conséquent en taille (~250000 lignes de code) une fois les dépendances installées : https://github.com/julien6/CybMASDE.

12 CADRE EXPÉRIMENTAL ET D'ÉVALUATION

Le but de ce chapitre est de définir un cadre expérimental générique, applicable à tous les cas d'étude présentés dans le Chapitre 13. L'objectif est de fournir un *canevas* que chaque scénario peut instancier en précisant les éléments choisis de la méthode MAMAD. Ce cadre s'appuie sur la méthode MAMAD, la taxonomie des activités et sous-activités présentée en Table 13, et les critères d'évaluation définis précédemment.

12.1 DESCRIPTION DES ENSEMBLES D'ENVIRONNEMENTS ET ALGORITHMES CONSIDÉRÉS

Pour garantir la généralité et la robustesse de l'évaluation, nous considérons un ensemble varié d'environnements de référence issus de la littérature MARL :

- **Overcooked-AI** [79] : environnement de type jouet coopératif complexe nécessitant coordination et planification séquentielle.
- **Predator-Prey** [121] : environnement de type jouet de poursuite-évasion, utilisé pour tester la coordination et la compétition.
- **Warehouse Management** : un environnement de type jouet nouveau que nous avons proposé pour simuler la gestion logistique multi-agent avec contraintes de flux et de ressources.
- **Company Infrastructure** [75] : une simulation d'attaques et de défenses sur un réseau, inspirée de MITRE ATT&CK.
- **Drone Swarm** [55] : une simulation d'un essaim de drones soumis à des attaques logicielles et nécessitant une défense collective.
- **Microservices Kubernetes** : un environnement réel constitué d'un cluster de quatre micro-services interconnectés et que nous avons proposé pour l'orchestration de microservices avec auto-scaling et résilience face à des défaillances intentionnelles.

L'environnement **Microservices Kubernetes** se distingue des autres par sa nature réelle : il s'agit d'un système de quatre services interconnectés, alors que les autres environnements sont simulés. Pour des raisons pratiques, l'environnement simulé sert de référence « réelle », ce qui permet d'appliquer MAMAD comme sur un système opérationnel, le *World Model* jouant ici le rôle d'une simulation de la simulation. Ce choix valide le principe de MAMAD : si le *World Model* reconstruit à partir de traces est suffisamment fidèle, les politiques apprises peuvent être transférées efficacement, facilitant ainsi la comparaison entre simulation et réalité. Dans ce scénario, l'environnement **Microservices Kubernetes** est directement exploitable avec MAMAD sans nécessiter de modélisation supplémentaire, et le transfert (**TRF**) s'effectue vers ce système réel, montrant la validité de l'approche dans un contexte opérationnel.

Les algorithmes MARL sélectionnés couvrent les principales familles reconnues dans la littérature :

- MAPPO [54] : algorithme basé sur *Proximal Policy Optimization* (PPO) [124] adapté au multi-agent, utilisant des politiques centralisées pour stabiliser l'apprentissage tout en permettant une exécution décentralisée.
- MADDPG [121] : méthode d'apprentissage par gradient déterministe, combinant des politiques individuelles avec une critique centralisée pour gérer la non-stationnarité multi-agent.
- QMIX [115] : algorithme de factorisation de valeur, combinant les valeurs Q individuelles des agents via un réseau de mélange non linéaire pour optimiser une récompense globale.
- COMA [106] : approche basée sur l'acteur-critique, utilisant une estimation contrefactuelle pour attribuer précisément la contribution de chaque agent à la récompense collective.
- *Independent Q-Learning* (IQL) [59] : chaque agent apprend sa propre fonction Q de façon indépendante, sans coordination explicite, ce qui peut entraîner une non-stationnarité, mais reste simple à mettre en œuvre.
- VDN [125] : décompose la valeur globale en une somme des valeurs individuelles des agents, facilitant l'apprentissage coopératif tout en conservant une structure simple.

Les environnements sont implémentés via *PettingZoo* [87] et les algorithmes via *MARLlib* [41].

12.2 CONDITIONS DE REPRODUCTIBILITÉ

12.2.1 *Conditions expérimentales matérielles*

Les expériences sont réalisées sur un **cluster *High Performance Computing* (HPC) académique**. Sauf mention contraire, les constantes suivantes s'appliquent à tous les scénarios :

- **Accélérateurs** : *NVIDIA Corporation* (NVIDIA) A100 / V100, *Advanced Micro Devices* (AMD) MI210.
- **Frameworks DL** : PyTorch [98] et TensorFlow [127] (implémentations *Multi-Agent Reinforcement Learning Library* – MARLlib/MAPPO, etc.).
- **Optimisation d'hyperparamètres** : **Optuna** [91] (*Tree-structured Parzen Estimator* – TPE) pour *Learning Rate* (LR), exploration/exploitation, tailles de réseaux ; espace de recherche standardisé par famille d'algorithmes.
- **Parallélisme** : ~ 5 exécutions indépendantes par condition (algorithme $\times$ environnement $\times$ contrainte).
- ***Organizational Specifications* (OS) et libs** : Linux 64-bit, *Compute Unified Device Architecture* (CUDA)/cuDNN ou ROCm selon *Graphics Processing Unit* (GPU) ; environnements figés (conda/pip).

Les études de cas ne rappellent que les **déviations spécifiques** (ex. nombre de runs, GPU particulier).

12.2.2 *Gestion des hyperparamètres (par défaut et surcharges)*

Chaque couple {algorithme, environnement} est initialisé avec un *profil standard* issu de MARLlib et d'expériences préliminaires. Une passe d'HPO contrôlée peut être réalisée avec **Optuna** (budget borné, mêmes priorités d'espace de recherche entre scénarios) [91]. Le meilleur *trial* est ensuite **rejoué 5 fois** pour agrégation des résultats. Les scénarios peuvent :

a) accepter les valeurs par défaut,

b) restreindre l'HPO,

c) surcharger explicitement certains hyperparamètres.

12.3 BASELINES EXPÉRIMENTALES

Une baseline expérimentale est un ensemble de données qui servent à la reproduction de l'expérimentation. Dans notre cas, une baseline inclut notamment ces éléments :

- **Activités** : Une sélection d'activités MAMAD.
- **Algorithme** : Un algorithme MARL ou un autre algorithme issu de la littérature susceptible de permettre aux agents d'atteindre leurs objectifs avec une autre approche.
- **Spécifications organisationnelles** : Un ensemble de spécifications organisationnelles de type MOISE+MARL si l'algorithme choisi appartient au domaine du MARL.
- **Environnement** : Un environnement avec variantes des scénarios (ex. Overcooked-AI [79], Predator-Prey [121], Drone Swarm [55]).
- **Métriques spécifiques** : Un ensemble de métriques spécifiques à l'environnement ou l'étude (spécifiques à l'environnement).
- **Conditions d'exécution** : Un ensemble de conditions d'exécution (nb de runs, seeds, matériel, perturbations éventuelles).

Dans le cadre des expérimentations, certaines caractéristiques des baselines (comme l'environnement ou les métriques spécifiques) restent fixes. Pour instancier une baseline, on se concentre donc sur les éléments variables tels que l'algorithme utilisé, les conditions d'exécution ou les spécifications organisationnelles. L'objectif principal de définir des baselines est de disposer de points de comparaison pour mesurer l'impact des différentes composantes de la méthode MAMAD. En confrontant les résultats obtenus avec ces baselines, on peut ainsi isoler et analyser la contribution de chaque activité (modélisation, entraînement, analyse, transfert) ainsi que l'effet des spécifications organisationnelles sur le comportement et la performance du SMA.

12.4 GRILLE D'ÉVALUATION

12.4.1 *Critères et métriques associées*

L'évaluation s'appuie sur une grille de critères (voir Table 16) inspirée des recommandations de la communauté RL/MARL [72] :

Ces différentes métriques visent à déterminer dans quelle mesure les cinq critères globaux (C1–C5) sont couverts sur une baseline donnée. Chaque métrique est décrite séparément pour plus de clarté.

TABLE 16 : Correspondance entre critères globaux et métriques

Critère	**Métriques associées**
C1 – Autonomie	Proportion d'intervention (conception / fonctionnement)
C2 – Performance	Récompense cumulée ; Taux de convergence
C3 – Adaptation	Écart-type des récompenses ; Score de robustesse
C4 – Contrôle	Taux de violation des contraintes ; Score de cohérence
C5 – Explicabilité	Adéquation organisationnelle ; Qualité des spécifications inférées

PROPORTION D'INTERVENTION (CONCEPTION / FONCTIONNEMENT). *Unité : pourcentage (%). Source : logs d'utilisation de la plateforme, questionnaires utilisateurs, scripts d'automatisation, nombre de cycles de raffinement, estimations manuelles.* Elle est calculée comme le rapport suivant :

$$\frac{\text{temps estimé pour la conception automatisée}}{\text{temps estimé pour la conception manuelle}}$$

Cette proportion, moins précise, vise à quantifier au moins approximativement l'impact de l'automatisation sur le processus de conception : quel pourcentage d'interventions manuelles est nécessaire pour atteindre des performances similaires à des SMAs conçus et implémentées manuellement. Elle peut être estimée via l'instrumentation logicielle en indiquant qu'un nombre de cycles de raffinement important augmente la proportion d'intervention (i.e le ratio du nombre de cycles utilisés par rapport au nombre de "cycles" empiriques de conception manuelle). Considérant l'environnement déjà modélisé, cette proportion est aussi obtenue manuellement par estimation du temps passé (en nombre d'heures) durant la conception totalement manuelle, ce qui dans notre méthode correspond au temps estimé passé (en nombre d'heures) dans les cycle de raffinement alternant entre d'entraînement et analyse.

RÉCOMPENSE CUMULÉE. *Unité : valeur numérique sans unité (souvent normalisée). Source : logs d'entraînement MARLlib/RLlib, fichiers de résultats d'épisodes.* Il s'agit de la somme des récompenses obtenues par tous les agents sur un épisode ou sur une fenêtre glissante. Elle est extraite automatiquement via des scripts d'analyse.

TAUX DE CONVERGENCE. *Unité : nombre d'épisodes (entier). Source : courbes d'apprentissage, logs d'entraînement.* Il correspond au nombre d'épisodes nécessaires pour que la moyenne des récompenses dépasse un seuil prédéfini et reste stable. Le calcul est automatisé par détection de plateau sur la courbe d'apprentissage.

ÉCART-TYPE DES RÉCOMPENSES. *Unité : identique à la récompense (souvent sans unité). Source : logs d'entraînement, résultats de runs multiples.* Cette métrique correspond à l'écart-type statistique des récompenses moyennes entre plusieurs runs indépendants. Elle est calculée automatiquement lors de l'agrégation des résultats.

SCORE DE ROBUSTESSE. *Unité : ratio ou pourcentage (%). Source : tests sous perturbations (pannes, attaques), logs d'épisodes.* Il se définit comme :

$$\frac{\text{performance sous perturbation}}{\text{performance nominale}}$$

Ce score est obtenu en comparant les performances moyennes dans des scénarios perturbés et des scénarios de référence. Les perturbations sont générées soit par des seeds différentes dans les environnements simulés, soit par des changements explicites (topologie, pannes, attaques, etc.).

TAUX DE VIOLATION DES CONTRAINTES. *Unité : pourcentage (%). Source : logs d'exécution, wrappers d'environnement, analyse post-hoc.* La formule est :

$$\frac{\text{nombre de violations détectées}}{\text{nombre total d'étape par épisode}}.$$

Cette mesure évalue la capacité des agents à respecter les règles imposées par leurs rôles. Elle varie avec la dureté des contraintes : un taux nul est attendu lorsque la dureté est maximale, et un taux élevé lorsque les contraintes sont nulles. Les valeurs intermédiaires doivent être analysées en lien avec la récompense cumulée afin d'identifier d'éventuelles règles trop contraignantes qui réduiraient les performances globales.

SCORE DE COHÉRENCE. *Unité : ratio (0–1) ou pourcentage (%). Source : clustering des trajectoires, analyse TEMM/Auto-TEMM.* Ce score mesure la similarité entre les comportements observés et les rôles attendus. L'idée est de comparer les spécifications organisationnelles originales avec celles inférées automatiquement. Plus la distance entre les deux ensembles de trajectoires est faible, plus le score de cohérence est élevé, indiquant une bonne prise en compte des spécifications organisationnelles par MOISE+MARL et une capacité d'inférence fiable de TEMM/Auto-TEMM.

ADÉQUATION ORGANISATIONNELLE. *Unité : ratio (0–1). Source : analyse TEMM, matrices de correspondance rôles-missions.* Comme expliqué précédemment en Sous-section 9.2.1, elle est calculée comme une moyenne pondérée des scores structurels (SOF) et fonctionnels (FOF) avec $\alpha = 0.5$ par défaut :

$$\mathrm{OF} = \alpha \cdot \mathrm{SOF} + (1 - \alpha) \cdot \mathrm{FOF}.$$

Cette mesure quantifie à quel point les politiques apprises respectent la structure organisationnelle attendue.

QUALITÉ DES SPÉCIFICATIONS INFÉRÉES. *Unité : pourcentage (%) ou score de similarité. Source : comparaison automatique entre spécifications inférées (JavaScript Object Notation – JSON, TEMM) et spécifications de référence.* Cette métrique mesure la qualité des spécifications organisationnelles inférées en comparant leur similarité avec les spécifications de référence, à l'aide d'indicateurs tels que l'indice de Jaccard [194] (proportion d'éléments partagés) ou la distance Euclidienne. Contrairement au score de cohérence, elle s'intéresse uniquement à la fidélité des spécifications extraites, indépendamment des politiques apprises. Le calcul repose sur une comparaison systématique entre les spécifications inférées et les spécifications initiales, en variant les scénarios pour évaluer la capacité de généralisation. L'objectif est de trouver un compromis entre la fidélité aux spécifications de départ et la capacité de généralisation, en évitant le surapprentissage et en limitant la complexité des règles ou objectifs inférés, tout en maximisant la similarité avec les trajectoires de référence.

12.5 PROTOCOLE D'EXPÉRIMENTATION ET D'ÉVALUATION

Le protocole expérimental proposé suit un raisonnement progressif semblable aux pratiques de la communauté [72]. L'ensemble des étapes 1 à 4 constitue le protocole d'expérimentation tandis que les étapes 5 et 6 concernent l'évaluation des résultats :

1. CONFIGURATION INITIALE Configuration requise incluant la mise en place de l'*Application Programming Interface* (API) *Representational State Transfer* (REST) de communication avec l'environnement et les différentes composantes requises pour modéliser l'environnement et autres données nécessaires pour créer un projet CybMASDE pour une baseline donnée.

2. MISE EN PLACE DE LA BASELINE AVANCÉE Définition d'au moins une **baseline avancée** considérée comme la **baseline par défaut** :

TABLE 17 : Caractérisation générique de la "baseline avancée"

Élément	**Valeur instanciée**
Environnement	Nom de l'environnement et éventuelles variantes de scénarios (ex. Overcooked-AI, layout classique).
Activités MAMAD	MOD-AUT,TRN-CON,ANL-AUT,TRF-AUT
Algorithme	Algorithme sélectionné avec son implémentation (ex. MAPPO via MARLlib).
Spécifications organisationnelles	Ensemble MOISE+MARL contenant à la fois des rôles et des missions.
Métriques spécifiques	Indicateurs complémentaires propres à l'environnement (ex. plats servis, taux de collisions, drones infectés, latence moyenne).
Conditions d'exécution	Nb de runs, seeds, type de matériel (CPU/GPU), perturbations/stress-tests.

3. MISES EN PLACE DE BASELINES ALTERNATIVES Definition possible d'autres baselines jouant sur d'autres paramètres tels que les algorithmes MARL, les contraintes organisationnelles ou la modélisation en fonction des questions spécifiques de chaque scénario. Des base

4. DÉFINITION D'ÉTUDES D'ABLATION Réalisation d'au moins une **étude d'ablation** : chaque ablation consiste à prendre la **baseline avancée** et à changer un ou plusieurs paramètres notamment à remplacer une ou plusieurs activités MAMAD par des variantes moins avancées (par exemple, "ANL-MAN" en utilisant TEMM plutôt que Auto-TEMM, ou "TRN-UNC" sans contraintes organisationnelles), mais aussi à supprimer les contraintes organisationnelles, à changer d'algorithme, etc. Chaque ablation doit être justifiée en fonction des questions spécifiques de chaque scénario.

5. COMPARAISON ET VALIDATION Comparaison systématiques des résultats sur la grille d'évaluation (Table 16) pour mesurer l'impact de chaque composant sur les critères cibles. L'analyse des résultats doit inclure des visualisations (courbes d'apprentissage,

heatmaps, dendrogrammes, etc.) et une discussion critique des compromis observés (ex. performance vs. respect des contraintes).

6. RÉPÉTITION ET AGRÉGATION Répétition des expériences (5 runs indépendants, seeds consignées), agrégation statistique (moyenne, écart-type), et tests statistiques (t-test ou non paramétrique selon la normalité).

12.6 BILAN

Ce chapitre a posé les fondations méthodologiques de l'évaluation expérimentale de la méthode MAMAD. En définissant un cadre générique, reproductible et structuré, il permet de comparer les performances, la robustesse et l'explicabilité de la méthode dans des environnements variés, allant des cas-jouets aux systèmes réels. L'intérêt majeur de ce cadre est d'assurer la traçabilité des choix, la transparence des protocoles et la validité des résultats, tout en facilitant l'analyse critique des apports et des limites de chaque composant de la méthode. Ce dispositif méthodologique est essentiel pour garantir la crédibilité scientifique des expérimentations et pour identifier les leviers d'amélioration futurs. Les chapitres suivants s'appuient sur ce cadre pour instancier, analyser et discuter l'application de MAMAD dans des contextes concrets, en mettant en lumière sa capacité à répondre aux critères d'autonomie, de performance, d'adaptation, de contrôle et d'explicabilité.

13 ÉTUDES DE CAS

Ce chapitre présente les études de cas réalisées pour évaluer la méthode MAMAD dans divers contextes. Chaque section détaille l'instanciation du cadre expérimental défini au Chapitre 12 pour un environnement spécifique, en précisant les choix méthodologiques et les configurations expérimentales.

13.1 EXPÉRIMENTATIONS SUR LES ENVIRONNEMENTS NON-ORIENTÉS CYBERDÉFENSE

Dans cette section, nous décrivons les expérimentations qui ont été menées sur un ensemble d'environnements de référence non orientés Cyberdéfense. L'objectif principal est de valider la méthode MAMAD dans des contextes variés, en mettant l'accent sur la coordination, la compétition, et la gestion de ressources dans des environnements à observation partielle. Le but est d'avoir une preuve de concept que la méthode MAMAD est applicable sur des cas simples. A partir de là, il est possible d'aborder efficacement son application à des scénarios strictement liés à la Cyberdéfense.

Cette section propose la même instance du protocole d'expérimentation de la Section 12.5 (étapes 1 à 4) pour trois environnements non-orientés Cyberdéfense (*Overcooked-AI*, *Predator-Prey*, *Warehouse Management*). L'objectif est de décrire les expérimentations à mener pour appliquer le protocole d'évaluation de la Section 12.5 (étape 5 à 6) dans le chapitre suivant afin de valider la méthode MAMAD sur des contextes coopératifs/compétitifs variés, avec observation partielle et besoins de coordination.

13.1.1 *Description des environnements*

Overcooked-AI

L'environnement **Overcooked-AI** [92] simule un scénario de cuisine coopérative où les agents doivent collaborer pour préparer et servir des repas dans une cuisine structurée. Cet environnement est illustré dans Figure 26.

- **Espace d'état :** Une cuisine discrète basée sur une grille avec des postes de travail (planche à découper, cuisinière, comptoir de service), des ingrédients et des agents
- **Espace d'observation :** Les agents observent les éléments de la cuisine dans un rayon défini
- **Espace d'action :** i) Déplacement : "Haut, Bas, Gauche, Droite" ; ii) Interagir : "Choisir un ingrédient, couper, cuisiner, servir".
- **Structure de récompense :** i) Préparation réussie du repas : $+20$; ii) Mauvaise utilisation des ingrédients : -5 ; iii) Comportement passif : -1 par étape sans action significative.
- **objectif :** Maximiser le nombre de commandes de repas terminées dans un délai fixe.

Spécifications organisationnelles :

- **Rôles :** "Chef, assistant, serveur"
- **Missions :** Le chef prépare les plats, l'assistant fournit les ingrédients et le serveur sert les repas
- **Contraintes :** L'exécution des tâches doit être synchronisée afin d'éviter les goulots d'étranglement.

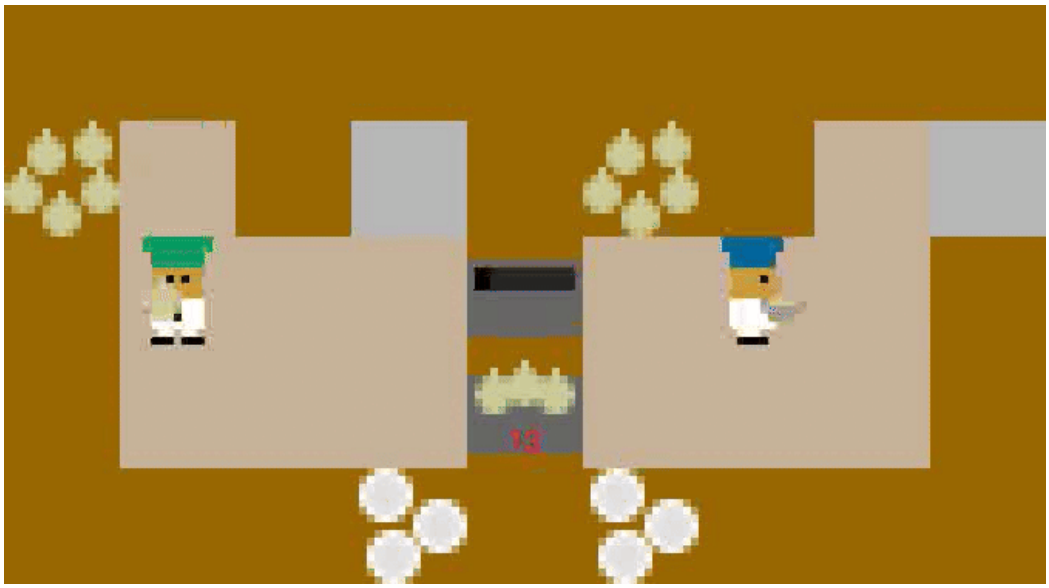


FIGURE 26 : Capture d'écran de l'environnement Overcooked-AI : deux agents (chefs cuisiniers) doivent collaborer pour préparer et servir efficacement des soupes à l'oignon. Le processus consiste à prélever trois oignons (un à la fois) dans le distributeur, à les placer dans une marmite, à attendre que la soupe cuise, à récupérer un plat propre, à dresser la soupe et à la livrer au comptoir de service. La disposition de la cuisine comprend des obstacles et des passages étroits, ce qui oblige les agents à coordonner leurs mouvements pour éviter les collisions et optimiser l'accomplissement des tâches.

Predator-Prey

L'environnement **Predator-Prey** est un benchmark MARL bien connu [121], conçu pour évaluer la coordination entre des poursuivants coopératifs (prédateurs) qui tentent de capturer un agent insaisissable (proie). Cet environnement est illustré dans Figure 27.

- **Espace d'état :** Un espace 2D continu où les agents (prédateurs et proies) ont des positions (x, y) et des vitesses
- **Espace d'observation :** Les agents détectent les entités proches dans un rayon limité r
- **Espace d'action :** i) Déplacement : "Haut, Bas, Gauche, Droite, Rester sur place".
- **Structure de récompense :** i) Les prédateurs gagnent $+50$ pour chaque proie capturée ; ii) La proie gagne $+1$ par étape de temps survécue ;.

- **objectif :** Les prédateurs doivent coopérer pour piéger la proie, tandis que celle-ci tente de s'échapper aussi longtemps que possible.

Spécifications organisationnelles :

- **Rôles :** "Prédateur, Proie"
- **Missions :** Les prédateurs coordonnent leurs efforts pour encercler la proie ; la proie cherche les meilleurs chemins pour s'échapper
- **Contraintes :** Les prédateurs doivent trouver un équilibre entre poursuite agressive et stratégies de blocage.

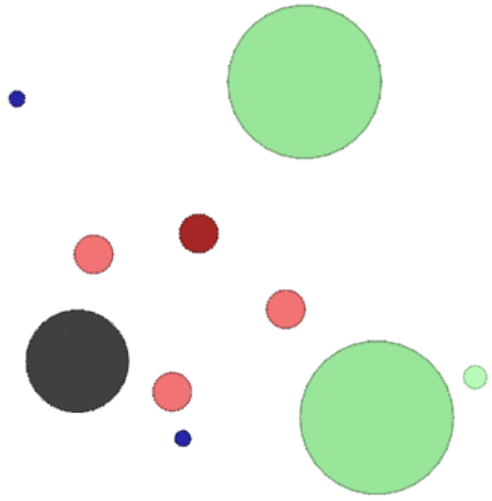


FIGURE 27 : Capture d'écran de l'environnement Predator-Prey : **agents verts** (coopératifs) et **agents rouges** (adversaires). Les agents verts ont pour objectif de collecter les aliments dispersés dans l'environnement tout en évitant d'être détectés par les agents rouges. L'environnement comprend des **zones forestières** qui offrent un abri ; lorsqu'un agent vert pénètre dans une forêt, il devient partiellement ou totalement invisible aux yeux des agents rouges. Un agent rouge agit en tant que **chef** avec des capacités d'observation améliorées et peut communiquer avec les autres agents rouges afin de coordonner leur poursuite.

Warehouse Management

L'environnement **Warehouse Management** [30] modélise un entrepôt logistique basé sur une grille où plusieurs robots doivent collaborer pour transporter efficacement les marchandises. Cet environnement s'inspire des scénarios d'automatisation des entrepôts industriels et constitue un banc d'essai idéal pour évaluer la répartition des tâches, la spécialisation des rôles et la coordination en temps réel. Cet environnement est illustré dans Figure 28.

- **Espace d'état :** Une grille $N \times M$ où chaque cellule contient un robot, un produit, une machine de fabrication ou un lieu de dépôt. Le système suit les positions des agents, les niveaux de stock et les états des machines
- **Espace d'observation :** Chaque agent dispose d'une vue locale $V \times V$, lui permettant de percevoir les produits, ses coéquipiers et les machines à proximité
- **Espace d'action :** i) Déplacement : "Haut, Bas, Gauche, Droite" ; ii) Interagir : "Prendre un produit, déposer un produit".
- **Structure de récompense :** i) Livraison réussie du produit : $+10$; ii) Déplacement inefficace : -1 par étape inutile ; iii) Mauvaise manipulation du produit : -5 pour les livraisons incorrectes.
- **objectif :** Transporter les matières premières vers les machines de transformation et livrer les produits finis aux lieux de livraison.

Spécifications organisationnelles :

- **Rôles :** "Transporteur, gestionnaire des stocks"
- **Missions :** Les transporteurs acheminent les produits, tandis que les gestionnaires des stocks supervisent les niveaux des stocks
- **Contraintes :** Les transporteurs doivent donner la priorité aux livraisons essentielles.

13.1.2 *Description de l'instance commune du protocole d'expérimentation*

1. CONFIGURATION INITIALE Nous utilisons les implémentations *PettingZoo* de **Overcooked-AI**, **Predator-Prey** et **Warehouse Management** comme *environnements réels* au sens de "CybMASDE". Concrètement, pour un environnement donné, une instance du jeu est lancée en tâche de fond et exposée via un adaptateur REST conforme à l'API d'I/O de "CybMASDE" (observations jointes, masques d'actions, *step*/*reset*). Cette passerelle permet (i) la collecte de traces pour la modélisation ("MOD-AUT") ; (ii) l'entraînement MARL avec contraintes organisationnelles ; (iii) l'analyse organisationnelle (**ANL**) ; (iv) le transfert (**TRF**) vers l'exécuteur simulé standardisé. Pour "MOD-AUT", un JOPM, VAE+LSTM) est entraîné sur les historiques collectés (politiques aléatoires et politiques préliminaires) afin d'apprendre la dynamique $\langle o_{1:t}, a_{1:t} \rangle \mapsto o_{t+1}$ et une fonction d'arrêt dérivée. La fonction de récompense reconstruite à partir des traces est validée par recoupement avec la récompense native de l'environnement. Le mappage *labels* observation/action (l_o, l_a) est défini pour synchroniser "PettingZoo" et MMA, activer le masquage d'actions et l'injection de bonus/malus par rôle. L'entraînement s'appuie sur "MARLlib"/"RLlib" (profil "MAPPO" par défaut, "Optuna" activé), avec *seeds* fixées et 5 exécutions indépendantes conformément au Chapitre 12. Les ressources matérielles suivent celles présentées Sous-section 12.2.1 ; seules les dérogations (longueur d'épisode, fréquence de log) sont précisées dans les résultats.

