

représentant leur importance ; il peut aussi être sollicité pour apprendre l'importance des anomalies en fonction de leur apparition dans le monde réel.

4.1. Architecture de MAS4AT

La figure 2 ci-dessous introduit cette architecture. Plus précisément, le sous-système OpMAS (*Operative Multi-Agent System*) regroupe un ensemble d'agents-navires, chacun représentant un navire surveillé par I2C dans le monde réel. Pour chaque agent-navire est associée une valeur appelée *SBv* (*Ship Behaviour value*) qui représente le niveau de suspicion associé au navire. *SBv* est calculée en combinant les différentes anomalies associées au navire. Une alerte est levée par OpMAS si cette valeur est plus grande qu'un seuil donné.

Le sous-système PaMAS (*Parameter Adjustment Multi-Agent System*) complète OpMAS et modélise les anomalies sous forme de valeurs numériques en utilisant deux types d'agents : les agents-anomalies et les agents-paramètres. Les agents-anomalies représentent les anomalies associées aux navires. La valeur de ces anomalies est décrite par trois paramètres, chacun représentés par un agent-paramètre (Mano *et al.*, 2010). Ainsi, PaMAS distingue la valeur que prend l'anomalie lors de son apparition (paramètre *Init*), la valeur que prend l'anomalie lorsqu'elle persiste dans le temps (paramètre *Incr*) et enfin la valeur que prend l'anomalie après sa disparition (paramètre *Decr*).

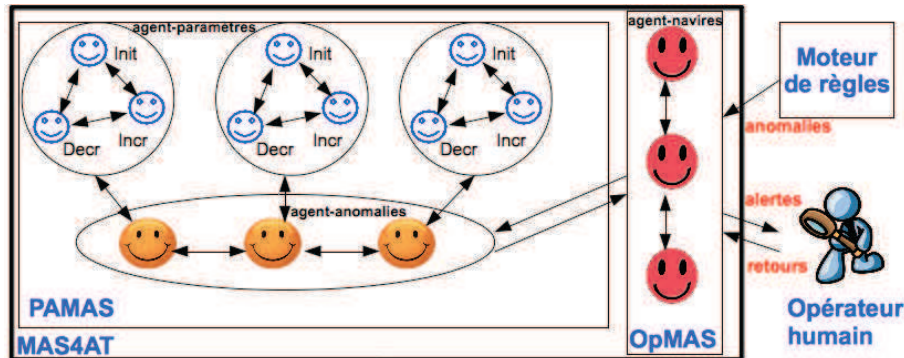


Figure 2. Architecture de MAS4AT

4.2. Les agents-navires dans OpMAS

Comme indiqué précédemment, chaque agent-navire représente un navire évoluant dans la zone sous surveillance et est doté d'un comportement lui permettant d'atteindre ses objectifs. Cette section décrit ce comportement en distinguant les capacités individuelles des agents-navires de leur comportement collectif ; ces

comportements sont les briques de base des algorithmes d'apprentissage introduits en section 5.1.

4.2.1 Capacités individuelles des agents- navires

Un agent-navire est doté de capacités