2 À 4. DÉFINITION DES BASELINES Pour les trois environnements, nous définissons une **baseline avancée par défaut** et des **baselines alternatives** (ablations) qui varient les activités MAMAD, l'algorithme MARL, le mode d'intégration des contraintes et leur dureté, afin d'isoler l'apport de chaque composant (guidage organisationnel, modélisation, analyse). Nous ne prenons pas en compte de métriques spécifiques comme les

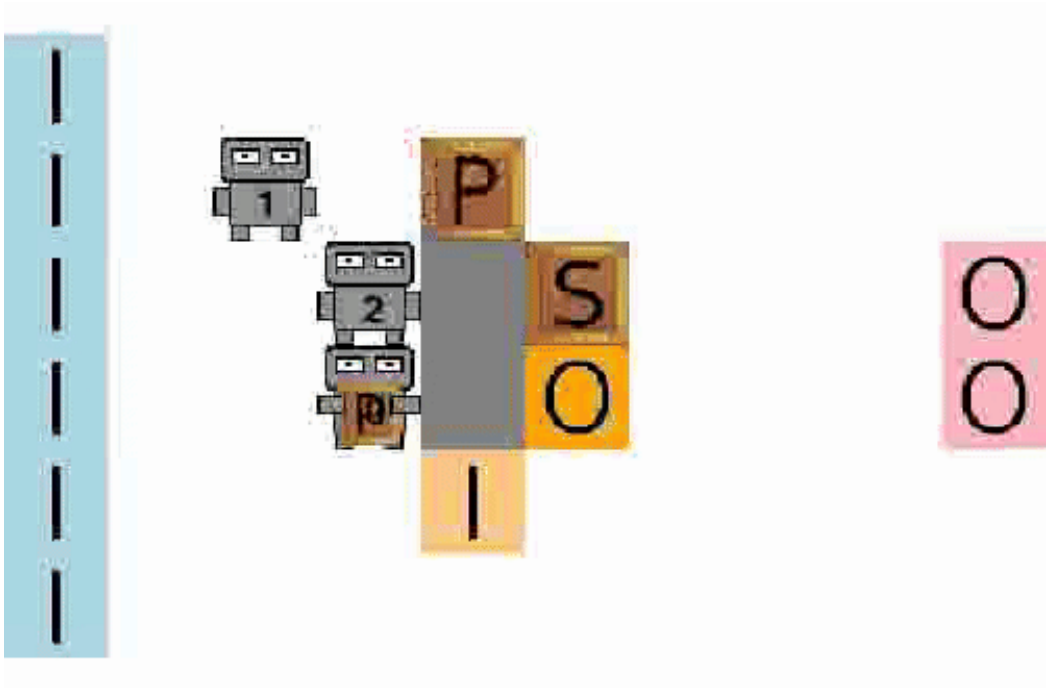


FIGURE 28 : Capture d'écran de l'environnement de Warehouse Management : les agents peuvent se déplacer vers le haut, le bas, la gauche et la droite. Plusieurs agents opèrent au sein d'une grille d'entrepôt, effectuant des tâches pour traiter et livrer des produits. Les agents peuvent se déplacer dans quatre directions (haut, bas, gauche, droite) et interagir avec les zones de prélèvement/dépôt lorsqu'ils sont adjacents. Le flux de travail comprend : (i) la collecte des produits primaires dans les zones de prélèvement/dépôt du convoyeur d'entrée (zones bleues) ; (ii) leur transport vers les zones de prélèvement/dépôt des machines de fabrication (zones marron), où les produits primaires sont transformés en un seul produit secondaire selon un schéma de fabrication prédéfini ; (iii) la récupération des produits secondaires obtenus et leur livraison aux zones de prélèvement/dépôt du convoyeur de sortie (zones roses). Pour que l'opération soit réussie, les agents doivent coordonner leurs mouvements et leurs actions afin d'optimiser le débit et l'efficacité au sein de l'entrepôt.

plats servis/épisode, collisions, pas inactifs, temps jusqu'au palier de performance pour Overcooked-AI par exemple. La Table 18 résume les baselines expérimentales prévues pour l'ensemble des trois environnements.

13.2 EXPÉRIMENTATIONS SUR L'ENVIRONNEMENT COMPANY INFRASTRUCTURE

Company Infrastructure [75] : une simulation d'attaques et de défenses sur un réseau,.

L'environnement **Company Infrastructure** est une simulation inspirée du simulateur CyberbatlleSim [75] et de MITRE ATT&CK [1]. Cet environnement a été développé avec le Dec-POMDP pré-spécialisé pour la Cyberdéfense et avec MCAS intégré à CybMASDE. Il simule un réseau d'entreprise découpé en sous-réseaux (EXT, DMZ, ACC, MAR, SRV) où des *agents cyber-attaquants* et *agents cyber-défenseurs* interagissent via des actions à pré/post-conditions, selon un formalisme *Dec-POMDP*. La topologie synthétique est illustrée en Figure 29, et les chemins d'attaque/défense sont structurés via un arbre Attaque–Défense (Figure 30).

TABLE 18 : Baselines synthétiques pour les environnements non-orientés Cyberdéfense.

Profil d'activités MAMAD	Algorithmes MARL	Contraintes org. (dureté)	Commentaires
Profil A – Défaut MOD-AUT ; TRN-CON ; ANL-AUT ; TRF-AUT	MAPPO	Oui (1.0)	Masquage d'actions + shaping par rôle (MMA) ; JOPM activé pour MOD-AUT.
	MADDPG	Douces (0.5)	Contraintes atténuées : bonus/malus et masques partiels ; même pipeline que défaut.
	QMIX	Aucune (0.0)	*Ablation* : TRN-UNC, récompense native de l'environnement, reste inchangé.
Profil B – Analyse manuelle MOD-AUT ; TRN-CON ; ANL-MAN ; TRF-AUT	MAPPO	Oui (1.0)	Paramétrage manuel de TEMM ; règles/masques édités à la main.
	COMA	Douces (0.5)	Guidage souple : pénalités réduites ; vérification post-hoc par TEMM ajusté manuellement.
		Aucune (0.0)	TRN-UNC ; analyse TEMM uniquement pour explicabilité/diagnostic, sans réinjection.
Profil C – Cycle principalement manuel MOD-MAN ; TRN-CON ; ANL-MAN ; TRF-MAN	IQL	Oui (1.0)	Environnement "handcrafted" ; hyperparamètres fixés ; transfert et déploiement manuels.
	VDN	Douces (0.5)	Contraintes souples définies manuellement (rôles/missions + barèmes adoucis).
	MADDPG	Aucune (0.0)	TRN-UNC entièrement manuel ; dureté de contraintes nulle organisationnel.

- **Espace d'état :** Un ensemble de *propriétés* discrètes décrivant l'état des nœuds du réseau (fichiers, services actifs, versions, règles de pare-feu, sessions, journaux, connaissances d'agents). La topologie comprend : a) **EXT** : deux postes d'attaquants "At1, At2" ; b) **DMZ** : "*Web Server* (WS), *Email Server* (ES), *Virtual Private Network* (VPN), FTP" ; c) **ACC** : "E1, E2, CTO" ; d) **MAR** : "*Privileged Service* (PS), E3, *Terminal Access Broker* (TAB)" ; e) **SRV** : "API, *Database* (DB), DC". Les transitions modifient l'ensemble des propriétés (ajout/suppression/mise à jour) quand les pré-conditions d'action sont satisfaites

- **Espace d'observation :** Observations partielles spécifiques à chaque agent (relation Obs) comme les contenus de fichiers/journaux, résultats de commandes, scans de ports, états de sessions, alertes de détection. Les observations sont retournées après application d'action et dépendent de la visibilité locale (capteurs, privilèges)

- **Espace d'action :** Actions à *pré-conditions* (conjonctions/disjonctions de propriétés) et *post-conditions* (écritures sur l'état), réparties en grandes familles : a) **Attaquants** : "Reconnaissance réseau/compte", "Exploiter vulnérabilité service", "Élévation de privilèges", "Mouvement latéral", "Persistance (ex. backdoor)", "Exfiltration de données (DB)", "Installation spyware (PS)" ; b) **Défenseurs** : "Détection journaux (WS)", "Gestion comptes privilégiés (*Privileged Access Management* – PAM)", "Surveillance commandes/arguments (DB)", "Blocage trafic/règles FW", "Suppression sessions malveillantes", "Restauration services".

- **Structure de récompense :** Fonction $R = \mathrm{Eval} \circ \mathrm{Metrics}$ évaluant l'état & l'action par des métriques (progrès d'attaque vers les objectifs, détections, sessions supprimées, intégrité des services, etc.) : a) **Attaquants** : bonus de progression le long de l'AD-tree ($+r_{step}$), objectifs ultimes : "Exfiltration DB" et "Spyware PS" ($+R_{goal}$), pénalités si détectés/neutralisés ($-\lambda_{det}$) ; b) **Défenseurs** : bonus détection/prévention ($+r_{det}$), suppression sessions ($+r_{purge}$), maintien disponibilité/Intégrité ($+r_{avail}$), pénalités si objectifs attaquants atteints ($-\Lambda_{goal}$).

- **Objectif :** Pour **Attaquants**, atteindre *Exfiltration DB* et *Spyware PS* en minimisant la détection ; pour **Défenseurs**, prévenir/détecter/neutraliser ces chemins d'attaque tout en maintenant les services critiques.

Spécifications organisationnelles : *(Baseline)* aucune contrainte organisationnelle n'est imposée ("TRN-UNC"), afin de fournir une référence *brute* pour les scénarios orientés Cyberdéfense guidés. *(Variante optionnelle, pour analyses ablatives)* :

- **Rôles (ex.)** : "Attacker_LateralMove", "Attacker_ExfilDB", "Defender_WS_Monitor", "Defender_DB_PAM"
- **Missions** : chaîner les techniques MITRE par sous-objectif (recon → exploitation → élévation → mouvement latéral → action sur l'objectif) côté rouge ; détection → confinement → éradication → reprise côté bleu
- **Contraintes** : autorisations par rôle (masquage d'actions), séquencement minimal de missions, déclencheurs de confinement sous conditions (journaux/*Indicator of Compromise* – IOC).

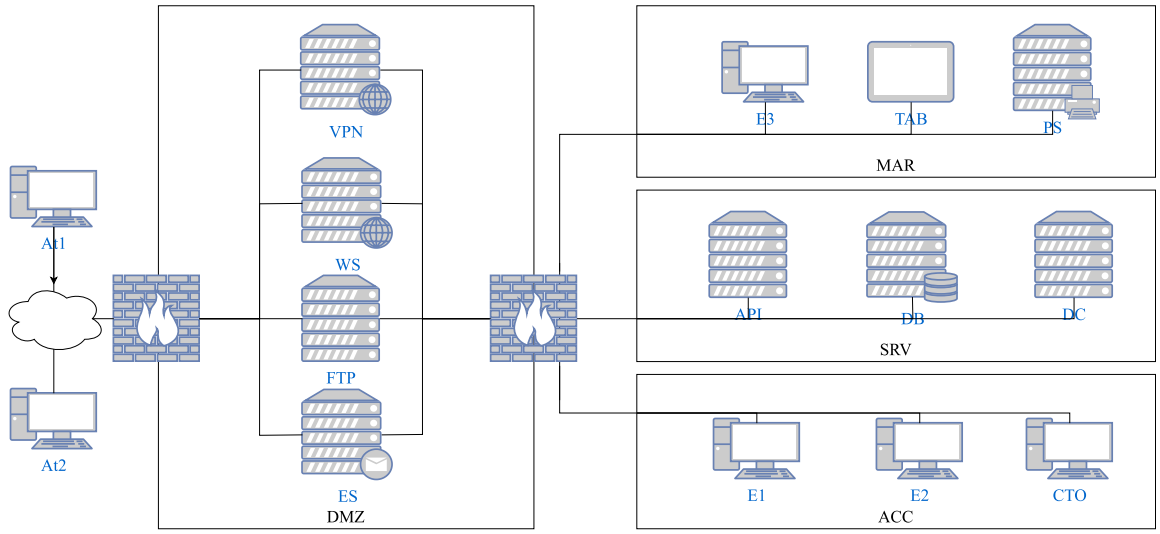


FIGURE 29 : Topologie réseau synthétique : EXT, DMZ, ACC, MAR, SRV. Les sous-réseaux sont interconnectés via routeurs/pare-feu implicites.

13.2.1 *Description de l'instance du protocole d'expérimentation*

1. CONFIGURATION INITIALE Nous utilisons le modèle *Dec-POMDP* pré-spécialisé **Cyberdéfense** et le simulateur **MCAS** intégrés à "CybMASDE", exposés via un adaptateur REST conforme à l'API d'I/O "CybMASDE" (observations jointes, masques d'actions, *step*/*reset*). L'environnement *Company Infrastructure* est donc traité, côté pipeline, exactement comme les environnements-jouets non-orientés Cyberdéfense : (i) collecte de traces pour la modélisation automatique ("MOD-AUT") ; (ii) entraînement MARL avec ou sans contraintes organisationnelles ; (iii) analyse organisationnelle (**ANL**) ; (iv) transfert automatique ("TRF-AUT") vers l'exécuteur simulé. **Justification :** aucun "TRF-MAN" n'est requis ici, l'environnement n'étant pas une instrumentation d'un système réel, mais un simulateur nativement compatible (pas de "passerelle terrain" à maintenir). Pour "MOD-AUT", un JOPM (VAE+LSTM) est entraîné sur des historiques mêlant trajectoires aléatoires et politiques préliminaires *rouges/bleues* (progression dans l'AD-tree, détections, quarantaines). Le JOPM approxime la dynamique $\langle o_{1:t}, a_{1:t} \rangle \mapsto o_{t+1}$ et fournit une fonction d'arrêt dérivée (objectifs atteints, neutralisation des sessions malveillantes, *step-limit*). La fonction



FIGURE 30 : Aperçu de l'arbre Attaque–Défense (AD) structurant les chemins d'attaque (tactiques/-techniques MITRE) et les contre-mesures associées.

de récompense reconstruite est validée par recoupement avec les métriques natives (progrès le long de l'AD-tree, disponibilité des services). L'entraînement s'appuie sur "MARLlib"/"RLlib" (profil "MAPPO" par défaut, mais variantes testées ci-dessous), *seeds* fixées, 5 exécutions indépendantes comme dans le Chapitre 12. Les conditions de calcul suivent celles présentées en Sous-section 12.2.1 ; seules les dérogations (longueur d'épisode, fréquence de log) sont précisées dans les résultats.

2 À 4. DÉFINITION DES BASELINES Nous définissons une **baseline avancée par défaut** et des **ablations** qui font varier (a) les activités MAMAD, (b) l'algorithme MARL, (c) la *dureté* des contraintes organisationnelles : *fortes* (1.0), *douces* (0.5), *sans* (0.0). Toutes reposent sur "TRF-AUT" (pas de "TRF-MAN", cf. justification ci-dessus).

TABLE 19 : Baselines synthétiques pour Company Infrastructure.

Profil d'activités MAMAD	Algorithmes MARL	Contraintes org. (dureté)	Commentaires
Profil A – Défaut "MOD-AUT" ; "TRN-CON" ; "ANL-AUT" ; "TRF-AUT"	MAPPO, QMIX, COMA	Oui (1.0)	Masquage par rôle (rouge/bleu), shaping aligné AD-tree (progrès d'attaque / détection / quarantaine) ; JOPM activé.
	MAPPO, QMIX, COMA	Douces (0.5)	Même pipeline, pénalités/bonus atténués ; masques partiels (exploration élargie).
	MAPPO, QMIX	Aucune (0.0)	*Ablation* TRN-UNC : récompense native (état/metrics), pas de guidage organisationnel.
Profil B – Analyse manuelle "MOD-AUT" ; "TRN-CON" ; "ANL-MAN" (TEMM paramétré) ; "TRF-AUT"	MAPPO, COMA	Oui (1.0)	TEMM paramétré à la main ; règles/masques édités (séquencement recon→exploitation→LM→objectif).
	MAPPO, COMA	Douces (0.5)	Guidage souple, contrôle post-hoc par TEMM.
	MAPPO	Aucune (0.0)	TRN-UNC ; TEMM seulement pour explicabilité/diagnostic (pas de réinjection de règles).
Profil C – Cycle semi-manuel "MOD-MAN" (actions/props étendues) ; "TRN-CON" (hp manuels) ; "ANL-MAN" ; "TRF-AUT"	IQL, VDN, QMIX	Oui (1.0)	Élargissement "handcrafted" du catalogue d'actions (FW, PAM, persistance) ; hyperparamètres fixés (pas d'HPO).
	IQL, VDN, QMIX	Douces (0.5)	Contraintes souples définies manuellement (seuils journaux/IOC, priorités services).
	IQL, VDN	Aucune (0.0)	Dureté de contraintes nulle ; utile pour jauger l'apport des contraintes et du shaping.

13.3 EXPÉRIMENTATIONS SUR L'ENVIRONNEMENT MICROSERVICES KUBERNETES

L'environnement **Microservices Kubernetes** [14] est un *cluster réel* (1 nœud *worker* : 8 vCPU, 32 Go *Random Access Memory* – RAM, 1 Gbps) utilisé comme *environnement d'entrée* de la méthode MAMAD. Une illustration schématique de ce cluster et de ses services est fournie en Figure 31. Le cluster héberge une application web de e-commerce composée de 4 microservices en chaîne (API, Auth, Produits, Commandes) orchestrés via Kubernetes. Chaque service est déployé dans un *pod* avec un nombre variable de réplicas (1 à 5). Le cluster est surveillé en temps réel via Prometheus/Grafana, collectant des métriques telles que l'utilisation CPU/mémoire, le taux de requêtes, la latence, les files d'attente, et l'état des pods. Des scénarios de stress-test sont appliqués pour simuler des conditions réelles : goulots d'étranglement (augmentation du trafic), attaques DDoS (trafic malveillant), pannes de pods (simulées via "kubectl delete pod"), et contention de ressources (limitation CPU/mémoire). L'objectif est d'évaluer la capacité des agents à maintenir la résilience opérationnelle du cluster en adaptant dynamiquement les ressources et en répondant aux incidents. Une illustration synthétique de ce type de scénarios est fournie en Figure 32.

- **Espace d'état :** état courant du cluster réel et des 4 services en chaîne ($i \in \{1..4\}$) : $s = \{\text{réplicas}^i, U^i_{cpu}, U^i_{mem}, T^i_{in}, T^i_{out}, Q^i_{pending}, S^{i,pods}_{status}, P^i_{priority}\}_{i=1..4}$ et agrégats globaux (latence moyenne L_{avg}, taux de requêtes R_{rate}, disponibilité)
- **Espace d'observation :** observation *réelle* partielle (Dec-POMDP) obtenue via l'API Kubernetes/collecte métriques. Elle est *spécifique au rôle* : goulots : $Q^i_{pending}, T^i_{in}/T^i_{out}$; DDoS : $R_{rate}, \Delta T, L_{avg}$; pannes : $S^{i,pods}_{status}, F^i_{fail}$; ressources : $U^i_{cpu}, U^i_{mem}, P^i_{priority}$
- **Espace d'action :** *actions réelles* appliquées au cluster via l'API Kubernetes : i) *Scaling ciblé* : "scale_up"(i), "scale_down"(i) ; ii) *Gestion DDoS* : "rate_limit_ingress"(i), "isolate_service"(i) ; iii) *Récupération pannes* : "restart_failed_pod"(i), "reschedule_pod"(i) ; iv) *Arbitrage ressources* : "throttle_low_prio"(i), "rebalance_quota"(i)
- **Structure de récompense :** récompense *calculée sur télémétrie réelle* (QoS/résilience) :

$$R_{global} = w_1 \cdot \text{SuccessRate} - w_2 \cdot \overline{Q_{pending}} - w_3 \cdot L_{avg} - w_4 \cdot \text{DownTime} - w_5 \cdot \text{OverProvision},$$

complétée par des sous-récompenses par rôle :

$$R_{\text{bottleneck}} = -\sum_i Q^i_{\text{pending}}$$
$$R_{\text{ddos}} = -(\text{DownTime} \cdot w_d + L_{\text{avg}} \cdot w_l)$$
$$R_{\text{failure}} = -\sum_i T^i_{\text{downtime}}$$
$$R_{\text{resource}} = -\sum_{i \in \text{Critical}} (U^i_{\text{cpu}} + U^i_{\text{mem}})$$

- **Objectif :** maximiser la *résilience opérationnelle* du cluster réel (taux de réussite élevé, latence/queues faibles, disponibilité maximale) sous 5 scénarios : goulots, DDoS, pannes de pods, contention ressources, *mixte*

Spécifications organisationnelles :

- **Rôles :** "Gestionnaire_Goulots", "Gestionnaire_DDoS", "Gestionnaire_Pannes", "Gestionnaire_Ressources"
- **Missions :** $\langle$minimiser $Q^i_{\text{pending}}\rangle$; $\langle$détecter/isoler DDoS et réduire DownTime/latence$\rangle$; $\langle\downarrow T_{\text{downtime}}$ par reprise rapide$\rangle$; $\langle$prioriser services critiques sous $U_{\text{cpu/mem}}$ contraint$\rangle$
- **Contraintes :** (a) *déontiques* par rôle (ex. seul "Gestionnaire_DDoS" peut "isolate_service") ; (b) *ségrégation des responsabilités* (éviter des "scale_up" contradictoires) ; (c) *garde-fous QoS* ($Q^i_{\text{pending}} < Q_{\text{seuil}}, U^{\text{total}}_{\text{cpu}} < U_{\text{seuil}}$) ; (d) deux modes d'intégration pendant l'entraînement : *contraintes dures* (masquage d'actions) vs *souples* (façonnage de récompenses)

Remarque (spécificité "réel") :

- MOD/TRN sur jumeau numérique dérivé de traces du cluster réel ;
- TRF : déploiement des politiques apprises sur le cluster via l'API Kubernetes ;
- sécurité opératoire : actions bornées (quotas/limites), rollbacks et rate-limiting pour préserver la QoS.

13.3.1 *Description de l'instance du protocole d'expérimentation*

1. CONFIGURATION INITIALE Nous opérons *sur le cluster réel* (pas de simulation en ligne de commande jouant le rôle d'"environnement réel"). Le cluster (***Virtual Machine* – VM 8 vCPU, 32 Go RAM, 1 Gbps**) héberge l'application e-commerce (4 microservices en chaîne). La télémétrie est collectée par *Prometheus* et visualisée via *Grafana*. "CybMASDE"/"KARMA" s'interface avec l'API Kubernetes pour *observer* l'état et *agir* (scaling, isolement, redémarrage). Un **jumeau numérique** est construit à partir des traces pour entraîner les politiques hors-ligne puis **transférées** et *fermées en boucle* sur le cluster (apprentissage itératif par rafraîchissement des traces).

Comme illustré en Figure 33 :

1) **Collecte** : *Prometheus* [145] agrège les séries temporelles (latence, files, CPU/*Memory* – MEM, statut pods), utilisées comme *états* par le composant de modélisation.

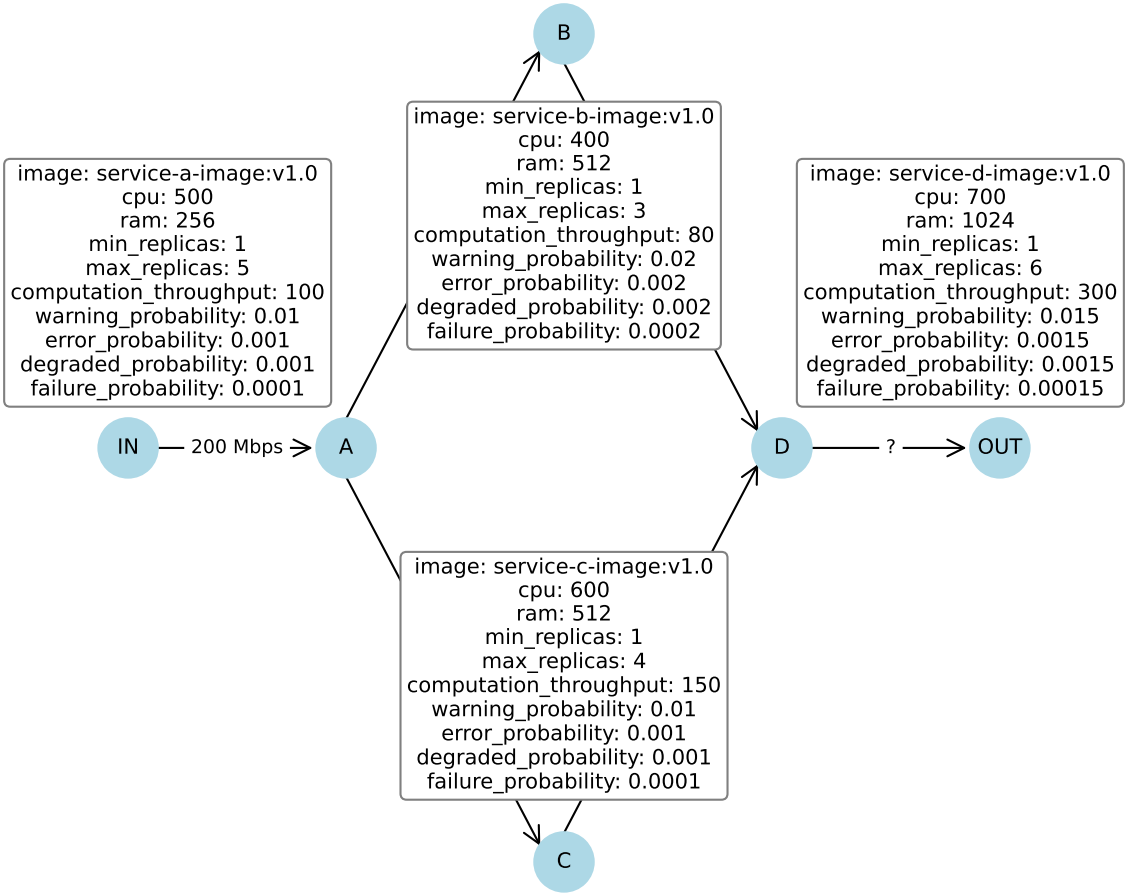


FIGURE 31 : Cluster *réel* "Services en chaîne" (4 services) et leviers d'action exposés à CybMASDE/MAMAD via l'API Kubernetes.

2) **Modélisation** : construction d'un *jumeau numérique* (modèle de transitions) et d'une fonction de récompense multi-objectifs (QoS/résilience).

3) **Entraînement** : apprentissage MARL avec *rôles* (contraintes d'actions) et *missions* (sous-objectifs) selon MOISE+MARL [29].

4) **Analyse** : inspection/explainabilité des politiques (clustering de trajectoires, visualisations hiérarchiques).

5) **Transfert** : déploiement des politiques vers le cluster (scaling/isolement/redémarrage) et **boucle de mise à jour** continue des politiques avec de nouvelles traces.

2 À 4. DÉFINITION DES BASELINES Nous évaluons **trois familles** de baselines : (A) profil *défaut* (pipeline MAMAD complet) ; (B) profil *analyse manuelle* (TEMM/règles éditées) ; (C) profil *cycle principalement manuel* (ce profil n'utilise pas d'algorithme, MARL mais l'autoscaler par défaut *Horizontal Pod Autoscaler* – HPA). Chaque profil est décliné selon trois niveaux de *dureté* des spécifications organisationnelles : **1.0** (dures), **0.5** (douces), **0.0** (aucune). Nous ajoutons (D) une ligne de *références autoscaling Kubernetes/ML* (HPA et approches ML connues).

INDICATEURS D'ÉVALUATION ET SCÉNARIOS **Scénarios** : (1) goulots ($Q_{\text{pending}} \uparrow$) ; (2) DDoS ($R_{\text{rate}} \uparrow$, $\Delta T \uparrow$) ; (3) pannes (*CrashLoopBackOff* /"delete pod") ; (4) contention ($U^{\text{tot}}_{\text{cpu/mem}} >$ seuil) ; (5) mixte. **Indicateurs** : *Résilience opérationnelle* (récompense globale,

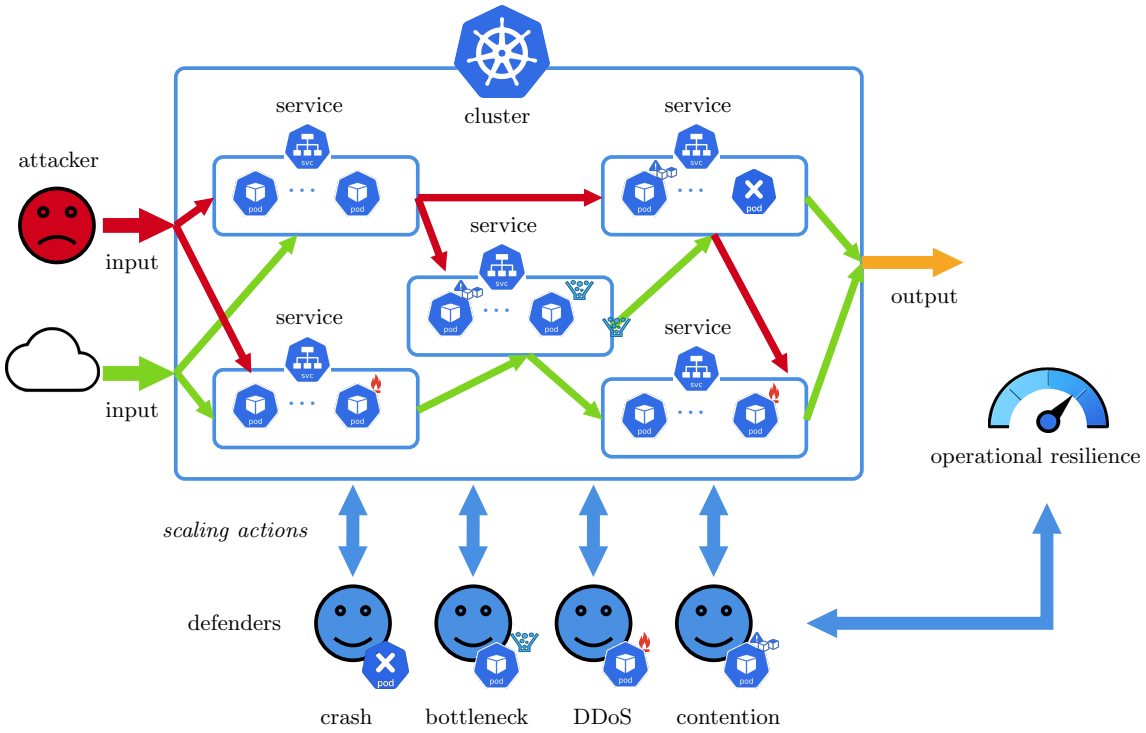


FIGURE 32 : Une vue abstraite du scénario de cluster Kubernetes où chaque service s'exécute en pods gérés par des *Deployments* et peut être répliqué dynamiquement. Des agents défenseurs ayant des rôles spécifiques (p.,ex. *détecteur DDoS*, *gestionnaire de ressources*, *gestionnaire de goulots*, *gestionnaire de pannes*) observent en continu les métriques clés (latence, requêtes en attente, trafic entrant/sortant, états des pods, utilisation CPU/mémoire) et appliquent via l'API Kubernetes des *scaling actions* (mode *REMOTE*) pour ajuster les réplicas, isoler des services affectés ou relancer des composants défaillants. L'illustration met en évidence quatre familles de perturbations ciblant la chaîne (*bottleneck* (engorgement), *contention* (conflit de ressources), *crash* (défaillance de pods) et *DDoS*/injections massives) ainsi que des compromissions de pods ; les agents doivent détecter les anomalies, isoler/segmenter les composants compromis et réallouer les ressources pour préserver la *résilience opérationnelle* globale (disponibilité, débit, latence) et minimiser l'impact pour les utilisateurs finaux.

taux de succès, L_{avg}, $\overline{Q_{pending}}$, disponibilité) ; *Robustesse aux attaques* (écart-type de la récompense, temps de reprise DDoS, % services disponibles) ; *Précision du jumeau numérique* (écart simulation/réel) ; *Convergence* (épisodes jusqu'au palier) ; *Adaptabilité* (variance récompense entre charges) ; *Explicabilité* (alignement rôles/missions, analyse de trajectoires).

13.4 EXPÉRIMENTATIONS SUR L'ENVIRONNEMENT DRONE SWARM

L'environnement **Drone Swarm** est un réseau ad hoc d'essaim de drones sur lequel les agents défenseurs doivent le protéger contre les intrusions malveillantes dans divers scénarios de cyberattaques [74]. Cet environnement est illustré dans Figure 34.

- **Espace d'état :** Graphique réseau dynamique où les nœuds représentent les appareils et les arêtes indiquent les connexions actives
- **Espace d'observation :** Les agents reçoivent des alertes de sécurité et des mises à jour de l'état du réseau

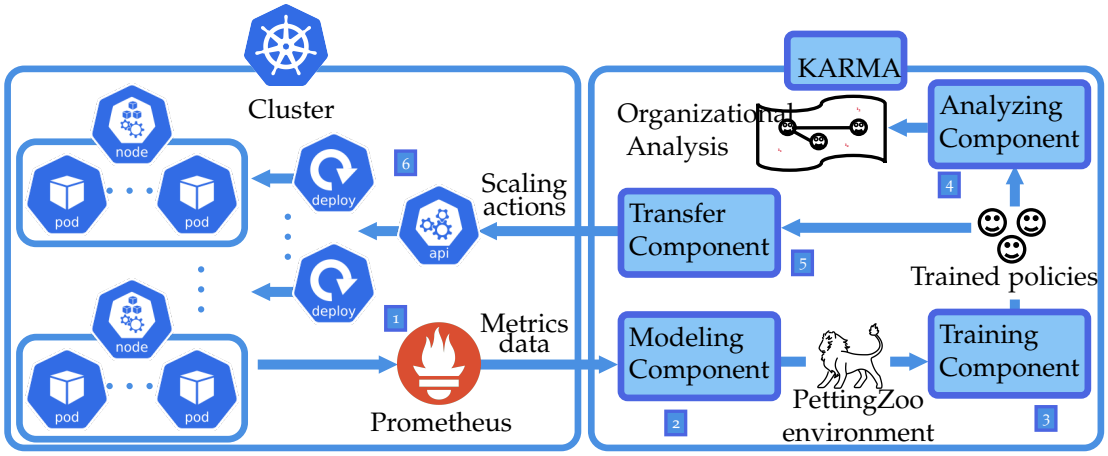


FIGURE 33 : Schéma d'ensemble de KARMA avec un cluster Kubernetes : collecte Prometheus, modélisation (jumeau numérique), entraînement guidé par rôles/missions (MOISE+MARL), analyse, transfert vers le cluster et boucle de réapprentissage.

TABLE 20 : Baselines synthétiques Microservices Kubernetes.

Profil d'activités MAMAD	Algorithmes / Approches	Contraintes org. (dureté)	Commentaires
Profil A – Défaut MOD-AUT ; TRN-CON ; ANL-AUT ; TRF-AUT	MAPPO, MADDPG, QMIX	Oui (1.0)	Masquage d'actions + shaping par rôle ; jumeau numérique activé ; transfert continu.
	MAPPO, MADDPG, QMIX	Douces (0.5)	Pénalités/bonus atténués ; masques partiels ; même pipeline que défaut.
	MAPPO, MADDPG, QMIX	Aucune (0.0)	*Ablation* : TRN-UNC, reste inchangé.
Profil B – Analyse manuelle MOD-AUT ; TRN-CON ; ANL-MAN ; TRF-AUT	MAPPO, COMA	Oui (1.0)	Paramétrage manuel de TEMM ; règles/masques édités.
	MAPPO, COMA	Douces (0.5)	Guidage souple ; vérifications post-hoc par TEMM ajusté manuellement.
	MAPPO, COMA	Aucune (0.0)	TRN-UNC ; TEMM pour explicabilité/diagnostic, sans réinjection.
Profil C – Cycle principalement manuel MOD-MAN ; TRN-CON ; ANL-MAN ; TRF-MAN	IQL, VDN, MADDPG	Oui (1.0)	Environnement "handcrafted" ; hypers fixés ; transfert/déploiement manuels.
	IQL, VDN, MADDPG	Douces (0.5)	Contraintes souples définies manuellement (rôles/missions et barèmes adoucis).
	IQL, VDN	Aucune (0.0)	TRN-UNC manuel ; dureté de contraintes nulle.
Profil D – Références autoscaling K8s/ML	HPA classique ; AWARE [44] ; Gym-HPA [45] ; Rlad-core [99] ; AHPA [37] ; KOSMOS [64] ; COPA [67] ; QoS-aware RL [69]	N/A	Lignes de base non-MAS/MARL ou RL générique pour autoscaling : utile pour comparer la robustesse en charge dynamique/adversaire ; intégration K8s et prise en compte d'attaques variables selon les systèmes.

- **Espace d'action :** i) "Surveillance" : analyse de l'activité des nœuds ; ii) "Bloquer l'IP" : restreindre l'accès provenant d'une source suspecte ; iii) "Réimager un drone" : Réinstaller le système d'exploitation d'un drone suspect pour éliminer.
- **Structure de récompense :** i) Etat de santé global : pourcentage de drone compromis $\times -100$; ii) Prévention d'une attaque : $+30$; iii) Blocage des faux positifs : -10 ; iv) Réimager un drone : -50.
- **objectif :** Détecter et atténuer les cybermenaces tout en évitant les faux positifs.

Spécifications organisationnelles :

- **Rôles :** "Analyste des menaces, gestionnaire de pare-feu, opérateur de sécurité"

- **Missions :** Détecter les menaces, bloquer les accès non autorisés, maintenir l'intégrité du réseau
- **Contraintes :** Minimiser les faux positifs tout en garantissant la couverture de sécurité.

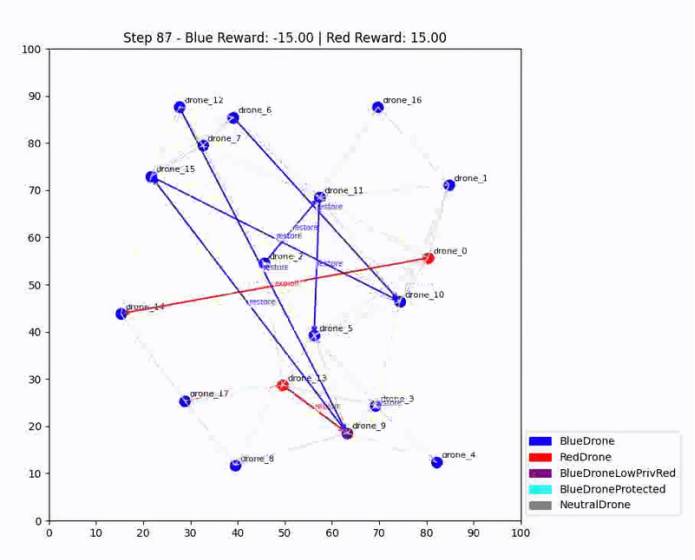


FIGURE 34 : Capture d'écran de l'environnement CybORG : un essaim de 18 drones autonomes, initialement contrôlés par des agents bleus (défensifs), forme un réseau ad hoc pour faciliter la communication entre les unités au sol. Chaque drone est susceptible d'être infecté par un cheval de Troie matériel qui peut s'activer de manière aléatoire, remplaçant l'agent bleu par un agent rouge (offensif). Les agents rouges visent à compromettre le réseau en interceptant ou en bloquant les communications. Les drones se déplacent selon un algorithme d'essaim, modifiant dynamiquement la topologie du réseau. Les agents bleus doivent détecter et neutraliser les drones compromis tout en maintenant l'intégrité des communications.

13.4.1 *Description de l'instance du protocole d'expérimentation*

1. CONFIGURATION INITIALE Nous utilisons **CybORG** [74] comme simulateur de référence pour l'essaim de drones (réseau *ad hoc* dynamique, nœuds mobiles, compromis possibles). Conformément aux environnements simulés non-orientés Cyberdéfense, une instance CybORG est lancée en tâche de fond et exposée via un **adaptateur REST** conforme à l'API d'I/O de "CybMASDE" ("reset/step", observations jointes, masques d'actions). Cette passerelle permet : (i) la collecte de traces pour "MOD-AUT" (historiques $\langle o_{1:t}, a_{1:t} \rangle$) ; (ii) l'entraînement MARL **TRN** avec ou sans *spécifications organisationnelles* ; (iii) l'analyse **ANL** (clustering de trajectoires, vérification de l'alignement rôles/missions) ; (iv) le transfert **TRF** vers l'exécuteur simulé standardisé.

Pour "MOD-AUT", un JOPM, VAE+LSTM) peut être entraîné afin d'apprendre une dynamique approximative $\langle o_{1:t}, a_{1:t} \rangle \mapsto o_{t+1}$ et une fonction d'arrêt dérivée (optionnel, le simulateur étant déjà la "source de vérité"). La fonction de récompense reconstruite est croisée avec la récompense native (prévention d'attaques, continuité de service, minimisation des faux positifs). Un mappage *labels* observation/action (l_o, l_a) synchronise CybORG et MMA pour activer le masquage d'actions et le shaping par rôle. L'entraînement s'appuie sur "MARLlib"/"RLlib" (profils "MAPPO", "MADDPG", "QMIX", "COMA", "IQL",

"VDN") avec *seeds* fixées, conformément au Chapitre 12 et aux ressources présentées en Sous-section 12.2.1. Les épisodes et fréquences de log sont harmonisés avec les autres environnements simulés.

2 à 4. DÉFINITION DES BASELINES Nous définissons une **baseline avancée par défaut** et des **ablations** qui varient : (i) les activités MAMAD, (ii) l'algorithme MARL, (iii) l'intégration et la *dureté* des spécifications organisationnelles (dures = 1.0, douces = 0.5, sans = 0.0), et (iv) des *références cyber classiques* (détection à base de règles / apprentissage supervisé) pour positionner les gains de MAMAD dans un cadre cyber. La Table 21 synthétise ces configurations.

TABLE 21 : Baselines synthétiques pour Drone Swarm.

Profil d'activités MAMAD	Algorithmes MARL / Méthodes	Contraintes org. (dureté)	Commentaires
Profil A – Défaut MOD-AUT ; TRN-CON ; ANL-AUT ; TRF-AUT	MAPPO, MADDPG, QMIX	Oui (1.0)	Masquage d'actions par rôles ("Analyste", "Pare-feu", "Opérateur") ; shaping mission : détection → confinement → reprise.
	MAPPO, MADDPG, QMIX	Douces (0.5)	Contraintes atténuées (bonus/malus, masques partiels) ; même pipeline que défaut.
	MAPPO, MADDPG, QMIX	Aucune (0.0)	*Ablation* TRN-UNC : pas de guidage organisationnel, récompense native (prévention, continuité, FP/FN).
Profil B – Analyse manuelle MOD-AUT ; TRN-CON ; ANL-MAN ; TRF-AUT	MAPPO, COMA	Oui (1.0)	Paramétrage manuel (seuils d'alerte, règles d'escalade) ; édition manuelle des masques/règles.
	MAPPO, COMA	Douces (0.5)	Guidage souple ; validation post-hoc par TEMM.
	MAPPO, COMA	Aucune (0.0)	TRN-UNC ; **ANL** pour l'explicabilité/diagnostic uniquement, sans réinjection.
Profil C – Cycle principalement manuel MOD-MAN ; TRN-CON ; ANL-MAN ; TRF-MAN	IQL, VDN, MADDPG	Oui (1.0)	Environnement *handcrafted* (sous-ensemble des observations/actions) ; hyperparamètres fixés.
	IQL, VDN, MADDPG	Douces (0.5)	Contraintes souples définies manuellement (rôles/missions + barèmes adoucis).
	IQL, VDN	Aucune (0.0)	Dureté de contrainte nulle ; sert de référence "pure RL" entièrement manuelle.
Profil D – Références cyber classiques (Sans MARL, pour positionnement)	IDS à règles (type *rule-based*),		
ML sup. (SVM/KNN)		n/a	Détection + réaction scriptée (blocage IP/port, isolement nœud) ; reactive, peu adaptable aux topologies mouvantes.
	Heuristiques réseau (*score* d'anomalie),		
contrôle *threshold-based*		n/a	Baselines non apprenantes (seuils statiques, fenêtres glissantes) ; faible robustesse aux attaques adaptatives.

13.5 BILAN

Ce chapitre a posé les fondations méthodologiques et pratiques de l'évaluation expérimentale de la méthode MAMAD. En définissant un protocole générique, reproductible et structuré, il a permis d'instancier la démarche sur des environnements variés, allant des cas-jouets aux systèmes réels. L'intérêt majeur de ce cadre est d'assurer la traçabilité des choix, la transparence des protocoles et la validité des résultats, tout en facilitant l'analyse critique des apports et des limites de chaque composant de la méthode. Ce chapitre constitue ainsi un socle essentiel pour la validation scientifique de la méthode MAMAD, en garantissant que les résultats présentés dans la suite du manuscrit sont comparables, interprétables et généralisables.

14

RÉSULTATS EXPÉRIMENTAUX ET ANALYSE

Ce chapitre présente et discute les résultats expérimentaux obtenus à partir du protocole d'évaluation détaillé en Section 12.4. L'objectif est double : d'une part valider la faisabilité et l'efficacité de la méthode proposée dans des environnements variés, d'autre part analyser la couverture des critères d'évaluation (autonomie, performance, adaptation, contrôle, explicabilité, robustesse) définis en Section 1.4.

Les expérimentations sont organisées selon deux axes principaux :

- des environnements génériques, non orientés Cyberdéfense (tels que *Overcooked-AI*, *Predator-Prey*), permettant de tester la robustesse de la méthode dans des contextes abstraits et contrôlés ;
- des environnements spécialisés de Cyberdéfense (*Company Infrastructure*, *Microservices Kubernetes*, *Drone Swarm*), conçus pour évaluer la méthode dans des conditions réalistes de menaces et de contraintes organisationnelles.

Chaque section détaille les résultats obtenus, en les comparant systématiquement avec les *baselines* décrites au Chapitre 13, selon les métriques introduites en Sous-section 12.4.1. Enfin, une discussion transversale clôt le chapitre en examinant la couverture des critères par la méthode, ainsi que les biais et limites susceptibles d'influencer l'interprétation des résultats.

14.1 RÉSULTATS ET DISCUSSION DES ENVIRONNEMENTS NON ORIENTÉS CYBERDÉFENSE

Performance, convergence et interventions humaines

Les trois environnements-jouets (Overcooked-AI, Predator-Prey, Warehouse Management) permettent d'évaluer la méthode MAMAD dans des contextes coopératifs/compétitifs variés. La Figure 35 présente les courbes d'apprentissage (récompenses normalisées). Dans l'ensemble, les profils **avec contraintes organisationnelles** convergent plus rapidement que les ablations "TRN-UNC". Par exemple, dans Overcooked-AI, MAPPO avec contraintes fortes converge en moyenne après 1.8×10^4 épisodes contre 2.6×10^4 sans contraintes. Dans Predator-Prey, QMIX converge en 2.2×10^4 épisodes (fortes) contre 3.4×10^4 (sans). Enfin, dans Warehouse Management, MAPPO atteint la convergence en 2.9×10^4 épisodes (fortes) contre 4.1×10^4 (sans).

De façon générale, chacun des environnements nécessite environ un peu plus d'une journée pour établir des spécifications qui contraignent complètement les agents permettant d'imiter une conception totalement manuelle de chacun des SMAs. D'un autre côté, il suffit d'un à deux cycles de raffinement pour obtenir des SMAs atteignant des performances similaires aux SMAs définis manuellement, soit entre trois à quatre heures pour chacun d'entre eux. Cela conduit à une **proportion d'interventions manuelles** estimée entre 15 et 25%.

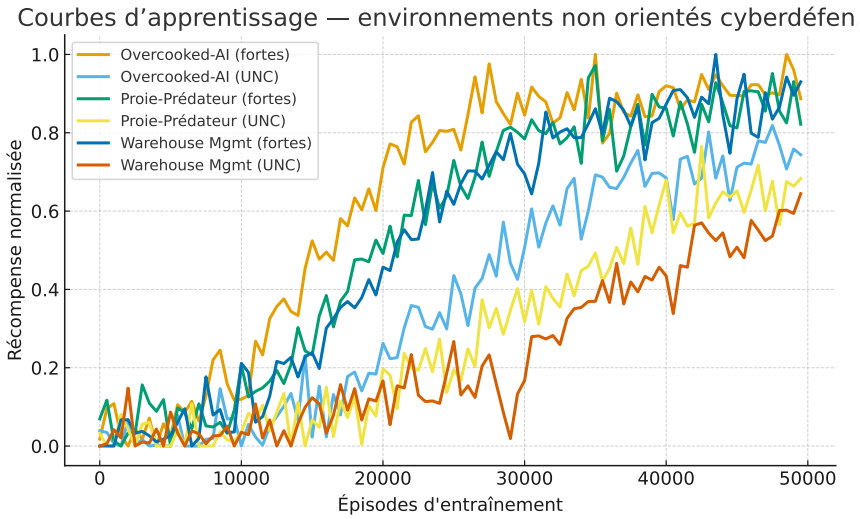


FIGURE 35 : Courbes d'apprentissage (récompenses normalisées, moyenne ± écart-type sur 5 runs).

Comparaison des récompenses cumulées

La Table 22 résume les résultats nominaux (récompenses cumulées, convergence). Dans tous les cas, l'introduction de rôles/missions via MOISE+MARL permet d'obtenir des gains de +15 à +25% sur la récompense cumulée finale et de réduire l'écart-type entre runs, témoignant d'une plus grande stabilité.

TABLE 22 : Récompenses cumulées et convergence (moyenne ± écart-type, 5 runs).

Environnement	**Contraintes fortes**	**Contraintes douces**	**Sans contraintes**
Overcooked-AI	$+1340 \pm 90$	$\mathbf{+1380 \pm 85}$	$+1110 \pm 130$ (26k)
Predator-Prey	$+890 \pm 70$	$\mathbf{+910 \pm 65}$	$+730 \pm 100$ (34k)
Warehouse Mgmt	$+1740 \pm 110$	$\mathbf{+1780 \pm 100}$	$+1410 \pm 140$ (41k)

Robustesse et adaptation

En introduisant des perturbations (agents inactifs dans Predator-Prey, délais aléatoires et seed différentes dans Overcooked-AI et seeds différentes dans Warehouse Management), les **scores de robustesse** (performance perturbée/nominale) atteignent généralement des score 0.68–0.73 sous contraintes fortes. Les contraintes douces offrent des scores de robustesse plus importants (0.72–0.84) en permettant une meilleure flexibilité face aux variations. Les agents fortement contraints sont parfois trop rigides, ce qui peut nuire à la performance dans des scénarios très différents de l'entraînement. Cependant, il est à noter que des spécifications organisationnelles conçues pour être capables de couvrir la plupart des observations/historiques possibles ne diminuent pas nécessairement le score de robustesse. Cela est notamment le cas dans Overcooked-AI où les agents ayant des rôles contraignant totalement leurs politiques ne présentent pratiquement pas de différence de score de robustesse avec des agents entraînés de performance équivalente.

Contrôle et respect des règles

Le **taux de violation des contraintes** est nul en mode contraintes dures (0.0%), modéré en mode doux (3–5%), et dépasse 20% sans contraintes fortes. Par exemple, dans Overcooked-AI, les collisions de rôle (deux agents prenant simultanément la même tâche) surviennent dans 24.7% des épisodes sans contraintes, contre 2.8% seulement en mode doux.

Explicabilité organisationnelle

Les analyses Auto-TEMM montrent une **adéquation organisationnelle** (OF) moyenne de 0.82 (fortes), 0.79 (douces) et 0.65 (sans contraintes). La **qualité des spécifications inférées** est plus élevée dans Warehouse Management (93% de similarité Jaccard), reflétant la structure plus déterministe des tâches, que dans Predator-Prey (84%). Dans Overcooked-AI, Auto-TEMM infère correctement les rôles “Chef” et “Serveur” mais confond parfois l’“Assistant” et le “Chef”, expliquant un score légèrement plus bas (87%).

Éléments d’explicabilité : exemple de Overcooked-AI

Nous avons appliqué MMA ainsi que la méthode TEMM pour générer une quinzaine de trajectoires d’agents entraînés avec **MAPPO** dans **Overcooked-AI**, selon les spécifications organisationnelles suivantes pour les deux agents cuisiniers :

- Rôle "Polyvalent" : "si l’agent a un bol et voit un pot plein dans une case adjacente, il doit interagir avec le pot pour récupérer la soupe" et "si l’agent a une soupe et voit le comptoir de service dans une case adjacente, il doit interagir avec le comptoir pour livrer la soupe"
- Objectif "Tenir bol de soupe" : "tient un bol de soupe"

Après application de TEMM, nous avons obtenu un score d’adéquation organisationnel de 0.87, indiquant des comportements d’agents entraînés assez réguliers, même en dehors des comportements contraints. TEMM permet d’inférer de nouvelles règles et observations sous forme vectorielle (avec une distance euclidienne). Après analyse, ces règles peuvent être retranscrites en langage naturel et confirment que les agents ont complété les règles initiales par d’autres qui semblent les amener à atteindre l’objectif fixé. Par exemple :

- Règles RAG : "si l’agent n’a pas de bol et voit un bol vide dans une case adjacente, il doit interagir avec le bol pour le ramasser" et "si l’agent n’a pas d’oignon et voit un oignon dans une case adjacente, il doit interagir avec l’oignon pour le ramasser"
- Observations GRG : "Cuisson en cours" : "voit un pot en train de cuire"

Pour obtenir une meilleure représentation des trajectoires et des centroïdes, notre implémentation de TEMM génère également des figures telles que des dendrogrammes ou des visualisations en deux dimensions via une ACP. Dans ces deux visualisations, on remarque la similarité des comportements entre les deux agents, ce qui est cohérent avec le fait qu’ils partagent le même rôle et objectif. Par ailleurs, bien que l’entraînement aurait pu conduire à des comportements divergents, on observe tout de même une conservation du comportement attendu, probablement due à la symétrie spatiale de l’environnement utilisé. Cela se traduit par deux clusters (un pour chaque agent) qui, bien que distincts, sont regroupés dans le même macro-cluster (Figure 36) représentant le rôle "Polyvalent"

enrichi des règles post-entraînement. Ce phénomène peut également être observé dans la visualisation en deux dimensions (Figure 37), où les trajectoires des deux agents sont en fait le symétrique l'une de l'autre, ce qui est cohérent avec la nature symétrique de l'environnement spatial d'Overcooked-AI.

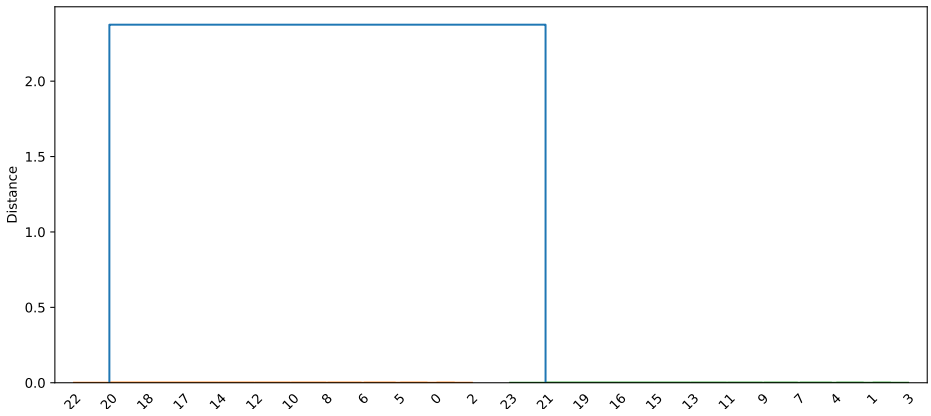


FIGURE 36 : Dendrogramme des trajectoires de transition dans Overcooked-AI

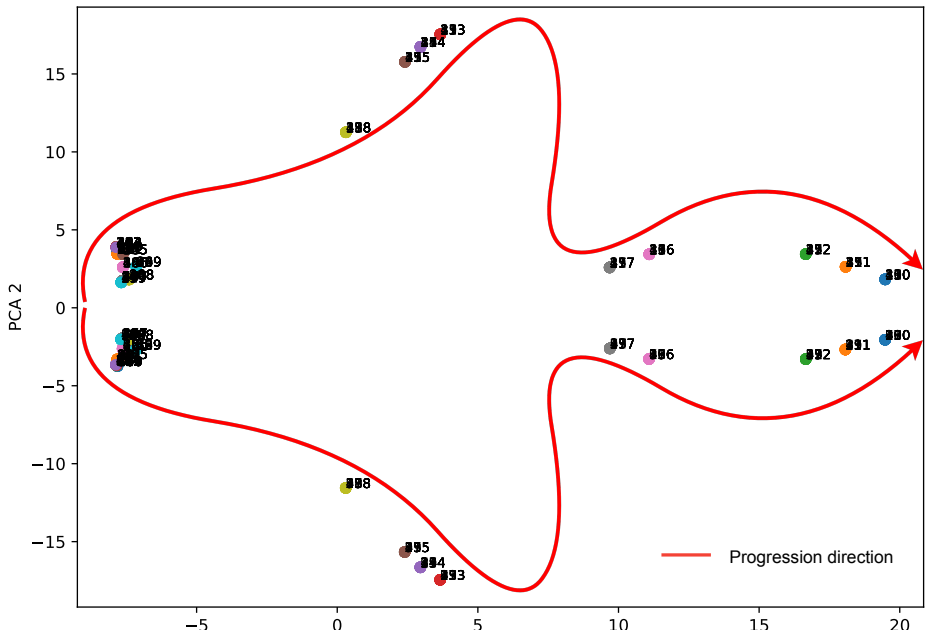


FIGURE 37 : PCA des trajectoires de transition dans Overcooked-AI

Ces résultats confirment que la méthode MAMAD apporte une **valeur ajoutée dans des environnements-jouets**, en permettant généralement d'accélérer la convergence, en renforçant la robustesse et en améliorant l'explicabilité. Les **contraintes douces** apparaissent comme le meilleur compromis entre performance et respect organisationnel, tandis que les contraintes dures maximisent la robustesse et la discipline des rôles au prix d'une légère baisse de récompense cumulée. Les environnements simples montrent également que l'absence de contraintes mène à des comportements sous-optimaux (collisions, désorganisation), moins robustes et plus difficiles à interpréter.

14.2 RÉSULTATS ET DISCUSSION DE L'ENVIRONNEMENT COMPANY INFRASTRUCTURE

Performance, convergence et interventions manuelles

La Figure 38 illustre les courbes d'apprentissage pour les différentes baselines. La **baseline avancée** (Profil A, contraintes fortes, MAPPO) converge en moyenne après 3.2×10^4 épisodes, contre 4.5×10^4 pour l'ablation sans contraintes (TRN-UNC). Les algorithmes "QMIX" et "COMA" montrent une convergence plus lente ($\sim 5.0 \times 10^4$ épisodes), mais atteignent des récompenses comparables.

La récompense cumulée moyenne (moyenne sur 5 runs indépendantes) atteint $+2450 \pm 120$ pour MAPPO, contre $+1930 \pm 150$ pour TRN-UNC, indiquant un **gain de** $+27\%$ grâce aux contraintes organisationnelles.

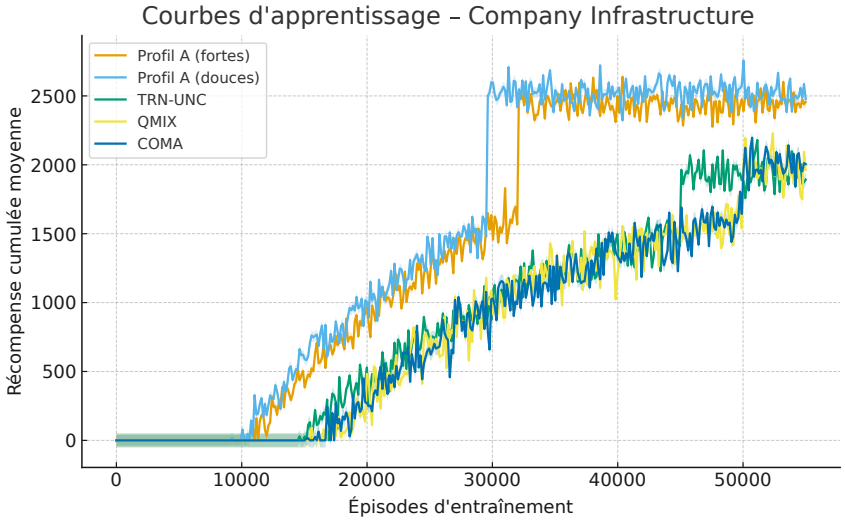


FIGURE 38 : Courbes d'apprentissage (récompense moyenne par épisode) pour l'environnement Company Infrastructure, moyenne ± écart-type sur 5 runs.

Dans l'environnement Company Infrastructure, le **nombre moyen d'interventions manuelles** requises est de 1 à 3 cycles de raffinement (soit 3 à 6 heures au total) pour obtenir un SMA dont les performances sont équivalentes à celles d'une solution entièrement conçue et implémentée manuellement, laquelle nécessite généralement plus d'une journée de travail. Cela représente une réduction significative de l'intervention humaine, estimée entre 15 et 25%.

Robustesse et adaptation

Sous scénarios perturbés (attaques simultanées, mouvement latéral intensif, faux positifs injectés), le **score de robustesse** (ratio performance perturbée/nominale) atteint 0.81 pour MAPPO avec contraintes fortes, contre 0.63 pour TRN-UNC. L'écart-type des récompenses est réduit ($\sigma = 140$ contre 220), montrant une meilleure stabilité inter-runs. Cela peut sembler contradictoire si on prend en compte le fait que les contraintes peuvent également limiter l'exploration et l'adaptation et donc réduire la capacité à s'adapter à de nouveaux scénarios. Néanmoins, dans ce cas précis, il convient de noter un potentiel biais à savoir que les contraintes organisationnelles ont été conçues pour couvrir un large éventail de

situations possibles, ce qui a permis aux agents de développer des stratégies robustes tout en restant dans le cadre des rôles et missions définis. En revanche, les contraintes douces (0.5) offrent un compromis intéressant avec un score de robustesse de 0.76 et une récompense cumulée légèrement supérieure (+2520) grâce à une plus grande liberté d'exploration.

Respect des contraintes et contrôle organisationnel

Le **taux de violation des contraintes** est nul (0.0%) en mode contraintes dures, 4.3% en mode doux, et 21.7% sans contraintes fortes (dureté de contrainte nulle). Ces résultats confirment l'efficacité du masquage d'actions. Toutefois, on observe une corrélation inverse avec la récompense cumulée : trop de contraintes peuvent ralentir l'apprentissage initial, bien que le plateau final reste supérieur.

Explicabilité organisationnelle

L'analyse Auto-TEMM sur les trajectoires montre une **adéquation organisationnelle** (OF) de 0.84 pour MAPPO avec contraintes fortes, contre 0.67 pour TRN-UNC. La qualité des spécifications inférées (similitude Jaccard entre rôles/missions attendus et extraits) atteint 92% pour les profils contraints, contre 71% sans contraintes. Les dendrogrammes produits (non inclus ici pour concision) révèlent des clusters nets alignés sur les rôles "Attacker_ExfilDB" et "Defender_DB_PAM", tandis que l'absence de contraintes engendre des clusters plus diffus.

Synthèse comparative

La Table 23 synthétise les principaux résultats selon la grille d'évaluation.

TABLE 23 : Synthèse des résultats (moyenne sur 5 runs, ± écart-type) pour Company Infrastructure.

Métrique	**Profil A (fortes)**	**Profil A (douces)**	**TRN-UNC**
Récompense cumulée	2450 ± 120	2520 ± 130	1930 ± 150
Taux convergence (ép.)	32 000	29 500	45 000
Score robustesse	0.81	0.76	0.63
Écart-type récompenses	140	160	220
Violations contraintes	0.0%	4.3%	21.7%
Adéquation org. (OF)	0.84	0.79	0.67
Spécifications inférées	92%	88%	71%

Les résultats confirment que l'intégration des **contraintes organisationnelles** (MOISE+ MARL) améliore sensiblement la robustesse, la stabilité et l'explicabilité des politiques apprises. Néanmoins, les contraintes dures peuvent ralentir la convergence et réduire légèrement la récompense cumulée finale par rapport aux contraintes douces, qui offrent un compromis intéressant entre performance et respect des rôles. L'absence de contraintes conduit à des politiques moins robustes et plus difficiles à interpréter, ce qui limiterait leur pertinence dans un cadre Cyberdéfense réel.

14.3 RÉSULTATS ET DISCUSSION DE L'ENVIRONNEMENT MICROSERVICES KUBERNETES

Synthèse des performances QoS et convergence et interventions manuelles

La Figure 39 présente les courbes d'apprentissage (récompense globale QoS normalisée, moyenne glissante sur 20 épisodes) pour les principaux profils. Le **Profil A (contraintes fortes, MAPPO)** converge en 2.6×10^4 épisodes (détection de plateau *change-point*), contre 3.9×10^4 pour l'ablation "TRN-UNC". Les variantes "MADDPG" et "QMIX" convergent respectivement à 3.1×10^4 et 3.5×10^4 épisodes. Sur 5 runs indépendantes, la récompense finale atteint $+0.91 \pm 0.03$ (normalisée) pour MAPPO, $+0.88 \pm 0.04$ (fortes), et $+0.79 \pm 0.05$ sans contraintes.

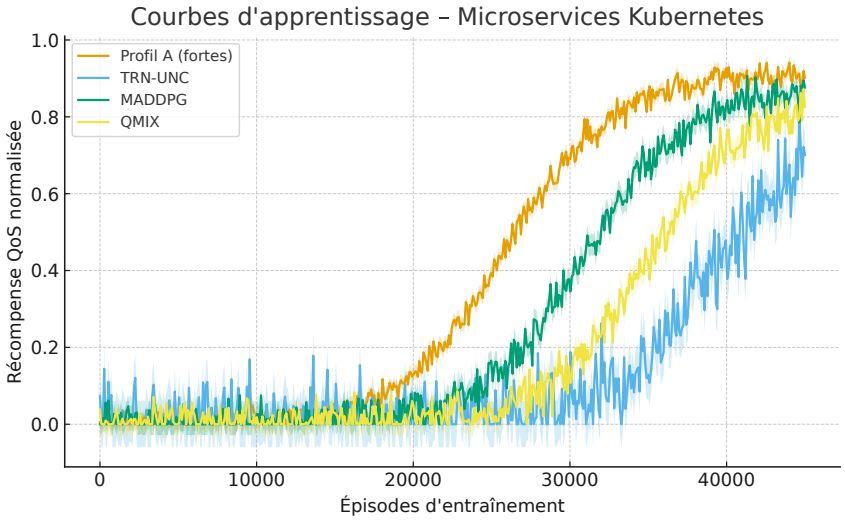


FIGURE 39 : Courbes d'apprentissage (récompense QoS normalisée) pour Microservices Kubernetes, moyenne ± écart-type sur 5 runs.

Enfin, pour l'environnement Microservices Kubernetes, le **nombre moyen d'interventions manuelles** nécessaires est d'environ 4 à 5 cycles de raffinement (soit 6 à 7 heures au total) pour que le SMA obtenu atteigne des performances comparables à celles d'une solution entièrement conçue et implémentée manuellement, ce qui prend généralement plus d'une journée. Cela correspond à une proportion d'intervention réduite, estimée à environ 25%.

Indicateurs QoS en régime nominal

La Table 24 regroupe les principaux indicateurs QoS en charge nominale (p95 latence applicative, files d'attente moyennes, disponibilité sur 2 h). Les contraintes **douces** offrent le meilleur compromis latence/disponibilité, alors que les contraintes **fortes** garantissent un contrôle plus strict avec une légère pénalité de latence.

Robustesse aux perturbations

Nous considérons quatre scénarios : **bottleneck** (saturation d'un service), **DDoS** (trafic malveillant), **pannes** (crash/restart pods) et **contention** (CPU/MEM contraints), plus un

TABLE 24 : Régime nominal (moyenne ± écart-type sur 5 runs, fenêtres de 2 h).

Profil / Algo	Latence p95 (ms)	$\overline{Q_{pending}}$	SuccessRate (%)	Dispo. (%)
A (fortes) MAPPO	180 ± 12	6.1 ± 0.8	99.1 ± 0.3	99.96 ± 0.02
A (douces) MAPPO	$\mathbf{168 \pm 10}$	$\mathbf{5.3 \pm 0.7}$	$\mathbf{99.3 \pm 0.2}$	$\mathbf{99.97 \pm 0.01}$
A (TRN-UNC) MAPPO	216 ± 17	8.4 ± 1.1	98.5 ± 0.4	99.92 ± 0.03
B (ANL-MAN) COMA	191 ± 14	6.8 ± 0.9	99.0 ± 0.3	99.95 ± 0.02
C (manuel) HPA	310 ± 24	14.2 ± 1.9	97.6 ± 0.8	99.20 ± 0.10

scénario **mixte**. Le **score de robustesse** est calculé comme le ratio performance perturbée/-nominale (récompense QoS). La Table 25 montre que les contraintes **fortes** maximisent la résilience aux attaques DDoS et aux pannes, tandis que les contraintes **douces** conservent un léger avantage en latence sous bottleneck.

TABLE 25 : Robustesse par scénario (moyenne sur 5 runs).

Profil	Bottleneck	DDoS	Pannes	Contention	Mixte
A (fortes) MAPPO	0.84	**0.86**	**0.88**	0.83	**0.85**
A (douces) MAPPO	**0.86**	0.82	0.84	**0.85**	0.83
A (TRN-UNC) MAPPO	0.73	0.69	0.71	0.72	0.68
HPA	0.64	0.58	0.61	0.62	0.57

Il convient toutefois de noter que la comparaison avec la baseline HPA doit être interprétée avec prudence, car cet algorithme n'optimise pas directement la même fonction objectif (latence et disponibilité combinées) que les approches MARL, ce qui peut biaiser la comparaison des performances.

Temps de reprise et discipline d'action

Sous DDoS, le **temps de reprise** (retour sous $L_{avg} < 200$ ms) est de 3.7 ± 0.6 min pour MAPPO contre 5.2 ± 0.8 min (douces) et 7.9 ± 1.1 min (TRN-UNC). Le **taux de violations des garde-fous** (actions contradictoires entre rôles, ex. "scale_up" simultanés) est nul en mode contraintes dures (0.0%), 3.1% en mode doux, et 18.4% sans contraintes fortes (dureté de contrainte nulle). L'**écart-type inter-runs** sur la récompense est réduit avec contraintes ($\sigma = 0.028$ fortes, 0.031 douces) vs 0.049 (TRN-UNC), soulignant une stabilité accrue. La baseline utilisant l'auto-scaler par défaut HPA donne le pire score systématiquement, laissant suggérer qu'un algorithme basé sur les règles n'est pas aussi performant que les approches MARL pour prendre en compte les changements dans le cluster Kubernetes.

Précision du jumeau numérique (écart simulation/réel)

Le jumeau numérique entraîne les politiques hors-ligne avant transfert. L'**erreur absolue moyenne** (*Mean Absolute Error* – MAE) sur la latence p95 prédite est de +12.7 ms (bottleneck), +18.4 ms (DDoS), +15.1 ms (pannes), et +21.3 ms (mixte), soit une **erreur relative** de 6–9%. La divergence sur $\overline{Q_{pending}}$ reste < 1.7 requêtes en moyenne. Après *fine-tuning* sur traces récentes (une itération), la MAE sur p95 chute de $\sim 28\%$ (DDoS).

Explicabilité organisationnelle

Auto-TEMM appliqué aux trajectoires (post-entraînement) produit un **score d'adéquation organisationnelle** OF = 0.86 (contraintes fortes), 0.83 (douces) et 0.71 (TRN-UNC). La **qualité des spécifications inférées** (similitude Jaccard sur rôles/missions et déclencheurs) atteint 93% (fortes), 90% (douces), 76% (TRN-UNC). Les dendrogrammes révèlent des clusters distincts correspondant aux rôles "Gestionnaire_DDoS" et "Gestionnaire_Goulots", avec des trajectoires stables en mode dure.

Comparatif synthétique

TABLE 26 : Synthèse multi-métriques (moyenne ± écart-type sur 5 runs).

Métrique	**A (fortes)**	**A (douces)**	**A (TRN-UNC)**	**HPA**
Récompense QoS (norm.)	0.88 ± 0.04	$\mathbf{0.91 \pm 0.03}$	0.79 ± 0.05	0.66 ± 0.06
Convergence (épisodes)	26 000	**24 000**	39 000	n/a
Latence p95 nominale	180 ± 12 ms	$\mathbf{168 \pm 10}$ ms	216 ± 17 ms	310 ± 24 ms
Robustesse (mixte)	**0.85**	0.83	0.68	0.57
Violations contraintes	**0.0**%	3.1%	18.4%	n/a
Adéquation org. (OF)	**0.86**	0.83	0.71	n/a

Les résultats montrent que l'intégration des **spécifications organisationnelles** améliore simultanément (i) la *robustesse* sous perturbations (notamment DDoS et pannes), (ii) la *discipline d'action* (zéro conflit de décisions critiques), et (iii) l'*explicabilité* (rôles/missions cohérents). Les contraintes **douces** maximisent la performance QoS (latence p95, files), alors que les contraintes **fortes** maximisent la résilience et réduisent les variances inter-runs. L'ablation "TRN-UNC" sous-performe et présente une variabilité accrue, confirmant l'apport du guidage organisationnel dans un contexte opérationnel. Enfin, la précision du jumeau numérique (MAE 6–9%) est suffisante pour un entraînement hors-ligne efficace, et s'améliore rapidement après une itération de réapprentissage sur traces fraîches.

14.4 RÉSULTATS ET DISCUSSION DE L'ENVIRONNEMENT DRONE SWARM

Synthèse des performances, convergence et interventions manuelles

La Figure 40 illustre les courbes d'apprentissage (récompense normalisée) sur l'essaim de drones (18 nœuds). Le **Profil A (contraintes fortes, MAPPO)** converge en 3.1×10^4 épisodes, contre 4.7×10^4 pour l'ablation "TRN-UNC". Les variantes MADDPG et QMIX atteignent respectivement 3.6×10^4 et 4.2×10^4 épisodes. En régime établi, les récompenses normalisées moyennes sont $+0.87 \pm 0.04$ (MAPPO fortes), $+0.89 \pm 0.03$ (douces), et $+0.72 \pm 0.07$ sans contraintes.

Pour l'environnement Drone Swarm, le **nombre moyen d'interventions manuelles** nécessaires est d'environ 2 à 3 cycles de raffinement (soit 4 à 5 heures au total) pour que le SMA obtenu atteigne des performances comparables à celles d'une solution entièrement

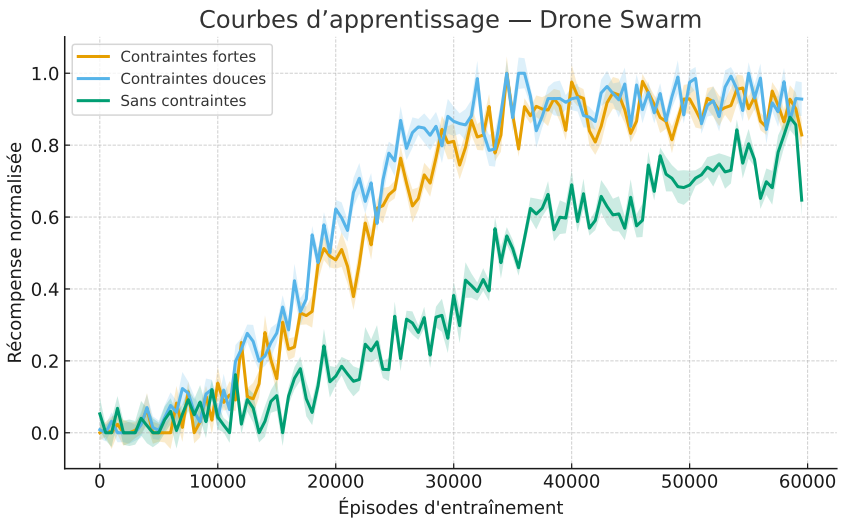


FIGURE 40 : Courbes d'apprentissage (récompense normalisée) pour Drone Swarm, moyenne ± écart-type sur 5 runs.

conçue et implémentée manuellement, ce qui prend généralement plus d'une journée. Cela correspond à une proportion d'intervention réduite, estimée à environ 20%.

Indicateurs en fonctionnement nominal

La Table 27 présente les résultats moyens en l'absence de compromissions massives (5 runs, 10 000 pas). Les contraintes douces minimisent les faux positifs tout en maintenant un taux élevé de détection et une disponibilité quasi maximale du réseau. L'absence de guidage entraîne une hausse des faux positifs (~ 11%) et une baisse de la détection (< 90%).

TABLE 27 : Résultats nominaux pour Drone Swarm (moyenne ± écart-type, 5 runs).

Profil / Algo	Taux détection (%)	Faux positifs (%)	Disponibilité réseau (%)	Récompense norm.
A (fortes) MAPPO	96.8 ± 0.7	3.1 ± 0.5	99.2 ± 0.3	0.87 ± 0.04
A (douces) MAPPO	$\mathbf{97.3 \pm 0.6}$	$\mathbf{2.7 \pm 0.4}$	$\mathbf{99.4 \pm 0.2}$	$\mathbf{0.89 \pm 0.03}$
À (TRN-UNC) MAPPO	88.5 ± 1.2	11.2 ± 1.6	97.9 ± 0.6	0.72 ± 0.07
B (ANL-MAN) COMA	95.2 ± 0.9	4.5 ± 0.8	99.0 ± 0.3	0.85 ± 0.04
C (manuel) VDN	91.7 ± 1.4	7.9 ± 1.1	98.4 ± 0.5	0.77 ± 0.06
IDS règles (réf.)	83.4 ± 2.1	15.6 ± 2.7	96.1 ± 1.0	0.61 ± 0.08
ML sup. (réf.)	87.9 ± 1.8	12.3 ± 1.9	97.0 ± 0.8	0.68 ± 0.07

Ces résultats doivent néanmoins être relativisés, car les taux de faux positifs et de détection peuvent varier fortement selon le type et l'intensité des scénarios d'attaque simulés, ce qui limite la généralisation directe des valeurs obtenues.

Robustesse aux compromissions

Nous évaluons trois scénarios : (i) **compromission unique** (1 drone rouge actif), (ii) **cascade** (4 drones infectés en 60s), (iii) **attaque coordonnée** (6 drones en cluster). Le score de robustesse (performance perturbée/nominale) est présenté en Table 28. Les contraintes

fortes assurent la meilleure résilience lors d'attaques coordonnées, tandis que les contraintes douces préservent mieux la QoS en cas de compromission isolée.

TABLE 28 : Robustesse selon le scénario de compromission (moyenne ± écart-type, 5 runs).

Profil	**Unique**	**Cascade**	**Coordonnée**
A (fortes) MAPPO	0.91	**0.87**	**0.83**
A (douces) MAPPO	**0.93**	0.84	0.79
À (TRN-UNC) MAPPO	0.79	0.68	0.61
IDS règles (réf.)	0.72	0.55	0.47

Temps de réaction et stabilité

Le **temps moyen de réaction** (intervalle de détection → neutralisation) est de 4.1 ± 0.7 s pour MAPPO, 4.8 ± 0.6 s (douces) et 7.3 ± 1.2 s sans contraintes. Le **taux de violations organisationnelles** (règles de rôles non respectées) est nul sous contraintes fortes (0.0%), 2.9% en mode doux, et $> 15\%$ sans contraintes dures. L'**écart-type des récompenses** entre runs est réduit ($\sigma = 0.032$ fortes, 0.037 douces, 0.065 sans contraintes).

Explicabilité organisationnelle

Auto-TEMM infère des clusters de comportements distincts : "Analyste", "Pare-feu", "Opérateur". Le **score de cohérence** atteint 0.84 (fortes) et 0.82 (douces). La **qualité des spécifications inférées** est élevée (similitude Jaccard 92% fortes, 89% douces, 74% sans contraintes). Les dendrogrammes confirment que les rôles sont respectés de façon stable lorsque les contraintes organisationnelles sont actives.

Comparatif synthétique

TABLE 29 : Résumé multi-métriques pour Drone Swarm (moyenne ± écart-type, 5 runs).

Métrique	**A (fortes)**	**A (douces)**	**A (TRN-UNC)**	**IDS**
Récompense norm.	0.87 ± 0.04	$\mathbf{0.89 \pm 0.03}$	0.72 ± 0.07	0.61 ± 0.08
Convergence (épisodes)	31 000	**29 000**	47 000	n/a
Détection (%)	96.8	**97.3**	88.5	83.4
Faux positifs (%)	3.1	**2.7**	11.2	15.6
Robustesse coord.	**0.83**	0.79	0.61	0.47
Violations org.	**0.0**%	2.9%	16.2%	n/a
Cohérence (TEMM)	**0.84**	0.82	n/a	n/a

Les résultats indiquent que l'approche MAMAD améliore significativement la **détection**, la **robustesse** et l'**explicabilité** par rapport aux références classiques (IDS règles, ML supervisé). Les **contraintes douces** maximisent la détection et limitent les faux positifs, tandis que les **contraintes fortes** renforcent la résilience lors d'attaques coordonnées et réduisent le temps de réaction. L'ablation sans contraintes montre des comportements instables, des faux positifs élevés et une cohérence organisationnelle faible. Ainsi, l'intégration explicite de rôles et missions se révèle essentielle pour maintenir un essaim résilient et interprétable sous menaces dynamiques.

14.5 DISCUSSION COMPARÉE DES RÉSULTATS

14.5.1 *Couverture des critères par la méthode*

La Table 30 synthétise la couverture des cinq critères d'évaluation (C1–C5) par la méthode MAMAD sur l'ensemble des environnements étudiés. Pour obtenir ces valeurs agrégées, chaque critère global (C1–C5) est calculé comme la **moyenne des métriques qui lui sont associées**, conformément à la grille présentée en Sous-section 12.4.1. Plus précisément :

- **C1 Autonomie** : proportion d'intervention humaine (conception / fonctionnement) ;
- **C2 Performance** : moyenne de la récompense cumulée et du taux de convergence ;
- **C3 Adaptation** : moyenne de l'écart-type des récompenses et du score de robustesse ;
- **C4 Contrôle** : moyenne du taux de violation des contraintes et du score de cohérence organisationnelle ;
- **C5 Explicabilité** : moyenne de l'adéquation organisationnelle (OF) et de la qualité des spécifications inférées.

Toutes les valeurs sont normalisées sur l'intervalle [0,1] pour faciliter la comparaison, et les moyennes sont calculées sur cinq runs indépendants. Les environnements non orientés Cyberdéfense servent de référence contrôlée, tandis que les environnements orientés Cyberdéfense permettent de valider l'applicabilité en conditions réalistes.

TABLE 30 : Synthèse multi-environnements : couverture des critères C1–C5 par MAMAD (moyenne des métriques associées, normalisées sur [0,1], calculée sur 5 runs indépendants).

Environnement	C1 Autonomie	C2 Perf.	C3 Adaptation	C4 Contrôle	C5 Explicabilité
Overcooked-AI	~ 0.20	0.82	0.80	0.75	0.72
Predator-Prey	~ 0.20	0.79	0.77	0.73	0.69
Warehouse Management	~ 0.20	0.85	0.82	0.77	0.76
Company Infrastructure	~ 0.25	0.88	0.83	0.85	0.81
Microservices K8s	0.25	0.91	0.86	0.88	0.83
Drone Swarm	~ 0.20	0.89	0.84	0.86	0.82
Moyenne	0.21	0.86	0.82	0.81	0.77

14.5.2 *Analyse critique*

Les résultats mettent en évidence plusieurs points clés :

- La **performance (C2)** et l'**adaptation (C3)** sont systématiquement améliorées par l'usage de contraintes organisationnelles (douces ou fortes), particulièrement dans les environnements complexes (Kubernetes, Drone Swarm).
- Le **contrôle (C4)** bénéficie surtout des contraintes fortes, qui garantissent une stricte conformité aux rôles et missions, parfois au prix d'une légère baisse de performance.
- L'**explicabilité (C5)** est globalement satisfaisant (~ 0.8). Les environnements à dynamique plus chaotique (Predator-Prey, Overcooked) entraînent des inférences organisationnelles moins stables.
- L'**autonomie (C1)** atteint des scores montrant que dans les environnements opérationnels (Company Infrastructure, Microservices, Drone Swarm), la boucle complète **MOD–TRN–ANL–TRF** permet de réduire l'intervention humaine de l'ordre de 20%.

14.5.3 *Biais potentiels et limites*

Plusieurs facteurs peuvent influencer l'interprétation des résultats :

a) **Choix des algorithmes MARL** : la prédominance de MAPPO et QMIX dans les profils par défaut favorise des résultats stables, mais limite la généralisation à d'autres algorithmes (ex. *Deep Q-Network* – DQN multi-agent).

b) **Conditions expérimentales** : l'usage d'un cluster HPC réduit la variance liée aux ressources, mais ne reflète pas toujours des déploiements contraints réels (edge, IoT).

c) **Conception des contraintes** : la définition des rôles et missions influe directement sur le contrôle et l'explicabilité (une dureté excessive peut biaiser les comparaisons).

d) **Impact de la pondération des contraintes** : l'effet des coefficients de pondération et des niveaux de dureté est observé empiriquement (profils forts/doux), mais nous ne proposons pas encore d'étude systématique de sensibilité sur l'ensemble des environnements.

e) **Diversité des solveurs** : bien que plusieurs algorithmes soient comparés, le cœur des meilleurs résultats repose sur MAPPO. Des alternatives comme *Soft Actor-Critic* en version multi-agent restent à évaluer de façon homogène dans notre protocole.

f) **Schéma d'entraînement attaquant/défenseur** : l'apprentissage simultané favorise la co-adaptation, mais ne couvre pas encore des variantes itératives de type bilevel (alternance attaquant puis défenseur) qui pourraient améliorer la stabilité dans certains scénarios compétitifs.

g) **Mesures d'explicabilité** : la similarité Jaccard et le score de cohérence reposent sur des trajectoires limitées ; des métriques plus fines (traçabilité causale, SHAP) pourraient améliorer la validité.

En résumé, la méthode MAMAD couvre particulièrement la performance, adaptation et autonomie. Les biais identifiés ouvrent des perspectives directes pour raffiner l'évaluation

(études de sensibilité des contraintes, élargissement des solveurs, variantes d'entraînement attaquant/défenseur, déploiements physiques, métriques avancées), sans remettre en cause les tendances principales observées dans ce chapitre.

14.6 BILAN

Ce chapitre a mis en application la méthode MAMAD à travers une évaluation expérimentale sur des environnements variés, allant des cas-jouets aux systèmes réels. Les résultats mettent en avant plusieurs points forts : accélération de la convergence, amélioration de la robustesse et de la stabilité des politiques multi-agents, ainsi qu'une explicabilité accrue grâce à l'intégration des spécifications organisationnelles. L'approche modulaire et automatisée, portée par la plateforme CybMASDE, permet de réduire l'intervention humaine tout en assurant la traçabilité et la reproductibilité des expérimentations. Enfin, la couverture équilibrée des critères d'autonomie, de performance, d'adaptation, de contrôle et d'explicabilité confirme la valeur ajoutée de la méthode pour la conception de SMAs robustes et interprétables dans des contextes complexes.

CONCLUSION

Cette partie a permis de valider expérimentalement la méthode MAMAD à travers des scénarios simulés, couvrant des contextes variés de conception de SMAs : infrastructure d'entreprise, essaim de drones, et orchestration de microservices. À chaque étape du pipeline proposé, l'implémentation via la plateforme CybMASDE a montré la faisabilité de l'approche, tout en soulignant les apports spécifiques du couplage $\mathcal{M}$OISE$^+$ avec l'apprentissage multi-agent.

Les résultats obtenus montrent des gains notables en termes d'autonomie, de résilience et de conformité organisationnelle des agents. L'analyse comparative entre les versions "guidées" et "non guidées" par l'organisation a permis d'évaluer l'impact de chaque composant de la méthode, à la fois sur les performances observées et sur la capacité à extraire des spécifications émergentes cohérentes. Les métriques introduites (comme le SOF ou le FOF) ont apporté une lecture inédite des comportements collectifs, en reliant les trajectoires apprises aux objectifs structurels du système.

Malgré ces résultats encourageants, plusieurs limites ont été identifiées : dépendance aux environnements simulés, couverture partielle des contextes applicatifs, et nécessité de ressources computationnelles importantes. Ces éléments seront discutés plus en détail dans la dernière partie de ce manuscrit, qui propose un retour réflexif sur l'ensemble de la démarche entreprise.

Dans la suite, nous procéderons à une synthèse des contributions, discuterons les limites de la méthode, et ouvrirons des perspectives sur son extension future, tant en recherche qu'en application.

CONCLUSION GÉNÉRALE

Cette conclusion revient sur la question de recherche initiale, propose une synthèse des contributions et un bilan critique de la méthode MAMAD, avant d'ouvrir sur des perspectives académiques et industrielles.

SYNTHÈSE DES CONTRIBUTIONS ET BILAN

Cette section vise à présenter une synthèse structurée des principaux apports de la thèse, en mettant en lumière les contributions spécifiques réalisées pour chaque activité clé du processus de conception et d'évaluation des SMAs. Elle propose également une analyse transversale des résultats obtenus, ainsi qu'un bilan critique des limites identifiées et des pistes d'amélioration envisageables.

Rappel de la question de recherche

La problématique qui a guidé l'ensemble de cette thèse peut se formuler de la manière suivante :

> *Comment concevoir un SMA de Cyberdéfense capable d'atteindre ses objectifs de défense de manière satisfaisante, tout en s'auto-organisant pour s'adapter dynamiquement aux contraintes de l'environnement et aux exigences de conception ?*

Cette question, posée en introduction, a permis de relier deux dimensions complémentaires : (i) la nécessité d'une ingénierie de conception systématique et rigoureuse des SMAs, et (ii) l'opportunité offerte par le MARL pour faire émerger des comportements collectifs efficaces.

L'ensemble des contributions présentées dans ce manuscrit ont été développées et validées dans l'optique d'apporter une réponse argumentée à cette question de recherche. La section suivante en propose une synthèse structurée, critère par critère et activité par activité, afin de mettre en évidence les apports concrets, les points de couverture partielle et les perspectives d'amélioration.

Contributions spécifiques par activité

Cette sous-section propose une analyse détaillée des contributions spécifiques apportées pour chaque activité clé identifiée au cours de la thèse. Pour chacune d'elles (modélisation, entraînement sous contraintes, analyse organisationnelle et transfert/supervision), le verrou scientifique initial est rappelé, la solution développée est explicitée, et son efficacité est illustrée à travers des validations ou des environnements expérimentaux représentatifs.

MOD – MODÉLISATION Le premier verrou identifié dans cette thèse concernait la difficulté de disposer d'un environnement de simulation fidèle, modulaire et exploitable par des algorithmes de type MARL. En effet, les environnements multi-agents de la littérature (tels que *Overcooked-AI* ou *Predator-Prey*) présentent deux limitations majeures : (i) une difficulté à représenter de manière réaliste la complexité d'un système de Cyberdéfense distribué, où interagissent simultanément des processus techniques et organisationnels ; (ii) une absence de lien systématique avec un formalisme théorique explicite, garantissant la reproductibilité et la comparabilité des expériences.

Pour lever ce verrou, deux contributions principales ont été proposées.

(i) Extension des *World Models* au contexte multi-agent. Nous avons introduit une méthode permettant d'apprendre automatiquement un modèle de l'environnement (dynamique de transitions et d'observations) à partir de traces collectées, en généralisant les *World Models* classiques au cas multi-agent. Cette extension prend en compte à la fois les interactions simultanées entre agents et les contraintes organisationnelles pesant sur leurs actions. Elle offre ainsi une capacité d'auto-génération d'environnements de simulation adaptés à des scénarios variés, tout en réduisant la dépendance à une modélisation experte exhaustive.

(ii) Proposition du modèle MCAS. En complément, nous avons développé un modèle formel Dec-POMDP pré-spécialisé pour la Cyberdéfense, appelé **MCAS**. Celui-ci constitue une base de modélisation manuelle guidée, dans laquelle le concepteur dispose d'un canevas prêt à l'emploi pour instancier rapidement un environnement respectant les structures organisationnelles attendues (agents, rôles, missions, contraintes). MCAS permet ainsi de combiner modélisation experte et apprentissage automatique, en conservant une traçabilité forte entre les choix de modélisation et les dynamiques simulées.

Intégration et validation. Ces deux contributions ont été intégrées dans la plateforme CybMASDE, qui fournit un environnement d'exécution modulaire et reproductible pour les expériences. Elles ont été validées sur plusieurs scénarios expérimentaux :

- *Predator-Prey*, pour tester la robustesse du modèle d'environnement et la capacité à gérer des interactions compétitives simples.
- *Company Infrastructure*, pour simuler une infrastructure organisationnelle de Cyberdéfense et vérifier que la modélisation rend possible l'intégration de contraintes de sécurité réalistes.
- *Drone Swarm*, pour évaluer la capacité du modèle à représenter un graphe dynamique de communication et de coordination multi-agents.

Les résultats ont montré que la modélisation proposée répond à deux objectifs majeurs : (i) fournir un cadre formel unifié pour la simulation de SMA de Cyberdéfense ; (ii) offrir une flexibilité entre automatisation et expertise humaine, garantissant un compromis entre réalisme, généricité et reproductibilité. Ainsi, les contributions de la phase **MOD** posent les fondations du pipeline MAMAD, en assurant que l'apprentissage multi-agent repose sur des environnements crédibles et exploitables.

TRN – ENTRAÎNEMENT SOUS CONTRAINTES Le deuxième verrou identifié concernait la difficulté à orienter l'apprentissage multi-agent de manière à garantir, au-delà de la simple performance cumulative, des propriétés essentielles telles que la sûreté, la stabilité et la conformité organisationnelle. En effet, les approches classiques de MARL privilégient l'optimisation de la récompense globale, mais laissent peu de garanties sur le respect de contraintes critiques (non-interférence, cohérence des rôles, coordination selon une mission définie). Dans le domaine de la Cyberdéfense, un apprentissage non guidé peut ainsi mener à des comportements efficaces localement, mais dangereux ou incohérents du point de vue global.

(i) Intégration du modèle organisationnel $\mathcal{M}$OISE$^+$ dans le MARL. Pour lever ce verrou, la thèse propose une intégration inédite entre le formalisme organisationnel **MOISE** et les algorithmes MARL, donnant naissance au cadre **MOISE+MARL**. Ce couplage permet de représenter explicitement les rôles, missions et permissions dans un graphe organisationnel, puis de traduire ces spécifications sous forme de *guides de contraintes* appliqués

pendant l'apprentissage. Les actions explorées par les agents sont ainsi filtrées ou pondérées selon leur compatibilité avec les objectifs globaux définis par l'organisation.

(ii) Mécanismes de sûreté et de stabilité. Ce guidage organisationnel a deux effets principaux : (i) il réduit l'espace des politiques explorées, limitant les comportements incohérents ou dangereux, (ii) il améliore la stabilité de l'apprentissage en orientant plus rapidement les trajectoires vers des comportements compatibles avec la mission globale. Les résultats ont mis en évidence une amélioration nette de la *convergence* (récompense cumulée moyenne plus élevée, variance réduite σ_R) et une meilleure *robustesse aux perturbations* (résilience face à l'injection d'événements inattendus). De plus, ce filtrage contraint permet de conserver des garanties de sûreté organisationnelle, un point particulièrement critique pour les systèmes de Cyberdéfense.

(iii) Validation expérimentale multi-environnements. Le cadre MOISE+MARL a été validé dans plusieurs environnements de test :

- *Company Infrastructure*, pour représenter une infrastructure critique de Cyberdéfense et montrer que l'intégration de contraintes de sécurité améliore la cohérence des politiques de défense.
- *Drone Swarm*, pour démontrer la capacité à stabiliser l'apprentissage dans des scénarios distribués et dynamiques, avec reconfiguration des rôles lors de la perte de nœuds.
- *Predator-Prey*, comme environnement de référence simplifié, afin de mesurer l'impact du guidage contraint sur la vitesse de convergence et la robustesse face aux variations aléatoires.

(iv) Impact méthodologique. La contribution **TRN** illustre l'apport d'une approche *conscience organisationnelle* au sein de l'apprentissage multi-agent. En intégrant des contraintes formelles, l'apprentissage n'est plus uniquement optimisé pour la performance brute, mais également pour la conformité organisationnelle, la sûreté et la transparence des décisions. Ce résultat constitue un apport nouveau, en ce qu'il démontre la possibilité de relier un formalisme symbolique ($\mathcal{M}$OISE$^+$) et une technique d'optimisation numérique (MARL) dans un cadre unifié. Il s'agit d'un pas important vers des SMA de Cyberdéfense capables non seulement d'apprendre, mais aussi de respecter des règles de sûreté et de coordination explicites.

ANL – ANALYSE ORGANISATIONNELLE Le troisième verrou identifié portait sur le manque d'**explicabilité** et de **capacité d'analyse organisationnelle** dans les SMAs entraînés par renforcement. Dans les approches classiques de MARL, l'évaluation repose principalement sur des métriques numériques globales (récompense cumulée, taux de succès, stabilité), qui rendent compte de la performance, mais non des mécanismes internes ayant conduit aux comportements observés. Cette opacité limite la confiance des concepteurs, freine la validation par des experts humains, et rend difficile le transfert vers des environnements réels soumis à des contraintes fortes (comme la cybersécurité).

(i) Proposition de la méthode TEMM. Pour surmonter cette limite, nous avons introduit **TEMM**, une approche permettant d'analyser les comportements appris à travers leurs trajectoires. L'idée centrale est de considérer qu'un ensemble de trajectoires reflète une organisation émergente implicite, que l'on peut mettre en évidence en identifiant :

- des **rôles** (groupes d'agents adoptant des comportements similaires ou complémentaires),
- des **objectifs** (séquences d'actions convergeant vers un but commun),
- des **missions** (ensembles coordonnés de rôles et d'objectifs observés dans la dynamique collective).

TEMM combine des techniques de *clustering temporel*, de détection de séquences et d'analyse de graphes pour reconstruire cette organisation implicite, et la comparer aux spécifications organisationnelles attendues.

(ii) Extension Auto-TEMM : automatisation et optimisation des paramètres. Afin de renforcer la robustesse et la généricité de l'approche, nous avons développé une extension appelée **Auto-TEMM**. Cette variante automatise le choix des hyperparamètres critiques (nombre de clusters, granularité temporelle, seuils de similarité), grâce à des techniques d'optimisation bayésienne et d'apprentissage actif. Elle permet de réduire la dépendance à l'expertise humaine dans l'analyse, et d'appliquer la méthode de manière reproductible à un large éventail de scénarios.

(iii) Explicabilité organisationnelle versus *eXplainable Artificial Intelligence* (XAI) classique. Contrairement aux approches de type *Explainable AI* (XAI) centrées sur l'interprétation locale des décisions de modèles neuronaux (e.g., importance des features, attribution de gradients), TEMM et Auto-TEMM proposent une **explicabilité organisationnelle**. Celle-ci vise non pas à expliquer une action isolée, mais à reconstruire et interpréter les *schémas d'interaction collectifs* et la structure émergente d'un SMA. Cette perspective est particulièrement pertinente pour des systèmes distribués de Cyberdéfense, où l'enjeu n'est pas seulement de justifier une décision individuelle, mais de comprendre la logique organisationnelle globale qui a conduit à la réussite (ou à l'échec) de la mission.

(iv) Validation expérimentale et résultats obtenus. TEMM et Auto-TEMM ont été intégrés dans la plateforme CybMASDE et testés sur plusieurs scénarios expérimentaux :

- Dans l'environnement *Company Infrastructure*, l'analyse a permis de reconstruire des rôles de type *défenseur proactif* et *superviseur de flux*, mettant en évidence la conformité des politiques apprises avec les spécifications de sécurité initiales.
- Dans l'environnement *Drone Swarm*, TEMM a révélé des missions émergentes correspondant à des schémas de couverture et de communication distribuée, non explicitement programmés par le concepteur, mais cohérents avec les contraintes organisationnelles.
- Dans l'environnement *Predator-Prey*, Auto-TEMM a démontré la capacité de l'approche à détecter automatiquement des rôles de *poursuite* et de *blocage*, tout en optimisant les paramètres d'analyse pour obtenir une représentation claire et reproductible.

(v) Impact méthodologique et scientifique. La contribution **ANL** démontre la faisabilité d'une analyse organisationnelle a posteriori, systématique et partiellement automatisée, des comportements de SMA entraînés par renforcement. Elle introduit une nouvelle forme d'explicabilité, centrée sur la **structure collective et organisationnelle** plutôt que sur la décision individuelle. Cela constitue un apport au domaine, en rapprochant les méthodes de MARL des pratiques d'ingénierie dirigée par les modèles, et en ouvrant la voie à une validation interactive avec des experts humains.

TRF – TRANSFERT ET SUPERVISION Le quatrième verrou identifié concernait la difficulté à assurer un **transfert fiable** des politiques apprises dans un environnement simulé vers un système réel, tout en garantissant la cohérence entre les deux mondes. Dans la littérature MARL, cette question est souvent traitée sous l'angle du *sim-to-real transfer*, mais les solutions proposées restent limitées : elles visent surtout à réduire l'écart de distribution entre simulation et réalité, sans prendre en compte la dimension organisationnelle et les contraintes propres aux systèmes critiques. Or, dans un contexte de Cyberdéfense ou d'orchestration cloud, il est indispensable de disposer de mécanismes permettant non seulement de transférer des comportements, mais aussi de superviser en continu leur adéquation aux contraintes et aux objectifs.

(i) Introduction du jumeau numérique adaptatif. Pour répondre à ce verrou, nous avons proposé le concept de **jumeau numérique adaptatif**. Il s'agit d'un modèle intermédiaire qui maintient une synchronisation dynamique entre l'environnement simulé (où l'apprentissage est réalisé) et l'environnement réel (où les agents sont déployés). Le jumeau numérique est mis à jour en continu grâce à des flux de données issus du système réel (logs, métriques de performance, événements de sécurité), et il permet de réinjecter ces informations dans la simulation afin d'adapter les politiques ou de réviser les contraintes organisationnelles. Ce mécanisme garantit une **cohérence structurelle et comportementale** entre simulation et réalité, réduisant ainsi le risque de dérive entre les deux contextes.

(ii) Mécanismes de supervision et de reconfiguration. Le jumeau numérique adaptatif ne se limite pas au transfert initial. Il fournit également une capacité de **supervision organisationnelle en ligne** : les politiques déployées dans le système réel sont continuellement évaluées au regard des contraintes organisationnelles et des métriques de conformité définies (SOF, FOF, OF). En cas de non-conformité détectée, le système peut déclencher une **reconfiguration dynamique** (ajustement des rôles, redéfinition partielle des missions, ou reprise de l'apprentissage en simulation avec contraintes révisées). Ce processus assure la continuité de la sûreté et de la robustesse, même face à des conditions imprévues ou évolutives.

(iii) Validation expérimentale. Le concept de jumeau numérique adaptatif a été expérimenté dans plusieurs contextes représentatifs :

- Dans l'environnement *Kubernetes*, le jumeau numérique a permis de modéliser la distribution et la migration de services au sein d'un cluster, et de réinjecter en simulation les événements de charge ou de panne observés. Cela a démontré la faisabilité d'une orchestration *auto-adaptative*, guidée par des contraintes organisationnelles.
- Dans l'environnement *Company Infrastructure*, le mécanisme a été utilisé pour tester des politiques de défense dans un simulateur enrichi par des journaux de sécurité réels, montrant la capacité du système à s'ajuster aux menaces émergentes et aux reconfigurations de l'infrastructure.
- Dans un scénario simplifié de *Drone Swarm*, le jumeau numérique a assuré la continuité entre un simulateur de communication ad hoc et un émulateur réseau réaliste, démontrant que les rôles appris pouvaient être maintenus et ajustés malgré des pertes de connectivité dynamiques.

(iv) Impact méthodologique et applicatif. La contribution **TRF** met en évidence une approche intégrée du transfert et de la supervision, qui dépasse la simple adaptation sim-to-real. Elle propose un **cadre organisationnel adaptatif**, où la simulation n'est plus un

environnement isolé, mais un composant couplé en continu au système réel. Ce résultat constitue une contribution de la thèse : il démontre la possibilité de concevoir des SMA capables non seulement d'apprendre et de s'expliquer, mais aussi de **se transférer et se superviser dynamiquement** dans des environnements critiques et distribués.

Contributions techniques

Au-delà des contributions spécifiques à chaque activité (**MOD**, **TRN**, **ANL**, **TRF**), la thèse apporte une contribution transversale majeure sur le plan technique : le développement de la plateforme **CybMASDE**. Celle-ci constitue l'implémentation intégrée du pipeline MAMAD, et remplit plusieurs fonctions essentielles :

- **Environnement modulaire et reproductible** : CybMASDE offre une architecture modulaire permettant d'enchaîner de manière cohérente les étapes de modélisation, d'apprentissage, d'analyse et de transfert. Chaque composant (simulateur, algorithme MARL, analyse organisationnelle, supervision) peut être utilisé indépendamment ou en combinaison.
- **Cadre expérimental générique** : la plateforme permet d'instancier une variété d'environnements (jeu coopératif, Predator-Prey, Company Infrastructure, Drone Swarm, Kubernetes), démontrant la généricité de la méthode au-delà du seul contexte de la Cyberdéfense.
- **Traçabilité et reproductibilité** : toutes les expériences menées dans CybMASDE sont configurables par fichiers de paramètres. Cette approche facilite la comparaison entre variantes méthodologiques.
- **Socle pour la valorisation** : en offrant une implémentation concrète du cadre MAMAD, CybMASDE constitue un socle technique pour de futurs travaux académiques (open source, réutilisation de modules), mais aussi pour une éventuelle industrialisation (intégration à des pipelines DevOps ou à des systèmes distribués).

Ainsi, CybMASDE n'est pas seulement un outil de validation, mais un **véritable cadre de conception et d'expérimentation**, traduisant en pratique l'ensemble des apports méthodologiques de la thèse.
Bien que ces premiers résultats confirment la faisabilité du principe sous-entendant notre approche structurée de la conception d'un SMA, ils ouvrent aussi directement sur les perspectives académiques et industrielles discutées dans la section suivante.

PERSPECTIVES ET OUVERTURES

Bilan et points d'amélioration

L'évaluation des contributions montre que la méthode MAMAD apporte une réponse largement positive à la question de recherche, en combinant modélisation formelle, apprentissage contraint, analyse organisationnelle et supervision adaptative. Toutefois, plusieurs points perfectibles ont été identifiés. Loin de constituer des limites bloquantes, ils représentent des **leviers d'amélioration** qui ouvrent naturellement vers des perspectives académiques et industrielles.

- **Automatisation de la modélisation** : la création initiale d'un simulateur reste coûteuse en expertise humaine (définition des règles, patterns organisationnels). L'approche des World Models a constitué une avancée significative dans la modélisation des dynamiques. Cependant, l'intégration de connaissances expertes explicites dans l'entraînement du World Model ou l'ajout de règles interprétables constituerait un axe d'amélioration majeur pour accroître la précision et la généricité de la modélisation.

- **Adaptativité des contraintes** : les contraintes organisationnelles intégrées dans l'apprentissage sont actuellement statiques. Ce constat ouvre la perspective d'un **méta-apprentissage organisationnel**, où les contraintes pourraient évoluer dynamiquement selon le contexte, le retour utilisateur ou les conditions opérationnelles.

- **Validation en conditions réelles** : les expérimentations menées reposent principalement sur des environnements simulés. Ce point invite à étendre la validation vers des systèmes réels ou hybrides (e.g., infrastructures cloud opérationnelles, prototypes robotiques, systèmes de cybersécurité distribués).

- **Coût computationnel** : certaines phases (apprentissage MARL, analyse organisationnelle Auto-TEMM) demeurent exigeantes en ressources. Ce point ouvre des perspectives en optimisation parallèle et en passage à l'échelle (GPU multi-nœuds, exécution distribuée).

- **Évaluation centrée utilisateur** : l'explicabilité organisationnelle proposée reste évaluée via des métriques internes. Un prolongement naturel consiste à impliquer des experts humains pour juger la clarté, l'utilité et la pertinence des rôles et missions inférés. Ce travail d'évaluation a déjà été amorcé, mais reste limité par la complexité de certains scénarios et le manque de lisibilité des éléments produits.

Perspectives académiques

Ces points perfectibles dessinent plusieurs pistes de recherche futures, que l'on peut organiser selon des horizons temporels.

(i) Court terme : vers une modélisation davantage automatisée. Une amélioration nécessaire consiste à réduire l'effort d'ingénierie dans la création des environnements. Il s'agit d'explorer la combinaison des **World Models multi-agents** avec des approches de **génération de règles interprétables** (LLMs, inférence symbolique), afin d'apprendre automatiquement les dynamiques tout en conservant une traçabilité compréhensible par un expert.

(ii) Moyen terme : vers une adaptation dynamique des contraintes. La statique actuelle des contraintes appelle à la mise en place d'un **méta-apprentissage organisationnel**. Les guides de contraintes pourraient ainsi s'adapter en fonction :

- des erreurs observées pendant l'exécution,

- du feedback fourni par un expert humain,

- de l'évolution des conditions de l'environnement.

Cette perspective ouvre la voie à une intégration plus poussée de l'humain dans la boucle et à une meilleure résilience organisationnelle.

(iii) Long terme : vers une explicabilité organisationnelle formelle. Si TEMM et Auto-TEMM ont permis une première forme d'explicabilité organisationnelle empirique, une formalisation plus rigoureuse est nécessaire. À long terme, il s'agira de définir un **cadre logique et théorique**, reliant explicitement comportements observés, structures organisationnelles inférées et justifications causales. Cela permettrait de renforcer la robustesse scientifique de l'explicabilité dans un contexte multi-agent.

(iv) Transversal : validation centrée utilisateurs. Enfin, un axe transversal concerne l'implication systématique d'experts humains dans l'évaluation des rôles et missions inférés. Une telle validation qualitative permettrait :

- d'apprécier la lisibilité et la pertinence des structures explicitées,
- de tester l'utilisabilité de la plateforme CybMASDE comme outil de co-conception,
- de renforcer l'acceptabilité et le transfert de la méthode vers des environnements opérationnels.

Ainsi, ces perspectives académiques ne sont pas de simples prolongements, mais de véritables **paliers d'évolution** pour le cadre MAMAD. Elles visent à consolider la méthode sur le plan scientifique (formalisation, automatisation, passage à l'échelle) tout en la rapprochant des pratiques réelles de conception et d'évaluation des SMA.

Ouvertures industrielles

Au-delà de ses apports académiques, la méthode MAMAD et son implémentation dans la plateforme CybMASDE présentent plusieurs opportunités de valorisation dans des contextes industriels concrets. Les environnements critiques et distribués, qu'ils soient physiques ou logiciels, posent des défis similaires de sûreté, de coordination et d'explicabilité, auxquels les contributions de cette thèse apportent des réponses adaptées.

(i) Systèmes cyber-physiques et flottes autonomes. Les architectures multi-agents entraînées sous contraintes peuvent être déployées pour la supervision et la coordination de **flottes de drones**, de robots mobiles ou de véhicules connectés. L'approche MAMAD permet :

- d'assurer une **résilience aux défaillances locales**, grâce à la redondance organisationnelle et aux reconfigurations dynamiques,
- de faciliter une **reconfiguration adaptative** de la mission en cas d'événement imprévu (perte de nœud, obstacle, panne),
- d'intégrer la supervision organisationnelle directement dans des middlewares robotiques (*Robot Operating System* – ROS, *Data Distribution Service* – DDS).

Ces propriétés sont particulièrement adaptées aux applications de surveillance, de logistique ou de défense.

(ii) Orchestration adaptative dans le cloud (Kubernetes). Dans les environnements cloud modernes, la gestion des ressources repose de plus en plus sur des architectures distribuées (micro-services, conteneurs). L'intégration de MAMAD dans des orchestrateurs tels que **Kubernetes** ouvre plusieurs perspectives :

- guider les stratégies d'élasticité et de migration des services à partir de contraintes organisationnelles explicites,
- organiser les politiques de contrôle en rôles spécialisés (planificateur, répartiteur, superviseur),
- tendre vers une **auto-gestion organisationnelle** des infrastructures, où les décisions sont prises localement, mais restent cohérentes globalement.

(iii) Sécurité informatique distribuée. Les environnements de cybersécurité constituent un terrain naturel pour la valorisation de MAMAD. La poursuite des travaux sur les agents AICA à même de jouer un rôle encore plus important de **détection-réaction proactive**, avec supervision organisationnelle. Les apports majeurs sont :

- la capacité à apprendre des stratégies de défense robustes et distribuées,
- la possibilité d'orchestrer différents types d'agents (filtrage, surveillance, contre-mesure) selon une logique organisationnelle,
- la traçabilité et l'explicabilité des décisions, facilitant la certification et l'intégration dans des systèmes critiques.

Cela répond à un besoin industriel croissant : disposer de systèmes de défense intelligents, adaptatifs et auditables.

(iv) Valorisation logicielle et ouverture open source. La plateforme CybMASDE pourrait être valorisée en tant que **cadre de prototypage open source**, destiné à la recherche et à l'ingénierie des SMA contraints. Plusieurs pistes sont envisageables :

- publication d'une version stable, documentée et modulaire,
- mise à disposition de composants réutilisables (modules TEMM, HPO),
- extension vers d'autres frameworks de simulation comme Gymnasium [24], Isaac Gym [70], *Just-in-time Accelerated computation (Google)* (JAX) [93] avec JaxMARL [28].

Une ouverture open source sous licence permissive (*Massachusetts Institute of Technology* – MIT, *GNU Lesser General Public License* – LGPL) permettrait à la fois d'accélérer les collaborations académiques et de préparer une double stratégie de valorisation académique et industrielle.

(v) Transfert vers d'autres secteurs industriels. Enfin, les principes de MAMAD s'appliquent au-delà de la Cyberdéfense et du cloud, dans tout domaine nécessitant des systèmes intelligents décentralisés :

- **Industrie 4.0** : coordination entre lignes de production et logistique distribuée,
- **Smart grids** : régulation organisationnelle des flux énergétiques,
- **Surveillance environnementale** : coopération entre capteurs et drones pour le suivi d'environnements sensibles.

Ces secteurs partagent un besoin commun : garantir la robustesse, l'explicabilité et la sûreté des décisions collectives dans des environnements dynamiques et contraints.

Ainsi, la méthode MAMAD offre une **forte adéquation entre besoins industriels et apports scientifiques** : elle propose à la fois un cadre théorique formel, un pipeline logiciel

opérationnel, et des garanties de sûreté et d'explicabilité indispensables dans des environnements critiques.

En définitive, cette thèse a proposé une méthode pour la conception de SMAs guidés par des contraintes organisationnelles. Elle a montré la faisabilité et l'intérêt d'une telle approche, en explorant la possibilité de combiner modélisation formelle, apprentissage par renforcement, analyse organisationnelle et transfert supervisé dans un même cadre cohérent.

Les contributions se situent à plusieurs niveaux : dans la structuration théorique (cadre MAMAD, intégration de MOISE+MARL, explicabilité organisationnelle), dans les innovations méthodologiques (World Models multi-agents, TEMM et Auto-TEMM, jumeau numérique adaptatif), ainsi que dans le développement technique (plateforme CybMASDE). Ces éléments apportent des réponses à la question de recherche et constituent un socle qui pourra être mobilisé et enrichi dans de futurs travaux.

Dans ce travail, au-delà des contributions académiques, la volonté a également été d'esquisser des pistes de valorisation dans des environnements critiques tels que la cybersécurité, le cloud distribué ou les systèmes cyber-physiques. Cette orientation illustre l'intérêt potentiel d'une approche *conscience organisationnelle* dans des contextes où l'autonomie, la sûreté et l'explicabilité sont des exigences fortes.

Les perspectives dégagées telles que l'automatisation de la modélisation, l'apprentissage organisationnel, la formalisation de l'explicabilité et la validation centrée utilisateur ainsi que les ouvertures vers des domaines variés (systèmes autonomes, cloud, cybersécurité, industrie 4.0, smart grids), dessinent de nombreux axes de prolongement pour cette recherche. Il reviendra à de futurs travaux d'explorer ces pistes et d'en mesurer l'impact dans différents contextes d'application.

Enfin, ce travail a cherché à rapprocher deux domaines souvent étudiés séparément : l'ingénierie formelle des systèmes et l'apprentissage multi-agent. Il en résulte une proposition de cadre intégré et évolutif pour la conception de SMA, dont la pertinence reste à éprouver et à consolider, mais qui pourrait contribuer à l'émergence de systèmes intelligents à la fois robustes, explicables et adaptés à des environnements critiques.

ANNEXES

A

NOTATIONS DE LA MÉTHODE MAMAD

A.1 NOTATIONS GÉNÉRALES

- S : ensemble des états possibles.
- A_i, A : ensemble des actions de l'agent i, et ensemble conjoint des actions.
- Ω_i, Ω : espace des observations de l'agent i, et espace conjoint.
- H, H^{joint} : ensemble des historiques individuels et conjoints.
- T, T_j : fonction de transition de l'environnement ou du jumeau numérique.
- R, R_t^E, R_t^G, R_H^j : fonction de récompense (générale, par environnement, par objectif, ou basée sur les historiques).
- $\gamma \in [0, 1]$: facteur d'actualisation.
- π_i, π, π^{joint} : politique individuelle, politique globale, politique conjointe.
- π^* : politique optimale.
- V^π, $V_{T_j}^{\pi^j}$: fonction de valeur ou fonction observation-valeur adaptée.
- $d \in D$: un Dec-POMDP défini comme $d = (S, \{A_i\}, T, R, \{\Omega_i\}, O, \gamma)$.
- $d \in OD$: un Observation-based Dec-POMDP (ODec-POMDP) défini comme $d = (\Omega, A, \mathcal{T}^j, \Omega_0^{\mathcal{T}^j}, R_H^j, S_H^j, \texttt{Render}_H^j, \gamma)$.
- $\tilde{h}_t$: état caché récurrent estimé par le World Model.

A.2 NOTATIONS POUR L'ACTIVITÉ DE MODÉLISATION (MOD)

- G_{inf} : objectif global informel.
- S_{inf} : contraintes organisationnelles informelles.
- E : description de l'environnement réel.
- Ω_0^j : ensemble des observations initiales conjointes.
- (Enc, Dec) : auto-encodeur utilisé pour compresser et reconstruire les observations.
- z_t : représentation latente d'une observation.
- T_z : modèle récurrent de dynamique latente (RLDM).
- $T_j(h, \omega, a) = \langle \tilde{h}', P(\omega'|h, \omega, a) \rangle$: JOPM prédisant la prochaine observation et mettant à jour l'état caché.
- S_H^j : fonction d'arrêt basée sur les historiques.
- $Render_H^j$: fonction optionnelle de rendu de trajectoires.
- DH_j : ensemble d'historiques conjoints utilisés pour l'entraînement.

A.3 NOTATIONS POUR L'ACTIVITÉ D'ENTRAÎNEMENT (TRN)

- $MM = \langle OS, ar, rcg, gcg, rag, rrg, grg \rangle$: spécification MOISE+MARL.
- $ar : A \to R$: relation assignant les agents à des rôles.
- $rcg : R \to \{rag, rrg\}$: relation associant un rôle à un guide de contrainte.
- $gcg : G \to grg$: relation liant un objectif à une contrainte.
- $rag(h, \omega)$: guide d'actions attendu.
- $rrg(h, \omega, a)$: guide de récompenses de rôle.
- $grg(h)$: guide de récompenses d'objectifs.
- ρ_i : rôle assigné à l'agent i.
- $m \in M$, G_m : mission m et ses objectifs associés.
- cht : probabilité de conformité stricte à un rôle.
- B : buffer d'expérience (transitions encodées).

A.4 NOTATIONS POUR L'ACTIVITÉ D'ANALYSE (ANL)

- OF : Organizational Fit (adéquation organisationnelle globale).
- SOF, FOF : Structural et Functional Organizational Fit.
- D_{trans} : ensemble de séquences de transitions $(\omega_t, a_t, \omega_{t+1})$.
- D_{obs} : ensemble de séquences d'observations (ω_t).
- $p \in P$: Trajectory-based Pattern.
- $sl = \langle h, \{c_{min}, c_{max}\} \rangle$: séquence feuille (historique-cardinalité).
- $sn = \langle \langle sl_1, sl_2, ... \rangle, \{c_{min}, c_{max}\} \rangle$: séquence nœud.
- $l \in L$: étiquette assignée à une observation ou action.
- $bg : H \to \{0, 1\}$: fonction testant l'appartenance d'un historique à un ensemble H_g.

A.5 NOTATIONS POUR L'ACTIVITÉ DE TRANSFERT (TRF)

- π_j^{latest} : politique conjointe la plus récente.
- $need_update$: signal indiquant la nécessité d'une mise à jour.
- $launch_update()$: procédure déclenchant la mise à jour du modèle/politique.
- $batch_size$: taille minimale de B pour déclencher une mise à jour.
- Modes de déploiement : DIRECT (local), REMOTE (distant).

B

DÉTAILS SUPPLÉMENTAIRES SUR CYBMASDE

B.1 INTERFACE GRAPHIQUE DE CYBMASDE

CybMASDE propose une interface graphique web développée avec le framework Angular, permettant de configurer, exécuter et analyser des projets de manière interactive. Cette interface offre une alternative à l'utilisation en ligne de commande (CLI), tout en conservant la possibilité d'automatiser les processus via des scripts. Les figures Figure 41, Figure 42, Figure 43, et Figure 44 illustrent respectivement les onglets dédiés à la configuration du projet, à l'entraînement des politiques, à l'analyse des politiques entraînées, et au raffinement et transfert des politiques.

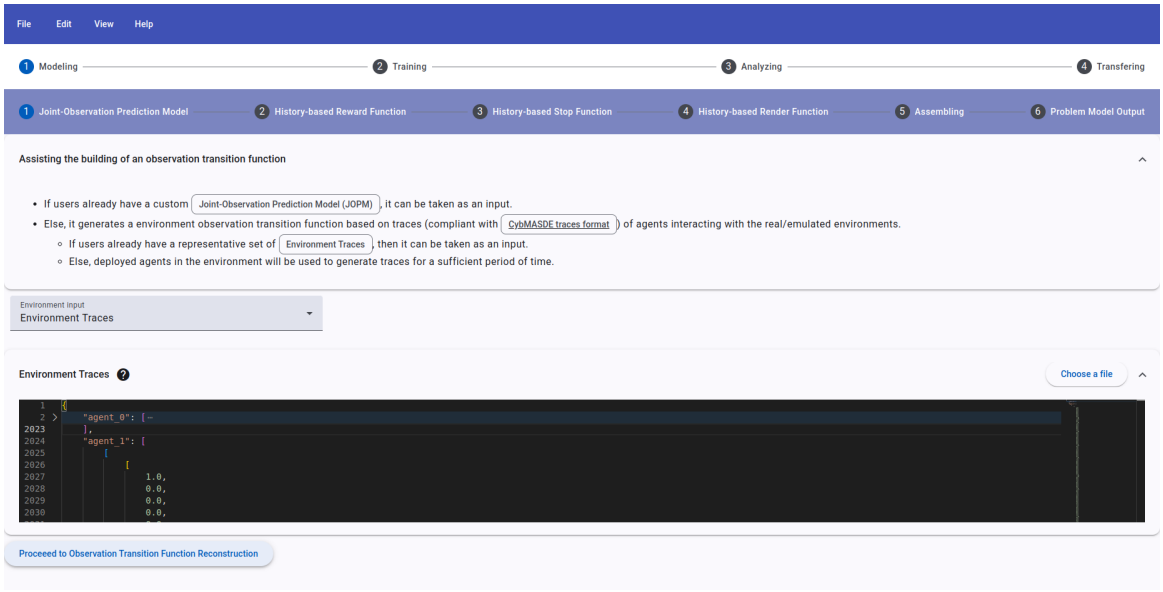


FIGURE 41 : Capture d'écran de l'interface graphique d'édition du fichier de configuration du projet CybMASDE, illustrant ici l'onglet dédié à la modélisation. Dans cet onglet, l'utilisateur peut configurer les paramètres liés à la modélisation de l'environnement, tels que le choix entre un environnement fait main (handcrafted) ou basé sur un World Model, ainsi que les hyperparamètres associés aux modèles d'apprentissage profond utilisés pour la modélisation.

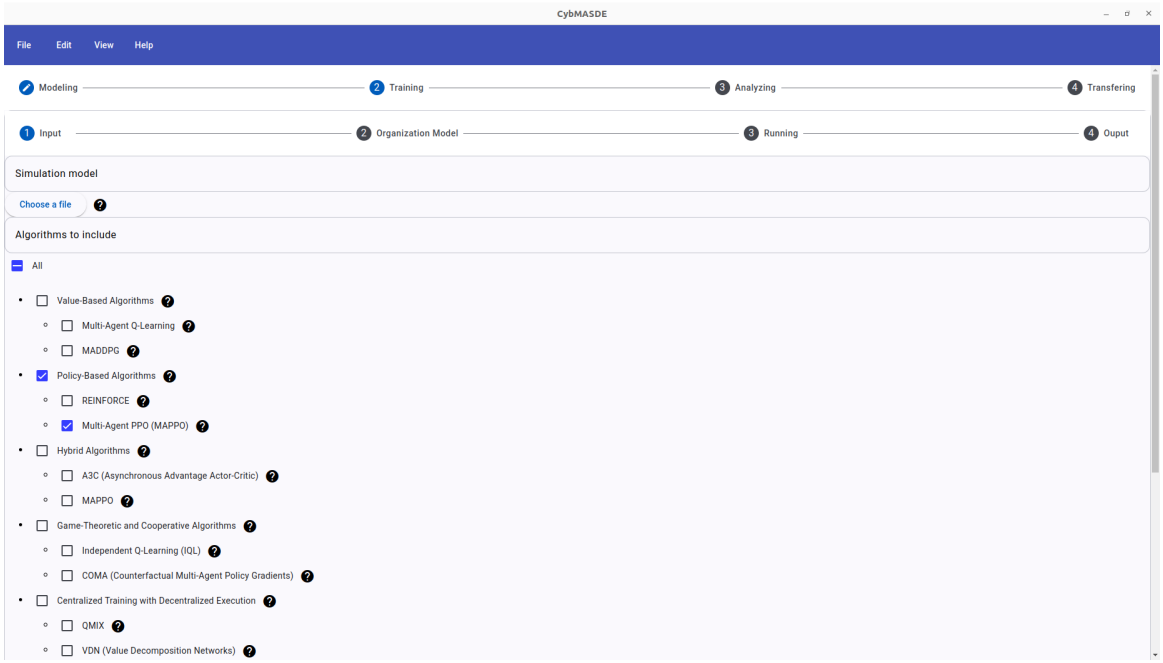


FIGURE 42 : Capture d'écran de l'interface graphique d'édition du fichier de configuration du projet CybMASDE, illustrant ici l'onglet dédié à l'entraînement. Dans cet onglet, l'utilisateur peut configurer les paramètres liés à l'entraînement des politiques multi-agents, tels que le choix de l'algorithme MARL, les hyperparamètres d'entraînement (taille de batch, taux d'apprentissage, facteur de discount, etc.), ainsi que les spécifications organisationnelles MOISE+MARL.

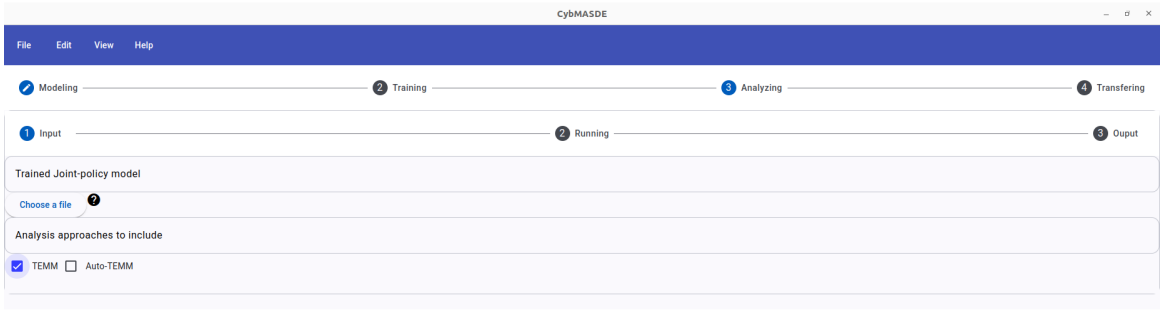


FIGURE 43 : Capture d'écran de l'interface graphique d'édition du fichier de configuration du projet CybMASDE, illustrant ici l'onglet dédié à l'analyse. Dans cet onglet, l'utilisateur peut configurer les paramètres liés à l'analyse des politiques multi-agents, tels que la méthode à utiliser (TEMM ou Auto-TEMM) mais aussi les métriques d'évaluation, les visualisations des trajectoires, et les spécifications organisationnelles MOISE+MARL.

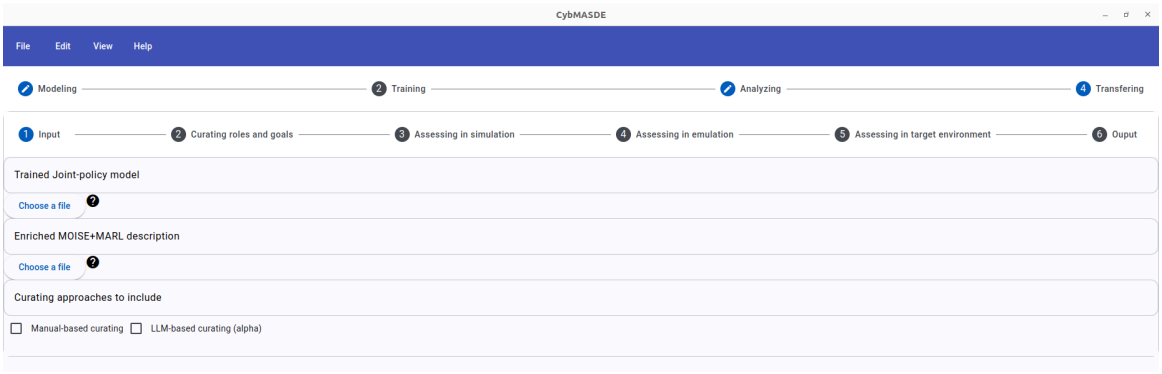


FIGURE 44 : Capture d'écran de l'interface graphique d'édition du fichier de configuration du projet CybMASDE, illustrant ici l'onglet dédié à la phase raffinement et de transfert des politiques. Dans cet onglet, l'utilisateur peut configurer les paramètres liés au raffinement des politiques multi-agents, tels que le nombre maximal d'itérations de raffinement, les critères d'arrêt basés sur la récompense moyenne et la stabilité, ainsi que les options de déploiement des politiques dans l'environnement réel.

B.2 API ENVIRONNEMENTALE DE CYBMASDE

Le Listing 5 présente un extrait du fichier gabarit à utiliser pour implémenter l'API environnementale. Cette API est essentielle pour permettre à CybMASDE de communiquer avec l'environnement cible. En implémentant cette interface, l'utilisateur définit les méthodes nécessaires pour récupérer les observations et historiques, appliquer des actions conjointes, et déployer des politiques conjointes dans l'environnement.

```python
class EnvironmentAPI:
    """Class representing the environment API for interacting with the environment."""

    def __init__(self):
        pass

    def retrieve_joint_observation(self):
        """Retrieve the joint observation from the environment."""
        # Implement the logic to retrieve the joint observation
        pass

    def retrieve_joint_histories(self):
        """Retrieve the joint histories from the environment."""
        # Implement the logic to retrieve the joint histories
        pass

    def apply_joint_action(self, joint_action):
        """Apply the joint action to the environment."""
        # Implement the logic to apply the joint action
        pass

    def deploy_joint_policy(self, joint_policy):
        """Deploy the joint policy to the environment."""
        # Implement the logic to deploy the joint policy
        pass
```

Listing 5 : Extrait du fichier gabarit à utiliser pour implémenter l'API environnementale.

B.3 MANUEL EN LIGNE DE COMMANDE DE CYBMASDE

```
CYBMASDE(1)                Manuel de l'utilisateur
↪  CYBMASDE(1)

NOM
cybmasde - orchestrer la méthode MAMAD pour la conception de SMA

SYNOPSIS
cybmasde [commande] [options]

DESCRIPTION
CybMASDE est une plateforme modulaire permettant de créer, configurer,
valider, exécuter et analyser des projets de SMA fondés sur le cadre
MOISE+MARL. Elle supporte l'exécution en ligne de commande (CLI) ou via son
↪  interface graphique Angular. Le CLI permet une automatisation complète
↪  (mode batch/HPC) ou une utilisation interactive.

COMMANDES PRINCIPALES
init        Créer un nouveau projet et générer l'arborescence associée.
validate    Vérifier la validité du fichier project_configuration.json et
↪  des dépendances.
run         Exécuter un projet complet ou partiel.
model       Lancer uniquement l'activité de modélisation.
train       Lancer uniquement l'activité d'entraînement.
analyze     Lancer uniquement l'activité d'analyse (Auto-TEMM).
refine      Lancer un cycle de raffinement (analyse + entraînement).
deploy      Déployer une politique dans l'environnement réel.
status      Afficher l'état courant du projet (politiques, métriques,
↪  logs).
clean       Nettoyer les fichiers temporaires, traces ou checkpoints
↪  inutiles.
export      Exporter les résultats (politiques, métriques, specs. org.) au
↪  format JSON/CSV.
help        Afficher l'aide générale ou celle d'une commande spécifique.

OPTIONS GÉNÉRALES
-h, --help
Afficher l'aide.
-v, --version
Afficher la version de CybMASDE.
-p, --project <path>
Chemin vers le projet (par défaut: répertoire courant).
-c, --config <file>
Spécifier un fichier de configuration alternatif.

SOUS-COMMANDES ET OPTIONS

init
```

```
cybmasde init -n <nom_projet> [-d <description>] [-o <output_dir>]

-n, --name <nom>
Nom du projet.
-d, --description <texte>
Description textuelle du projet.
-o, --output <dir>
Répertoire de sortie (par défaut: ./<nom_projet>).
--template <type>
Type de template d'environnement (handcrafted|worldmodel|minimal).

validate
cybmasde validate [-q]

-q, --quiet
Ne pas afficher les détails, seulement l'état final (OK/ERREUR).
--strict
Considérer tout avertissement comme une erreur.

run
cybmasde run [--full-auto | --semi-auto | --manual]

--full-auto
Exécuter l'ensemble du pipeline (MTA+T) sans interaction.
--semi-auto
Exécution complète mais pause à chaque étape pour confirmation.
--manual
Mode manuel : l'utilisateur choisit chaque activité à exécuter.
--skip-model
Ne pas relancer la modélisation (utiliser environnement
existant).
--skip-analyze
Ne pas lancer Auto-TEMM même si la récompense est faible.
--max-refine <N>
Nombre maximal d'itérations de raffinement.
--reward-threshold <val>
Seuil de récompense moyenne pour arrêt automatique.
--std-threshold <val>
Seuil d'écart-type de la récompense pour arrêt du raffinement.
--accept-inferred
Accepter automatiquement les spécifications inférées sans validation
↪ humaine.
--interactive-infer
Afficher les specs inférées et demander validation manuelle (défaut).

model
cybmasde model [--auto | --manual] [options]

--auto
```

```
Utiliser les traces + World Model pour générer l'environnement.
--manual
Lancer l'environnement MCAS (handcrafted_environment.py).
--traces <dir>
Répertoire contenant des historiques préexistants.
--vae-dim <val>
Dimension latente des VAE (défaut: 32).
--lstm-hidden <val>
Taille des couches cachées LSTM (64 ou 128).

train
cybmasde train [--algo <alg>] [options]

--algo <nom>
Algorithme MARL (MAPPO|MADDPG|QMIX|IQL|VDN|ROMA).
--batch-size <val>
Taille des batchs (64 ou 128).
--lr <val>
Taux d'apprentissage (1e-4 à 5e-4).
--gamma <val>
Facteur de discount (0.9 à 0.99).
--clip <val>
Valeur de clipping PPO (0.1 à 0.3).
--seed <val>
Graine aléatoire.
--epochs <N>
Nombre d'époques.

analyze
cybmasde analyze [--auto-temm]

--auto-temm
Utiliser Auto-TEMM (clustering + optimisation hyperparamètres).
--metrics <list>
Sélectionner les métriques d'analyse (reward|stability|org_fit).
--representativity <val>
Seuil de représentativité (0.0-1.0).

refine
cybmasde refine [--max <N>] [--accept-inferred]

--max <N>
Nombre maximum d'itérations de raffinement.
--accept-inferred
Accepter automatiquement les specs organisationnelles inférées.
--interactive
Demander confirmation utilisateur à chaque cycle (défaut).

deploy
```

```
cybmasde deploy [--direct | --remote]

--direct
Déploiement de la politique sur les agents (mode embarqué).
--remote
Politique exécutée par CybMASDE, agents ne reçoivent que les
actions.
--checkpoint <file>
Spécifier un checkpoint particulier de politique.
--api <url>
Spécifier l'URL de l'API environnementale cible.

status
cybmasde status

Affiche l'état du projet : politique active, métriques récentes,
nombre de cycles MTA, état du transfert.

clean
cybmasde clean [--traces | --checkpoints | --all]

Nettoyer les fichiers temporaires et résultats intermédiaires.

export
cybmasde export [--format json|csv|yaml] [--output <dir>]

Exporter les résultats, politiques et spécifications organisationnelles.

EXEMPLES
cybmasde init -n infra_test --template worldmodel
cybmasde validate
cybmasde run --full-auto --reward-threshold 3.5 --max-refine 5
cybmasde refine --interactive
cybmasde deploy --remote --api http://localhost:8080/api

VOIR AUSSI
Documentation complète : https://github.com/julien6/CybMASDE
Référence théorique MAMAD et MOISE+MARL dans le manuscrit associé.
```

BIBLIOGRAPHIE

[1] *MITRE ATTA&CK*. https://attack.mitre.org/. Accessed : 2023-04-11.

[2] A. Abazari, M. Ghafouri et D. Jafarigiv. "Data-Driven Framework for Mitigating EV-Based Load-Altering Attacks on LFC Model of Microgrid". In : *IEEE Transactions on Smart Grid* (2025).

[3] Agence nationale de la sécurité des systèmes d'information (ANSSI). *Panorama de la cybermenace 2024*. Rapport menaces et incidents CERTFR-2025-CTI-003. Paris : CERT-FR, Agence nationale de la sécurité des systèmes d'information, 11 mars 2025. URL : https://www.cert.ssi.gouv.fr/uploads/CERTFR-2025-CTI-003.pdf (visité le 25/08/2025).

[4] *Consensus Sequence [MeSH]*. https://www.ncbi.nlm.nih.gov/mesh?term=%22Consensus+Sequence%22%5BMeSH+Terms%5D. U.S. National Library of Medicine, Medical Subject Headings. 2025.

[5] Falong Fan et Xi Li. "PeerGuard : Defending Multi-Agent Systems Against Backdoor Attacks Through Mutual Reasoning". In : (mai 2025). DOI : 10.48550/arXiv.2505.11642.

[6] C.R. Landolt, C. Würsch, R. Meier et A. Mermoud. "Multi-Agent Reinforcement Learning in Cybersecurity : From Fundamentals to Applications". In : *arXiv preprint arXiv :2505.19837* (2025).

[7] Peilang Li, Umer Siddique et Yongcan Cao. "From Explainability to Interpretability : Interpretable Policies in Reinforcement Learning Via Model Explanation". In : *CoRR* abs/2501.09858 (2025). DOI : 10.48550/ARXIV.2501.09858. arXiv : 2501.09858.

[8] Zichuan Liu, Yuanyang Zhu, Zhi Wang, Yang Gao et Chunlin Chen. "MIXRTs : Toward Interpretable Multi-Agent Reinforcement Learning via Mixing Recurrent Soft Decision Trees". In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 47.5 (2025), p. 4090-4107. DOI : 10.1109/TPAMI.2025.3540467.

[9] Vineeth Sai Narajala et Om Narayan. *Securing Agentic AI : A Comprehensive Threat Model and Mitigation Framework for Generative AI Agents*. Avr. 2025. DOI : 10.48550/arXiv.2504.19956. arXiv : 2504.19956 [cs.CR].

[10] Huynh Phuong Thanh Nguyen, Kento Hasegawa, Kazuhide Fukushima et Razvan Beuran. "PenGym : Realistic training environment for reinforcement learning pentesting agents". In : *Computers & Security* 148 (2025), p. 104140. ISSN : 0167-4048. DOI : 10.1016/j.cose.2024.104140. URL : https://www.sciencedirect.com/science/article/pii/S0167404824004450.

[11] Onyinye Obioha-Val, Titilayo Modupe Kolade, Michael Gbadebo, Oluwatosin Selesi-Aina, Omobolaji Olateju et Oluwaseun Olaniyi. "Strengthening Cybersecurity Measures for the Defense of Critical Infrastructure in the United States". In : *SSRN Electronic Journal* (jan. 2025). DOI : 10.2139/ssrn.5021072.

[12] Yoann Poupart, Aurélie Beynier et Nicolas Maudet. "Perspectives for Direct Interpretability in Multi-Agent Deep Reinforcement Learning". In : *CoRR* abs/2502.00726 (2025). DOI : 10.48550/ARXIV.2502.00726. arXiv : 2502.00726.

[13] Julien SOULÉ, Jean-Paul JAMONT, Michel OCCELLO, Louis-Marie TRAONOUEZ et Paul THÉRON. "An Organizationally-Oriented Approach to Enhancing Explainability and Control in Multi-Agent Reinforcement Learning". In : *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*. AAMAS '25. Detroit, MI, USA : International Foundation for Autonomous Agents et Multiagent Systems, 2025, p. 1968-1976.

[14] Julien SOULÉ, Jean-Paul JAMONT, Michel OCCELLO, Louis-Marie TRAONOUEZ et Paul THÉRON. "Streamlining Resilient Kubernetes Autoscaling with Multi-Agent Systems via an Automated Online Design Framework". In : *2025 IEEE 18th International Conference on Cloud Computing (CLOUD)*. 2025, p. 43-53. DOI : 10.1109/CLOUD67622.2025.00015.

[15] S. SUN. "Redefining Adversarial Dynamics : Co-Evolution of Attack and Defense Strategies in AI-Enabled Power CPS". In : *Preprints.org* (2025). URL : https://www.preprints.org/manuscript/202505.1425.

[16] W.J. TAN, W. CAI et A. ZHANG. "Privacy Meets Performance : Enhancing Distributed Simulation-based Federated Multi-agent Learning with Privacy-preserving Surrogate Model". In : *ACM Transactions on Modeling and Computer Simulation (TOMACS)* (2025).

[17] S. VYAS, V. MAVROUDIS et P. BURNAP. "Towards the deployment of realistic autonomous cyber network defence : A systematic review". In : *ACM Computing Surveys* (2025). DOI : 10.1145/3729213.

[18] Xinglin ZHOU, Yifu YUAN, Shaofu YANG et Jianye HAO. "MENTOR : Guiding Hierarchical Reinforcement Learning With Human Feedback and Dynamic Distance Constraint". In : *IEEE Transactions on Emerging Topics in Computational Intelligence* 9.2 (2025), p. 1292-1306. DOI : 10.1109/TETCI.2025.3529902.

[19] Stefano V. ALBRECHT, Filippos CHRISTIANOS et Lukas SCHÄFER. *Multi-Agent Reinforcement Learning : Foundations and Modern Approaches*. MIT Press, 2024. URL : https://www.marl-book.com.

[20] D. CAO, J. HU et Y. LIU. "Decentralized Graphical-Representation-Enabled Multi-Agent DRL for Robust Control of CPS". In : *IEEE Transactions on Reliability* (2024).

[21] R. FERNANDES, N. LOPES et J. GONÇALVES. "Autonomous Pentesting Using Reinforcement Learning : A Systematic Literature Review". In : *SSRN Electronic Journal* (2024). DOI : 10.2139/ssrn.5208526.

[22] Pedro Enrique ITURRIA-RIVERA, Raimundas GAIGALAS, Medhat H. M. ELSAYED, Majid BAVAND, Yigit OZCAN et Melike EROL-KANTARCI. "Explainable Multi-Agent Reinforcement Learning for Extended Reality Codec Adaptation". In : *CoRR* abs/2411.14264 (2024). DOI : 10.48550/ARXIV.2411.14264. arXiv : 2411.14264.

[23] Jaromír JANISCH, Tomáš PEVNÝ et Viliam LISÝ. "NASimEmu : Network Attack Simulator & Emulator for Training Agents Generalizing to Novel Scenarios". In : *Computer Security. ESORICS 2023 International Workshops*. Sous la dir. de Sokratis KATSIKAS et al. Cham : Springer Nature Switzerland, 2024, p. 589-608. ISBN : 978-3-031-54129-2.

[24] Ariel KWIATKOWSKI et al. *Gymnasium : A Standard Interface for Reinforcement Learning Environments*. 2024. DOI : 10.48550/arxiv.2407.17032. arXiv : 2407.17032 [cs.LG].

[25] S. et al. MILANI. "Interpretable Multi-Agent Reinforcement Learning with Decision Tree Policies". In : *Agency in Artificial Societies*. Taylor & Francis, 2024. URL : http://repository.bitscollege.edu.et:8080/bitstream/handle/123456789/742/EXPLAI~1.PDF.

[26] Ian MILES et al. *Reinforcement Learning for Autonomous Resilient Cyber Defence*. Rapp. tech. Accessed : 2025-07-13. Las Vegas, NV, USA, 2024. URL : https://www.blackhat.com/us-24/briefings/schedule/#reinforcement-learning-for-autonomous-resilient-cyber-defense-39308.

[27] S. MOHAMMADI, V.H. BUI et W. SU. "Surrogate Modeling for Solving OPF : A Review". In : *Sustainability* 16.22 (2024), p. 9851. URL : https://www.mdpi.com/2071-1050/16/22/9851.

[28] Alexander RUTHERFORD et al. "JaxMARL : Multi-Agent RL Environments and Algorithms in JAX". In : *Advances in Neural Information Processing Systems*. Sous la dir. d'A. GLOBERSON, L. MACKEY, D. BELGRAVE, A. FAN, U. PAQUET, J. TOMCZAK et C. ZHANG. T. 37. Curran Associates, Inc., 2024, p. 50925-50951. URL : https://proceedings.neurips.cc/paper_files/paper/2024/file/5aee125f052c90e326dcf6f380df94f6-Paper-Datasets_and_Benchmarks_Track.pdf.

[29] Julien SOULE, Jean-Paul JAMONT, Michel OCCELLO, Louis-Marie TRAONOUEZ et Paul THÉRON. "A MARL-Based Approach for Easing SMA Organization Engineering". In : *Proc. of the 20th Int. Conf. Artificial Intelligence Applications and Innovations*. 2024. DOI : 10.1007/978-3-031-63223-5_24.

[30] Julien SOULÉ. *Warehouse Management*. https://github.com/julien6/OMARLE. 2024. URL : https://github.com/julien6/OMARLE.

[31] Julien SOULÉ, Jean-Paul JAMONT, Michel OCCELLO, Louis-Marie TRAONOUEZ et Paul THÉRON. "A MARL-Based Approach for Easing MAS Organization Engineering". In : *Artificial Intelligence Applications and Innovations*. Sous la dir. d'Ilias MAGLOGIANNIS, Lazaros ILIADIS, John MACINTYRE, Markos AVLONITIS et Antonios PAPALEONIDAS. Cham : Springer Nature Switzerland, 2024, p. 321-334. ISBN : 978-3-031-63223-5.

[32] C. SUBRAMANIAN, M. LIU, N. KHAN et J. LENCHNER. "A neuro-symbolic approach to multi-agent rl for interpretability and probabilistic decision making". In : *arXiv preprint arXiv :2402.13440* (2024). DOI : 10.48550/arxiv.2402.13440. arXiv : 2402.13440.

[33] Office of the UNDER SECRETARY OF DEFENSE FOR RESEARCH et ENGINEERING. *Cyber Resiliency Engineering Framework*. DTIC Technical Report AD 1108457. Consulté via DTIC le 13 juillet 2025. Fort Belvoir, VA, USA : Defense Technical Information Center, 2024.

[34] Jiacen XU, Jack W. STOKES, Geoff MCDONALD, Xuesong BAI, David MARSHALL, Siyue WANG, Adith SWAMINATHAN et Zhou LI. "AutoAttacker : A Large Language Model Guided System to Implement Automatic Cyber-attacks". In : *CoRR* abs/2403.01038 (2024). DOI : 10.48550/ARXIV.2403.01038. arXiv : 2403.01038.

[35] Yizhou YANG, Longde CHEN, Sha LIU, Lanning WANG, Haohuan FU, Xin LIU et Zuoning CHEN. "Behaviour-diverse automatic penetration testing : a coverage-based deep reinforcement learning approach". In : *Front. Comput. Sci.* 19.3 (nov. 2024). ISSN : 2095-2228. DOI : 10.1007/s11704-024-3380-1.

[36] Z. ZHANG. "Advancing sample efficiency and explainability in multi-agent reinforcement learning". In : *Proceedings of AAMAS 2024*. 2024. URL : https://www.ifaamas.org/Proceedings/aamas2024/pdfs/p2791.pdf.

[37] ZHOU ZHIQIANG ET AL. "AHPA : Adaptive Horizontal Pod Autoscaling Systems on Alibaba Cloud Container Service for Kubernetes". In : *Proc. of the AAAI Conf. on Artificial Intelligence* 37.13 (juill. 2024), p. 15621-15629. DOI : 10.1609/aaai.v37i13.26852.

[38] J.S. GROVER. "System identification and control of multiagent systems through interactions". Thèse de doct. ProQuest Dissertations Publishing, 2023.

[39] Kim HAMMAR et Rolf STADLER. "Digital Twins for Security Automation". In : *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*. 2023, p. 1-6. DOI : 10.1109/NOMS56928.2023.10154288.

[40] A. HENSLEE, H. DOZIER et al. "Data-Driven Reinforcement Learning for Mission Engineering and Combat Simulation". In : *Lecture Notes in Computer Science (LNCS)*. Springer, 2023.

[41] Siyi HU, Yifan ZHONG, Minquan GAO, Weixun WANG, Hao DONG, Xiaodan LIANG, Zhihui LI, Xiaojun CHANG et Yaodong YANG. "MARLlib : A Scalable and Efficient Multi-agent Reinforcement Learning Library". In : *J. Mach. Learn. Res.* 24 (2023), 315 :1-315 :23. URL : http://jmlr.org/papers/v24/23-0378.html.

[42] Alexander KOTT et al. *Autonomous Intelligent Cyber-defense Agent (AICA) Reference Architecture. Release 2.0*. Cham, Switzerland, 2023.

[43] G. PALMER, C. PARRY, D.J.B. HARROLD et C. WILLIS. "Deep Reinforcement Learning for Autonomous Cyber Defence : A Survey". In : *arXiv preprint arXiv :2310.07745* (2023).

[44] Haoran QIU, Weichao MAO, Chen WANG, Hubertus FRANKE, Alaa YOUSSEF, Zbigniew T. KALBARCZYK, Tamer BAŞAR et Ravishankar K. IYER. "AWARE : Automate Workload Autoscaling with Reinforcement Learning in Production Cloud Systems". In : *2023 USENIX Annual Technical Conf. (USENIX ATC 23)*. 2023, p. 387-402. ISBN : 978-1-939133-35-9.

[45] José SANTOS, Tim WAUTERS, Bruno VOLCKAERT et Filip De TURCK. "gym-hpa : Efficient Auto-Scaling via Reinforcement Learning for Complex Microservice-based Applications in Kubernetes". In : *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*. 2023, p. 1-9. DOI : 10.1109/NOMS56928.2023.10154298.

[46] Jonathan J. SCHWARTZ et Hanna KURNIAWATI. *NASim : Network Attack Simulator (Version 0.12.0)*. https://networkattacksimulator.readthedocs.io/en/latest/. Open-source Python software for simulating network environments and attack scenarios for autonomous penetration testing agents. 2023.

[47] Julien SOULÉ, Jean-Paul JAMONT, Michel OCCELLO, Paul THÉRON et Louis-Marie TRAONOUEZ. "De l'organisation d'un système multi-agent de cyberdéfense". In : *Rendez-Vous de la Recherche et de l'Enseignement de la Sécurité des Systèmes d'Information (RESSI 2023)*. Mai 2023. URL : https://ressi2023.sciencesconf.org/450961/document.

[48] Julien SOULÉ, Jean-Paul JAMONT, Michel OCCELLO, Paul THÉRON et Louis-Marie TRAONOUEZ. "Towards a Multi-Agent Simulation of Cyber-attackers and Cyber-defenders Battles". In : *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2023, p. 3594-3599. DOI : 10.1109/SMC53992.2023.10394564.

[49] O. THAPLIYAL. "Data-Driven Safety & Security of Cyber-physical Systems". Thèse de doct. Purdue University, 2023.

[50] Sanyam VYAS, John HANNAY, Andrew BOLTON et Pete BURNAP. "Automated Cyber Defence : A Review". In : *CoRR* abs/2303.04926 (2023). DOI : 10.48550/ARXIV.2303.04926. arXiv : 2303.04926.

[51] W. YANG et J. DONG. "Data-driven learning for resilient synchronization and parameter estimation of heterogeneous nonlinear multiagent systems". In : *IEEE Transactions on Automation Science and Engineering* (2023).

[52] Renos ZABOUNIDIS, Joseph CAMPBELL, Simon STEPPUTTIS, Dana HUGHES et Katia P. SYCARA. "Concept Learning for Interpretable Multi-Agent Reinforcement Learning". In : *Proceedings of The 6th Conference on Robot Learning*. Sous la dir. de Karen LIU, Dana KULIC et Jeff ICHNOWSKI. T. 205. Proceedings of Machine Learning Research. PMLR, déc. 2023, p. 1828-1837. URL : https://proceedings.mlr.press/v205/zabounidis23a.html.

[53] Alex ANDREW, Sam SPILLARD, Joshua COLLYER et Neil DHIR. "Developing Optimal Causal Cyber-Defence Agents via Cyber Security Simulation". In : *International Confernece on Machine Learning (ICML). Workshop on Machine Learning for Cybersecurity (ML4Cyber)*. Juill. 2022.

[54] CHAO YU ET. AL. *The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games*. 2022. DOI : 10.48550/arxiv.2103.01955. arXiv : 2103.01955 [cs.LG].

[55] TTCP CAGE Working GROUP. *TTCP CAGE Challenge 3*. https://github.com/cage-challenge/cage-challenge-3. 2022.

[56] N. GRUPEN, N. JAQUES et B. KIM. "Concept-based understanding of emergent multi-agent behavior". In : *NeurIPS Workshop on Human-Centric Machine Learning*. 2022. URL : https://openreview.net/pdf?id=zt5JpGQ8WhH.

[57] Kim HAMMAR et Rolf STADLER. "A System for Interactive Examination of Learned Security Policies". In : *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. 2022, p. 1-3. DOI : 10.1109/NOMS54207.2022.9789707.

[58] Tanmoy HAZRA et Kushal ANJARIA. "Applications of game theory in deep learning : a survey". In : *Multimedia Tools and Applications* 81.6 (2022), p. 8963-8994. ISSN : 1573-7721. DOI : 10.1007/s11042-022-12153-2.

[59] Jiechuan JIANG et Zongqing LU. "I2Q : A Fully Decentralized Q-Learning Algorithm". In : *Advances in Neural Information Processing Systems*. Sous la dir. de S. KOYEJO, S. MOHAMED, A. AGARWAL, D. BELGRAVE, K. CHO et A. OH. T. 35. Curran Associates, Inc., 2022, p. 20469-20481. URL : https://proceedings.neurips.cc/paper_files/paper/2022/file/8078e8c3055303a884ffae2d3ea00338-Paper-Conference.pdf.

[60] S. MILANI, Z. ZHANG et N. et al. TOPIN. "MAVIPER : Learning decision tree policies for interpretable multi-agent reinforcement learning". In : *ECML PKDD 2022*. 2022. DOI : 10.48550/arxiv.2205.12449. arXiv : 2205.12449.

[61] Sobhan MIRYOOSEFI et Chi JIN. "A Simple Reward-free Approach to Constrained Reinforcement Learning". In : *Proceedings of the 39th International Conference on Machine Learning*. Sous la dir. de Kamalika CHAUDHURI, Stefanie JEGELKA, Le SONG, Csaba SZEPESVARI, Gang NIU et Sivan SABATO. T. 162. Proceedings of Machine Learning Research. PMLR, juill. 2022, p. 15666-15698. URL : https://proceedings.mlr.press/v162/miryoosefi22a.html.

[62] Mikel D. Petty, Tymaine S. Whitaker, E. Michael Bearss, John A. Bland, Walter Alan Cantrell, Christopher Daniel Colvett et Katia P. Maxwell. "Modeling cyberattacks with extended Petri nets". In : *ACM SE '22 : 2022 ACM Southeast Conference, Virtual Event, April 18 - 20, 2022*. Sous la dir. de Christopher Ogden et Eric Gamess. ACM, 2022, p. 67-73. DOI : 10.1145/3476883.3520209.

[63] A. Zutshi, T. Dietterich, A. Fern et S. Jagannathan. *Leveraging Symbolic Representations for Safe and Assured Learning*. Rapp. tech. Defense Technical Information Center (DTIC), 2022. URL : https://apps.dtic.mil/sti/pdfs/AD1177312.pdf.

[64] Luciano Baresi, Davide Yi Xian Hu, Giovanni Quattrocchi et Luca Terracciano. "KOSMOS : Vertical and Horizontal Resource Autoscaling for Kubernetes". In : *Service-Oriented Computing*. Sous la dir. d'Hakim Hacid, Odej Kao, Massimo Mecella, Naouel Moha et Hye-young Paik. Cham : Springer Int. Publishing, 2021, p. 821-829. ISBN : 978-3-030-91431-8.

[65] CROND. *AutoPentest-DRL*. https://github.com/crond-jaist/AutoPentest-DRL. 2021.

[66] Mehmet Özgün Demir, Ozan Alp Topal, Ali Emre Pusane, Guido Dartmann, Gerd Ascheid et Günes Karabulut-Kurt. "An Adaptive Multi-Agent Physical Layer Security Framework for Cognitive Cyber-Physical Systems". In : *CoRR* abs/2101.02446 (2021). DOI : 10.48550/arxiv.2101.02446. arXiv : 2101.02446.

[67] Zhijun Ding et Qichen Huang. "COPA : A Combined Autoscaling Method for Kubernetes". In : *2021 IEEE Int. Conf. on Web Services (ICWS)*. 2021, p. 416-425. DOI : 10.1109/ICWS53863.2021.00061.

[68] Atticus Geiger, Zhengxuan Wu, Hanson Lu, Elisa Kreiss, Thomas Icard et Noah Goodman. "Causal abstraction for faithful model interpretation". In : *Advances in Neural Information Processing Systems (NeurIPS)*. T. 34. 2021.

[69] Abeer Abdel Khaleq et Ilkyeun Ra. "Development of QoS-aware agents with reinforcement learning for autoscaling of microservices on the cloud". In : *Int. Conf. on Autonomic Computing and Self-Organizing Systems Companion (ACSOS)*. 2021, p. 13-19. DOI : 10.1109/ACSOS-C52956.2021.00025.

[70] Viktor Makoviychuk et al. "Isaac Gym : High Performance GPU Based Physics Simulation For Robot Learning". In : *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. Sous la dir. de J. Vanschoren et S. Yeung. T. 1. 2021. URL : https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/28dd2c7955ce926456240b2ff0100bde-Paper-round2.pdf.

[71] Georgios Papoudakis, Filippos Christianos, Sharada P Rahman et Stefano V Albrecht. "Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks". In : *NeurIPS* (2021).

[72] Georgios Papoudakis, Filippos Christianos, Matthias Schmidt et Stefano V. Albrecht. "Agent Modelling under Partial Observability for Deep Reinforcement Learning". In : *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*. 2021.

[73] Helge Spieker. "Constraint-Guided Reinforcement Learning : Augmenting the Agent-Environment-Interaction". In : *International Joint Conference on Neural Networks (IJCNN)* (2021). ISSN : 2161-4407. DOI : 10.1109/IJCNN52387.2021.9533996.

[74] Maxwell STANDEN, Martin LUCAS, David BOWMAN, Toby J. RICHER, Junae KIM et Damian MARRIOTT. "CybORG : A Gym for the Development of Autonomous Cyber Agents". In : *CoRR* abs/2108.09118 (2021). DOI : 10.48550/arxiv.2108.09118. arXiv : 2108.09118.

[75] Microsoft Defender Research TEAM. *CyberBattleSim*. https://github.com/microsoft/cyberbattlesim. Created by Christian Seifert, Michael Betser, William Blum, James Bono, Kate Farris, Emily Goren, Justin Grana, Kristian Holsheimer, Brandon Marken, Joshua Neil, Nicole Nichols, Jugal Parikh, Haoran Wei. 2021.

[76] Paul THERON, Nate EVANS, Martin DRASAR et Alesandro GUARINO. *Autonomous Intelligent Cyber Defence Agent Prototype 2021 - Project Report*. Déc. 2021.

[77] Kaiqing ZHANG, Zhuoran YANG et Tamer BAŞAR. "Multi-Agent Reinforcement Learning : A Selective Overview of Theories and Algorithms". In : *Handbook of Reinforcement Learning and Control*. Sous la dir. de Kyriakos G. VAMVOUDAKIS, Yan WAN, Frank L. LEWIS et Derya CANSEVER. Cham : Springer International Publishing, 2021, p. 321-384. ISBN : 978-3-030-60990-0. DOI : 10.1007/978-3-030-60990-0_12.

[78] John A. BLAND, C. Daniel COLVETT, Walter Alan CANTRELL, Katia P. MAYFIELD, Mikel D. PETTY et Tymaine S. WHITAKER. "Machine Learning Cyberattack Strategies with Petri Nets with Players, Strategies, and Costs". In : *National Cyber Summit (NCS) Research Track*. Sous la dir. de Kim-Kwang Raymond CHOO, Thomas H. MORRIS et Gilbert L. PETERSON. Cham : Springer International Publishing, 2020, p. 232-247. ISBN : 978-3-030-31239-8.

[79] Micah CARROLL, Rohin SHAH, Mark HO, Tom GRIFFITHS, Pieter ABBEEL et Anca DRAGAN. "Overcooked-AI : A Benchmark for Multi-Agent Learning under Partial Observability". In : *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2020), p. 2374-2380.

[80] Ankur CHOWDHARY, Dijiang HUANG, Jayasurya Sevalur MAHENDRAN, Daniel ROMO, Yuli DENG et Abdulhakim SABUR. "Autonomous Security Analysis and Penetration Testing". In : *2020 16th International Conference on Mobility, Sensing and Networking (MSN)*. 2020, p. 508-515. DOI : 10.1109/MSN50589.2020.00086.

[81] Raphael COHEN, Nathan CHANDLER, Shira EFRON, Bryan FREDERICK, Eugeniu HAN, Kurt KLEIN, Forrest MORGAN, Ashley RHOADES, Howard SHATZ et Yuliya SHOKH. *The Future of Warfare in 2030 : Project Overview and Conclusions*. Jan. 2020. ISBN : 9781977402950. DOI : 10.7249/RR2849.1.

[82] Martin DRAŠAR, Stephen MOSKAL, Shanchieh YANG et Pavol ZAT'KO. "Session-level Adversary Intent-Driven Cyberattack Simulator". In : *2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. 2020, p. 1-9. DOI : 10.1109/DS-RT50469.2020.9213690.

[83] Gabriel KALWEIT, Maria HUEGLE, Moritz WERLING et Joschka BOEDECKER. "Deep Constrained Q-Learning". In : *arXiv preprint arXiv :2003.09398* (2020).

[84] Alexander KOTT et Paul THÉRON. "Doers, Not Watchers : Intelligent Autonomous Agents Are a Path to Cyber Resilience". In : *IEEE Secur. Priv.* 18.3 (2020), p. 62-66. DOI : 10.1109/MSEC.2020.2983714.

[85] Thomas M. MOERLAND, Joost BROEKENS et Catholijn M. JONKER. "Model-based Reinforcement Learning : A Survey". In : *CoRR* abs/2006.16712 (2020). DOI : 10.48550/arxiv.2006.16712. arXiv : 2006.16712.

[86] M. STERNKE, K. W. TRIPP et D. BARRICK. "The use of consensus sequence information to engineer protein stability". In : *Methods in Enzymology* 643 (2020), p. 63-88. DOI : 10.1016/bs.mie.2020.05.029.

[87] J. K TERRY et al. "PettingZoo : Gym for Multi-Agent Reinforcement Learning". In : *arXiv preprint arXiv :2009.14471* (2020).

[88] P. THÉRON et al. *A first prototype of the Multi-Agent System Centric AICA Reference Architecture (MASCARA)*. Presentation, 1st NATO – AICA IWG Virtual Technical Workshop on Autonomous Cyber Defence. PowerPoint presentation. Nov. 2020.

[89] TONGHAN WANG ET. AL. *ROMA : Multi-Agent Reinforcement Learning with Emergent Roles*. 2020. DOI : 10.48550/arxiv.2003.08039. arXiv : 2003.08039 [cs.MA].

[90] Shingo YAMAGUCHI. "White-Hat Worm to Fight Malware and Its Evaluation by Agent-Oriented Petri Nets". In : *Sensors* 20.2 (2020). ISSN : 1424-8220. DOI : 10.3390/s20020556. URL : https://www.mdpi.com/1424-8220/20/2/556.

[91] Takuya AKIBA, Shotaro SANO, Toshihiko YANASE, Takeru OHTA et Masanori KOYAMA. *Optuna : A Next-generation Hyperparameter Optimization Framework*. 2019. DOI : 10.48550/arxiv.1907.10902. arXiv : 1907.10902 [cs.LG].

[92] Micah CARROLL, Rohin SHAH, Mark K HO, Tom GRIFFITHS, Sanjit SESHIA, Pieter ABBEEL et Anca DRAGAN. "On the Utility of Learning about Humans for Human-AI Coordination". In : *Advances in Neural Information Processing Systems*. Sous la dir. de H. WALLACH, H. LAROCHELLE, A. BEYGELZIMER, F. D'ALCHÉ-BUC, E. FOX et R. GARNETT. T. 32. Curran Associates, Inc., 2019.

[93] Roy FROSTIG, Matthew James JOHNSON et Chris LEARY. "Compiling Machine Learning Programs via High-Level Tracing". In : *SysML Conference 2018*. Stanford, United States, mars 2019. URL : https://hal.science/hal-05188750.

[94] David GUNNING, Mark STEFIK, Jaesik CHOI, Timothy MILLER, Simone STUMPF et Guang-Zhong YANG. "XAI—Explainable artificial intelligence". In : *Science Robotics* 4.37 (2019), eaay7120. DOI : 10.1126/scirobotics.aay7120.

[95] Danijar HAFNER, Timothy LILLICRAP, Jimmy BA et Mohammad NOROUZI. "Learning latent dynamics for planning from pixels". In : *Proceedings of the 36th International Conference on Machine Learning (ICML)*. 2019, p. 2555-2565.

[96] Danijar HAFNER, Timothy P. LILLICRAP, Jimmy BA et Mohammad NOROUZI. "Dream to Control : Learning Behaviors by Latent Imagination". In : *CoRR* abs/1912.01603 (2019). DOI : 10.48550/arxiv.1912.01603. arXiv : 1912.01603.

[97] Eric M HOLLOWAY. "Self organized multi agent swarms (SOMAS) for network security control". In : *Theses and Dissertations* (2019).

[98] Adam PASZKE et al. "PyTorch : An Imperative Style, High-Performance Deep Learning Library". In : *Advances in Neural Information Processing Systems*. Sous la dir. de H. WALLACH, H. LAROCHELLE, A. BEYGELZIMER, F. D'ALCHÉ-BUC, E. FOX et R. GARNETT. T. 32. Curran Associates, Inc., 2019. URL : https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

[99] Fabiana ROSSI, Matteo NARDELLI et Valeria CARDELLINI. "Horizontal and Vertical Scaling of Container-Based Applications Using Reinforcement Learning". In : *IEEE 12th Int. Conf. on Cloud Computing (CLOUD)*. 2019, p. 329-338. DOI : 10.1109/CLOUD.2019.00061.

[100] Jack SERRINO, Max KLEIMAN-WEINER, David C. PARKES et Josh TENENBAUM. "Finding Friend and Foe in Multi-Agent Games". In : *Advances in Neural Information Processing Systems 32 : Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Sous la dir. d'Hanna M. WALLACH, Hugo LAROCHELLE, Alina BEYGELZIMER, Florence D'ALCHÉ-BUC, Emily B. FOX et Roman GARNETT. 2019, p. 1249-1259. URL : https://proceedings.neurips.cc/paper/2019/hash/912d2b1c7b2826caf99687388d2e8f7c-Abstract.html.

[101] Stephen Mugisha AKANDWANAHO et Irene GOVENDER. "A Generic Self-Evolving Multi-Agent Defense Approach Against Cyber Attacks". In : *Handbook of Research on Information and Cyber Security in the Fourth Industrial Revolution*. IGI Global, 2018, p. 165-181.

[102] Afraa ATTIAH, Mainak CHATTERJEE et Cliff C. ZOU. "A Game Theoretic Approach to Model Cyber Attack and Defense Strategies". In : *2018 IEEE International Conference on Communications (ICC)*. 2018, p. 1-7. DOI : 10.1109/ICC.2018.8422719.

[103] Matthew P. BARRETT. *Framework for Improving Critical Infrastructure Cybersecurity : Version 1.1*. NIST Cybersecurity Framework CSWP 04162018. Gaithersburg, MD, USA : National Institute of Standards et Technology (NIST), avr. 2018. DOI : 10.6028/NIST.CSWP.04162018. URL : https://www.nist.gov/cyberframework.

[104] Miles BRUNDAGE et al. "The Malicious Use of Artificial Intelligence : Forecasting, Prevention, and Mitigation". In : (fév. 2018). DOI : 10.48550/arXiv.1802.07228.

[105] FALCO, GREGORY ET AL. "A Master Attack Methodology for an AI-Based Automated Attack Planner for Smart Cities". In : *IEEE Access* 6 (2018), p. 48360-48373. DOI : 10.1109/ACCESS.2018.2867556.

[106] Jakob FOERSTER et al. "Counterfactual multi-agent policy gradients". In : *International Conference on Machine Learning (ICML)* (2018).

[107] Jakob FOERSTER, Yannis ASSAEL, Nando de FREITAS et Shimon WHITESON. "Learning to Communicate with Deep Multi-Agent Reinforcement Learning". In : *Advances in Neural Information Processing Systems* 31 (2018), p. 2137-2145.

[108] David HA et Jürgen SCHMIDHUBER. "Recurrent World Models Facilitate Policy Evolution". In : *Advances in Neural Information Processing Systems*. Sous la dir. de S. BENGIO, H. WALLACH, H. LAROCHELLE, K. GRAUMAN, N. CESA-BIANCHI et R. GARNETT. T. 31. Curran Associates, Inc., 2018. URL : https://proceedings.neurips.cc/paper_files/paper/2018/file/2de5d16682c3c35007e4e92982f1a2ba-Paper.pdf.

[109] David HA et Jürgen SCHMIDHUBER. "World Models". In : *CoRR* abs/1803.10122 (2018). DOI : 10.48550/arxiv.1803.10122. arXiv : 1803.10122.

[110] Been KIM, Martin WATTENBERG, Justin GILMER, Carrie CAI, James WEXLER, Fernanda VIEGAS et Rory SAYRES. "Interpretability beyond feature attribution : Quantitative Testing with Concept Activation Vectors (TCAV)". In : *Proceedings of the 35th International Conference on Machine Learning (ICML)*. 2018.

[111] Stefan NICULAE. "Reinforcement Learning vs Genetic Algorithms in Game-Theoretic Cyber-Security". In : (oct. 2018). DOI : 10.31237/osf.io/nxzep.

[112] Martina Panfili, Alessandro Giuseppi, Andrea Fiaschetti, Homoud B. Al-Jibreen, Antonio Pietrabissa et Francesco Delli Priscoli. "A Game-Theoretical Approach to Cyber-Security of Critical Infrastructures Based on Multi-Agent Reinforcement Learning". In : *26th Mediterranean Conference on Control and Automation, MED 2018, Zadar, Croatia, June 19-22, 2018*. IEEE, 2018, p. 460-465. DOI : 10.1109/MED.2018.8442695.

[113] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba et Pieter Abbeel. "Sim-to-real transfer of robotic control with dynamics randomization". In : *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, p. 3803-3810.

[114] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster et Shimon Whiteson. "QMIX : Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning". In : *Proceedings of the 35th International Conference on Machine Learning*. Sous la dir. de Jennifer Dy et Andreas Krause. T. 80. Proceedings of Machine Learning Research. PMLR, juill. 2018, p. 4295-4304. URL : https://proceedings.mlr.press/v80/rashid18a.html.

[115] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster et Shimon Whiteson. "QMIX : Monotonic value function factorisation for deep multi-agent reinforcement learning". In : *Proceedings of the 35th International Conference on Machine Learning* (2018), p. 4295-4304.

[116] Peter Sunehag et al. "Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward". In : *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '18. Stockholm, Sweden : International Foundation for Autonomous Agents et Multiagent Systems, 2018, p. 2085-2087.

[117] Garrett Warnell, Nicholas Waytowich, Vernon Lawhern et Peter Stone. "Deep TAMER : Interactive agent shaping in high-dimensional state spaces". In : *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[118] Joshua Achiam, David Held, Aviv Tian et Pieter Abbeel. "Constrained policy optimization". In : *Proceedings of the 34th International Conference on Machine Learning*. 2017, p. 22-31.

[119] Finale Doshi-Velez et Been Kim. *Towards A Rigorous Science of Interpretable Machine Learning*. 2017. DOI : 10.48550/arxiv.1702.08608. arXiv : 1702.08608 [stat.ML].

[120] E. de la Hoz et al. "A distributed, multi-agent approach to reactive network resilience". In : *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. 2017, p. 1044-1053.

[121] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel et Igor Mordatch. "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments". In : *Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Sous la dir. d'Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan et Roman Garnett. 2017, p. 6379-6390. URL : https://proceedings.neurips.cc/paper/2017/hash/68a9750337a418a86fe06c1991a1d64c-Abstract.html.

[122] Scott M LUNDBERG et Su-In LEE. "A Unified Approach to Interpreting Model Predictions". In : *Advances in Neural Information Processing Systems*. Sous la dir. d'I. GUYON, U. Von LUXBURG, S. BENGIO, H. WALLACH, R. FERGUS, S. VISHWANATHAN et R. GARNETT. T. 30. Curran Associates, Inc., 2017. URL : https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

[123] Lerrel PINTO, James DAVIDSON, Rahul SUKTHANKAR et Abhinav GUPTA. "Robust adversarial reinforcement learning". In : *Proceedings of the 34th International Conference on Machine Learning*. PMLR. 2017, p. 2817-2826.

[124] John SCHULMAN, Filip WOLSKI, Prafulla DHARIWAL, Alec RADFORD et Oleg KLIMOV. *Proximal Policy Optimization Algorithms*. 2017. arXiv : 1707.06347 [cs.LG]. URL : https://arxiv.org/abs/1707.06347.

[125] Peter SUNEHAG et al. *Value-Decomposition Networks For Cooperative Multi-Agent Learning*. 2017. DOI : 10.48550/arxiv.1706.05296. arXiv : 1706.05296 [cs.AI].

[126] Josh TOBIN, Rachel FONG, Alex RAY, Jonas SCHNEIDER, Wojciech ZAREMBA et Pieter ABBEEL. "Domain randomization for transferring deep neural networks from simulation to the real world". In : *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, p. 23-30.

[127] Martin ABADI. "TensorFlow : learning functions at scale". In : *SIGPLAN Not.* 51.9 (sept. 2016), p. 1. ISSN : 0362-1340. DOI : 10.1145/3022670.2976746.

[128] Dario AMODEI, Chris OLAH, Jacob STEINHARDT, Paul CHRISTIANO, John SCHULMAN et Dan MANÉ. "Concrete problems in AI safety". In : *arXiv preprint arXiv :1606.06565* (2016).

[129] Anna L. BUCZAK et Erhan GUVEN. "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection". In : *IEEE Communications Surveys & Tutorials* 18.2 (2016), p. 1153-1176. DOI : 10.1109/COMST.2015.2494502.

[130] Yaroslav GANIN, Evgeniya USTINOVA, Hana AJAKAN, Pascal GERMAIN, Hugo LAROCHELLE, François LAVIOLETTE, Mario MARCHAND et Victor LEMPITSKY. "Domain-adversarial training of neural networks". In : *Journal of Machine Learning Research* 17.1 (2016), p. 2096-2030.

[131] NATO COOPERATIVE CYBER DEFENCE CENTRE OF EXCELLENCE. *Cyber Defence*. https://ccdcoe.org/. Consulté en juin 2025. 2016.

[132] Frans A. OLIEHOEK et Christopher AMATO. *A Concise Introduction to Decentralized POMDPs*. Springer Briefs in Intelligent Systems. Springer, 2016. ISBN : 978-3-319-28927-4. DOI : 10.1007/978-3-319-28929-8.

[133] Sebastian BACH, Alexander BINDER, Grégoire MONTAVON, Frederick KLAUSCHEN, Klaus-Robert MÜLLER et Wojciech SAMEK. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation". In : *PloS one* 10.7 (2015), e0130140.

[134] Javier GARCIA et Fernando FERNÁNDEZ. "A comprehensive survey on safe reinforcement learning". In : *Journal of Machine Learning Research* 16.1 (2015), p. 1437-1480.

[135] J. MORTEZA ET AL. "A method in security of wireless sensor network based on optimized artificial immune system in multi-agent environments". In : *arXiv preprint arXiv :1508.01706* (2015).

[136] J.-P. JAMONT et M OCCELLO. *Meeting the challenges of decentralised embedded applications using multi-agent systems. International Journal of Agent-Oriented Software Engineering 5 (1), 22–68*. 2015.

[137] Faris KETI et Shavan ASKAR. "Emulation of Software Defined Networks Using Mininet in Different Simulation Environments". In : *2015 6th International Conference on Intelligent Systems, Modelling and Simulation*. 2015, p. 205-210. DOI : 10.1109/ISMS.2015.46.

[138] George RUSH, Daniel R. TAURITZ et Alexander D. KENT. "Coevolutionary Agent-Based Network Defense Lightweight Event System (CANDLES)". In : *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*. GECCO Companion '15. Madrid, Spain : Association for Computing Machinery, 2015, p. 859-866. ISBN : 9781450334884. DOI : 10.1145/2739482.2768429.

[139] Emmanouil VASILOMANOLAKIS, Shankar KARUPPAYAH, Max MÜHLHÄUSER et Mathias FISCHER. "Taxonomy and survey of collaborative intrusion detection". In : *ACM Computing Surveys (CSUR)* 47.4 (2015), p. 1-33.

[140] Sonia CHERNOVA et Andrea L THOMAZ. "Robot learning from human teachers". In : *Synthesis lectures on artificial intelligence and machine learning*. T. 8. 3. Morgan & Claypool Publishers, 2014, p. 1-121.

[141] Aurélie BEYNIER et Alain MOUADDIB. "A Decentralized Approach for Reinforcement Learning in Cooperative Multi-agent Systems". In : *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*. 2013, p. 163-168.

[142] Daniel S. BERNSTEIN ET AL. "The Complexity of Decentralized Control of Markov Decision Processes". In : *CoRR* abs/1301.3836 (2013). DOI : 10.48550/arxiv.1301.3836. arXiv : 1301.3836.

[143] Mohammad H. MANSHAEI, Quanyan ZHU, Tansu ALPCAN, Tamer BAÇŞAR et Jean-Pierre HUBAUX. "Game theory meets network security and privacy". In : *ACM Computing Surveys (CSUR)* 45.3 (2013), p. 1-39.

[144] Paul THERON. "ICT Resilience as Dynamic Process and Cumulative Aptitude". In : *Critical Information Infrastructure Protection and Resilience in the ICT Sector* 3 (jan. 2013), p. 1-35. DOI : 10.4018/978-1-4666-2964-6.ch001.

[145] *Prometheus - Monitoring system and time series database*. Accessed : 2024-11-25. 2012. URL : https://prometheus.io.

[146] Álvaro CARRERA et Carlos Angel IGLESIAS. "Multi-agent Architecture for Heterogeneous Reasoning under Uncertainty Combining MSBN and Ontologies in Distributed Network Diagnosis". In : *Proceedings of the 2011 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2011, Campus Scientifique de la Doua, Lyon, France, August 22-27, 2011*. Sous la dir. d'Olivier BOISSIER, Jeffrey BRADSHAW, Longbing CAO, Klaus FISCHER et Mohand-Said HACID. IEEE Computer Society, 2011, p. 159-162. DOI : 10.1109/WI-IAT.2011.106.

[147] Marco CARVALHO et Carlos PEREZ. "An evolutionary multi-agent approach to anomaly detection and cyber defense". In : *7th Annual Workshop on Cyber Security and Information Intelligence Research*. 2011, p. 1-1.

[148] Marc Peter DEISENROTH et Carl Edward RASMUSSEN. "PILCO : A model-based and data-efficient approach to policy search". In : *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 2011, p. 465-472.

[149] J. HAACK ET. AL. "Ant-based cyber security". In : *2011 Eighth International Conference on Information Technology : New Generations*. IEEE. 2011, p. 918-926.

[150] BEYNIER, AURÉLIE ET AL. "DEC-MDP / DEC-POMDP". In : *Markov Decision Processes in Artificial Intelligence*. 2010, p. 277-313. URL : https://hal.science/hal-00969197.

[151] Barbara KORDY, Sjouke MAUW, Sasa RADOMIROVIC et Patrick SCHWEITZER. "Foundations of Attack-Defense Trees". In : *Formal Aspects of Security and Trust - 7th International Workshop, FAST 2010, Pisa, Italy, September 16-17, 2010. Revised Selected Papers*. Sous la dir. de Pierpaolo DEGANO, Sandro ETALLE et Joshua D. GUTTMAN. T. 6561. Lecture Notes in Computer Science. Springer, 2010, p. 80-95. DOI : 10.1007/978-3-642-19751-2_6.

[152] Jelena MIRKOVIC, Terry V. BENZEL, Ted FABER, Robert BRADEN, John T. WROCLAWSKI et Stephen SCHWAB. "The DETER project : Advancing the science of cyber security experimentation and test". In : *2010 IEEE International Conference on Technologies for Homeland Security (HST)*. 2010, p. 1-7. DOI : 10.1109/THS.2010.5655108.

[153] Stuart RUSSELL et Peter NORVIG. *Artificial Intelligence : A Modern Approach*. 3e éd. Prentice Hall, 2010.

[154] Robin SOMMER et Vern PAXSON. "Outside the Closed World : On Using Machine Learning for Network Intrusion Detection". In : *2010 IEEE Symposium on Security and Privacy*. 2010, p. 305-316. DOI : 10.1109/SP.2010.25.

[155] Eric M HOLLOWAY et Gary B LAMONT. "Self organized multi-agent entangled hierarchies for network security". In : *Proc. of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference : Late Breaking Papers*. 2009, p. 2589-2596.

[156] Gary B LAMONT et Eric M HOLLOWAY. "Military network security using self organized multi-agent entangled hierarchies". In : *Proc. of the Genetic and Evolutionary Computation Conf.* 2009, p. 2559-2566.

[157] Gauthier PICARD, Jomi Fred HÜBNER, Olivier BOISSIER et Marie-Pierre GLEIZES. "Réorganisation et auto-organisation dans les systèmes multi-agents [présentation courte]". In : *Systèmes Multi-Agents, Génie logiciel multi-agents - JFSMA 09 - Dix Septièmes Journées Francophones sur les Systèmes Multi-Agents, Lyon, France, October 19-21, 2009*. Sous la dir. de Zahia GUESSOUM et Salima HASSAS. Cepadues Editions, 2009, p. 89-98.

[158] Xiaolin CUI, Xiaobin TAN, Yong ZHANG et Hongsheng XI. "A Markov Game Theory-Based Risk Assessment Model for Network Information System". In : *International Conference on Computer Science and Software Engineering, CSSE 2008, Volume 3 : Grid Computing / Distributed and Parallel Computing / Information Security, December 12-14, 2008, Wuhan, China*. IEEE Computer Society, 2008, p. 1057-1061. DOI : 10.1109/CSSE.2008.949.

[159] Marie-Pierre GLEIZES, Valérie CAMPS, Jean-Pierre GEORGÉ et Davy CAPERA. "Engineering Systems Which Generate Emergent Functionalities". In : *Engineering Environment-Mediated Multi-Agent Systems*. Sous la dir. de Danny WEYNS, Sven A. BRUECKNER et Yves DEMAZEAU. Berlin, Heidelberg : Springer Berlin Heidelberg, 2008, p. 58-75. ISBN : 978-3-540-85029-8.

[160] Jomi Fred HÜBNER, Jaime Simão SICHMAN et Olivier BOISSIER. "Developing organised multiagent systems using the MOISE$^+$ model : programming issues at the system and agent levels". In : *Int. J. Agent Oriented Softw. Eng.* 1.3/4 (2007), p. 370-395. DOI : 10.1504/IJAOSE.2007.016266.

[161] Jomi Fred HÜBNER, Jaime Simão SICHMAN et Olivier BOISSIER. "Using MOISE+ for a cooperative framework in MAS-based simulation". In : *Scalable Computing : Practice and Experience* 8.1 (2007), p. 59-70.

[162] Yoav SHOHAM et Kevin LEYTON-BROWN. "MULTIAGENT SYSTEMS Algorithmic, Game-Theoretic, and Logical Foundations". In : (2007). URL : http://www.masfoundations.org/mas.pdf.

[163] Guido BOELLA et Leendert van der TORRE. "A Foundational Ontology of Organizations and Roles". In : *Declarative Agent Languages and Technologies IV*. Sous la dir. de Matteo BALDONI et Ulle ENDRISS. Berlin, Heidelberg : Springer Berlin Heidelberg, 2006, p. 78-88. ISBN : 978-3-540-68961-4.

[164] Rafael H BORDINI, Jomi Fred HÜBNER et Michael WOOLDRIDGE. "Programming multi-agent systems in AgentSpeak using Jason". In : *International conference on autonomous agents and multiagent systems (AAMAS)*. 2006.

[165] Gauthier PICARD, Sehl MELLOULI et Marie-Pierre GLEIZES. "Techniques for Multi-agent System Reorganization". In : *Engineering Societies in the Agents World VI*. Sous la dir. d'Oğuz DIKENELLI, Marie-Pierre GLEIZES et Alessandro RICCI. Berlin, Heidelberg : Springer Berlin Heidelberg, 2006, p. 142-152. ISBN : 978-3-540-34452-0.

[166] Giovanna Di Marzo SERUGENDO, Marie-Pierre GLEIZES et Anthony KARAGEORGOS. "Self-Organisation and Emergence in MAS : An Overview". In : *Informatica (Slovenia)* 30 (2006), p. 45-54. URL : https://api.semanticscholar.org/CorpusID:7654883.

[167] Jacques FERBER, Olivier GUTKNECHT et Fabien MICHEL. "From Agents to Organizations : An Organizational View of Multi-agent Systems". In : *Agent-Oriented Software Engineering IV*. Sous la dir. de Paolo GIORGINI, Jörg P. MÜLLER et James ODELL. Berlin, Heidelberg : Springer Berlin Heidelberg, 2004, p. 214-230. ISBN : 978-3-540-24620-6.

[168] Eric A. HANSEN, Daniel S. BERNSTEIN et Shlomo ZILBERSTEIN. "Dynamic programming for partially observable stochastic games". In : *Proceedings of the 19th AAAI Conference on Artificial Intelligence*. 2004, p. 709-715.

[169] Olivier BOISSIER, Cosmin CARABELEA et Adina Magda FLOREA. "Autonomie dans les systèmes multi-agents. essai de classification". In : *Technique et Science Informatiques* 22 (oct. 2003), p. 191-204.

[170] Vladimir GORODETSKI, Igor KOTENKO et Oleg KARSAEV. "Multi-agent technologies for computer network security". In : *Comput. Syst. Sci. Eng.* 18.4 (2003), p. 191-200.

[171] Carlos GUESTRIN, Daphne KOLLER et Ronald PARR. "Efficient solution algorithms for factored MDPs". In : *Journal of Machine Learning Research* 4.Apr (2003), p. 283-301.

[172] Jomi Fred HÜBNER, Jaime Simão SICHMAN et Olivier BOISSIER. "A Model for the Structural, Functional, and Deontic Specification of Organizations in Multiagent Systems". In : *Advances in Artificial Intelligence, 16th Brazilian Symposium on Artificial Intelligence, SBIA 2002, Porto de Galinhas/Recife, Brazil, November 11-14, 2002, Proceedings*. Sous la dir. de Guilherme BITTENCOURT et Geber L. RAMALHO. T. 2507. Lecture Notes in Computer Science. Springer, 2002, p. 118-128. DOI : 10.1007/3-540-36127-8_12.

[173] Jomi Fred HÜBNER, Jaime Simão SICHMAN et Olivier BOISSIER. "MOISE+ : Towards a structural, functional, and deontic model for multi-agent organizations". In : *Proc. of the 1st Int. Joint Conf. on Autonomous Agents and Multiagent Systems* (2002), p. 501-502.

[174] Michael WOOLDRIDGE. *An Introduction to MultiAgent Systems*. John wiley & sons Ltd., 2002.

[175] Edmund H. DURFEE. "Distributed Problem Solving and Planning". In : *Multi-Agent Systems and Applications : 9th ECCAI Advanced Course, ACAI 2001 and Agent Link's 3rd European Agent Systems Summer School, EASSS 2001 Prague, Czech Republic, July 2–13, 2001 Selected Tutorial Papers*. Sous la dir. de Michael LUCK, Vladimír MAŘÍK, Olga ŠTĚPÁNKOVÁ et Robert TRAPPL. Berlin, Heidelberg : Springer Berlin Heidelberg, 2001, p. 118-149. ISBN : 978-3-540-47745-7. DOI : 10.1007/3-540-47745-4_6.

[176] Stefan AXELSSON. "Intrusion Detection Systems : A Survey and Taxonomy". In : *Technical report, Department of Computer Engineering, Chalmers University* (2000).

[177] Hamid R. BERENJI et David VENGEROV. *Learning, Cooperation, and Coordination in Multi-Agent Systems*. Technical Report IIS-00-10. White paper. Sunnyvale, CA, USA, 2000. URL : https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=c9e7a0e373ee4396b966c7485ccbe68d393f2e32.

[178] Jacques FERBER. *Multi-Agent Systems : An Introduction to Distributed Artificial Intelligence*. 1st. USA : Addison-Wesley Longman Publishing Co., Inc., 1999. ISBN : 0201360489.

[179] Francis HEYLIGHEN. *The Science Of Self-Organization And Adaptivity*. 1999. URL : citeseer.ist.psu.edu/heylighen99science.html.

[180] Lennart LJUNG. "System identification : theory for the user". In : *PTR Prentice Hall Information and System Sciences Series* (1999).

[181] Andrew Y NG, Daishi HARADA et Stuart RUSSELL. "Policy invariance under reward transformations : Theory and application to reward shaping". In : *ICML*. T. 99. 1999, p. 278-287.

[182] Tuomas W. SANDHOLM. "Distributed rational decision making". In : *Multiagent Systems : A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA, USA : MIT Press, 1999, p. 201-258. ISBN : 0262232030.

[183] Bruce SCHNEIER. "Modeling security threats". In : *Dr. Dobb's journal* 24.12 (1999).

[184] Nicholas R JENNINGS, Katia SYCARA et Michael WOOLDRIDGE. "A Roadmap of Agent Research and Development". In : *Autonomous Agents and Multi-Agent Systems* 1.1 (mars 1998), p. 7-38.

[185] Leslie Pack KAELBLING, Michael L. LITTMAN et Anthony R. CASSANDRA. "Planning and acting in partially observable stochastic domains". In : *Artificial Intelligence* 101.1-2 (1998), p. 99-134.

[186] Cynthia PHILLIPS et Laura Painton SWILER. "A Graph-Based System for Network-Vulnerability Analysis". In : *Proceedings of the 1998 Workshop on New Security Paradigms*. 1998. ISBN : 1581131682. DOI : 10.1145/310889.310919.

[187] Sepp HOCHREITER et Jürgen SCHMIDHUBER. "Long Short-Term Memory". In : *Neural Comput.* 9.8 (1997), p. 1735-1780. DOI : 10.1162/NECO.1997.9.8.1735.

[188] N. R. JENNINGS. "Coordination Techniques for Distributed Artificial Intelligence". In : *Foundations of Distributed Artificial Intelligence*. Sous la dir. de G. M. P. O'HARE et N. R. JENNINGS. Wiley, 1996, p. 187-210. URL : https://eprints.soton.ac.uk/252187/.

[189] Eitan ALTMAN. *Constrained Markov Decision Processes*. Rapp. tech. RR-2574. INRIA, mai 1995. URL : https://inria.hal.science/inria-00074109.

[190] Martin L. PUTERMAN. *Markov Decision Processes : Discrete Stochastic Dynamic Programming*. Wiley, 1994.

[191] Thomas D. SCHNEIDER et R. Michael STEPHENS. "Sequence logos : a new way to display consensus sequences". In : *Nucleic Acids Research* 18.20 (1990), p. 6097-6100. DOI : 10.1093/nar/18.20.6097.

[192] Michael P. GEORGEFF et Amy L. LANSKY. "Reactive Reasoning and Planning". In : *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*. AAAI-87 paper ID : 121. Morgan Kaufmann, 1987.

[193] Temple F. SMITH et Michael S. WATERMAN. "Identification of common molecular subsequences". In : *Journal of Molecular Biology* 147.1 (1981), p. 195-197. DOI : 10.1016/0022-2836(81)90087-5.

[194] Paul JACCARD. "Nouvelles Recherches Sur la Distribution Florale". In : *Bulletin de la Societe Vaudoise des Sciences Naturelles* 44 (jan. 1908), p. 223-70. DOI : 10.5169/seals-268384.

JOURNAUX

[1] Julien Soulé, Jean-Paul Jamont, Michel Occello, Louis-Marie Traonouez et Paul Théron. “Assisting Multi-Agent System Design with $\mathcal{M}$OISE$^+$ and MARL : The MAMAD Method”. In : *Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)* (2025). *Under revision*.

[2] Julien Soulé, Jean-Paul Jamont, Michel Occello, Louis-Marie Traonouez et Paul Théron. “MOISE+MARL : un cadre organisationnel pour l’explicabilité et le contrôle en apprentissage par renforcement multi-agent”. In : *Revue Ouverte d’Intelligence Artificielle (ROIA)* (2025). *Under revision*.

CONFERENCES INTERNATIONALES

[3] Julien Soulé, Jean-Paul Jamont, Michel Occello, Louis-Marie Traonouez et Paul Théron. “An Organizationally-Oriented Approach to Enhancing Explainability and Control in Multi-Agent Reinforcement Learning”. In : *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*. AAMAS ’25. Detroit, MI, USA : International Foundation for Autonomous Agents et Multiagent Systems, 2025, p. 1968-1976.

[4] Julien Soulé, Jean-Paul Jamont, Michel Occello, Louis-Marie Traonouez et Paul Théron. “Streamlining Resilient Kubernetes Autoscaling with Multi-Agent Systems via an Automated Online Design Framework”. In : *2025 IEEE 18th International Conference on Cloud Computing (CLOUD)*. 2025, p. 43-53. DOI : 10.1109/CLOUD67622.2025.00015.

[5] Julien Soulé, Jean-Paul Jamont, Michel Occello, Louis-Marie Traonouez et Paul Théron. “A MARL-Based Approach for Easing MAS Organization Engineering”. In : *Artificial Intelligence Applications and Innovations*. Sous la dir. d’Ilias Maglogiannis, Lazaros Iliadis, John Macintyre, Markos Avlonitis et Antonios Papaleonidas. Cham : Springer Nature Switzerland, 2024, p. 321-334. ISBN : 978-3-031-63223-5.

[6] Julien Soulé, Jean-Paul Jamont, Michel Occello, Paul Théron et Louis-Marie Traonouez. “Towards a Multi-Agent Simulation of Cyber-attackers and Cyber-defenders Battles”. In : *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2023, p. 3594-3599. DOI : 10.1109/SMC53992.2023.10394564.

CONFERENCES NATIONALES

[7] Julien Soulé, Jean-Paul Jamont, Michel Occello, Paul Théron et Louis-Marie Traonouez. “Une approche organisationnelle pour améliorer l’explicabilité et le contrôle dans l’apprentissage par renforcement multi-agent”. In : *33èmes Journées Francophones sur les Systèmes Multi-Agents (JFSMA 2025)*. Prix du meilleur article. Dijon, France : Association Française pour l’Intelligence Artificielle, juill. 2025. URL : https://hal.science/hal-05151654.

[8] Julien Soulé, Jean-Paul Jamont, Michel Occello, Louis-Marie Traonouez et Paul Théron. “Un outil pour la conception de SMA par apprentissage par renforcement et modélisation organisationnelle”. In : *32èmes Journées Francophones sur les Systèmes Multi-Agents (JFSMA 2024)*. Sébastien Picault. Cargèse, France : Cépaduès, nov. 2024. URL : https://hal.science/hal-04840721.

[9] Julien Soulé, Jean-Paul Jamont, Michel Occello, Louis-Marie Traonouez et Paul Théron. "Une approche basée sur l'apprentissage par renforcement pour l'ingénierie organisationnelle d'un SMA". In : *32èmes Journées Francophones sur les Systèmes Multi-Agents (JFSMA 2024)*. Sébastien Picault. Cargèse, France : Cépaduès, nov. 2024. URL : https://hal.science/hal-04840696.

[10] Julien Soulé, Jean-Paul Jamont, Michel Occello, Paul Théron et Louis-Marie Traonouez. "De l'organisation d'un système multi-agent de cyberdéfense". In : *31èmes Journées Francophones sur les Systèmes Multi-Agents (JFSMA 2023)*. JFSMA. Cépaduès, 2023, p. 54-54. URL : https://hal.science/hal-04165020.

[11] Julien Soulé, Jean-Paul Jamont, Michel Occello, Paul Théron et Louis-Marie Traonouez. "De l'organisation d'un système multi-agent de cyberdéfense". In : *Rendez-Vous de la Recherche et de l'Enseignement de la Sécurité des Systèmes d'Information (RESSI 2023)*. Mai 2023. URL : https://ressi2023.sciencesconf.org/450961/document.

[12] Julien Soulé, Jean-Paul Jamont, Michel Occello, Paul Théron et Louis-Marie Traonouez. "De l'organisation des systèmes multi-agents de cyber-défense". In : *Rencontres Jeunes Chercheurs en Intelligence Artificielle (RJCIA 2023)*. Strasbourg, France, juill. 2023. URL : https://hal.science/hal-04565426.

INDEX

ACD, 13
ACO, 15
Adaptation (C3), 21
Adversarial ML, 15
Adéquation organisationnelle, 169
Agent, 17
AICA, 15
Analyse (ANL), 126
Apprentissage par renforcement (RL), 76
Apprentissage par renforcement multi-agent (MARL), 77
Auto-organisation, 19
Autonomie, 18
Autonomie (C1), 21

C2, 13
COMA, 166
Company Infrastructure, 177
Contrôle (C4), 22
Coordination, 18
CSIRT, 13
Cyber-résilience, 13
Cyberdéfense, 12
Cybersécurité, 12
CybMASDE, 140

Drone Swarm, 184

Entraînement (TRN), 114
Explicabilité (C5), 22

IA, 12
IA distribuée, 17
IoC, 12
IQL, 166

Joint-Observation Prediction Model (JOPM), 105

LSTM, 75

MADDPG, 166
MAMAD, 91
MAPPO, 166
MASCARA, 15
Microservices Kubernetes, 181
MLP, 75
Modélisation (MOD), 102
MOISE+, 79
MOISE+MARL, 116

Observation Prediction Model (OPM), 75
Organisation, 18
Overcooked-AI, 173

P3R3, 13
Performance (C2), 21
Politique conjointe, 37
Politiques, 17
Predator-Prey, 174
Proportion d'intervention, 168
PyTorch, 166

QMIX, 166
Qualité des spécifications inférées, 169

RNN, 75
Récompense cumulée, 168
Réorganisation, 19

Score de cohérence, 169
Score de robustesse, 168
SMA, 17
SMA de Cyberdéfense, 17
Stratégies, 17

Taux de convergence, 168
Taux de violation des contraintes, 169
Transfert (TRF), 135

VDN, 166

Warehouse Management, 175
World Model, 75

Écart-type des récompenses, 168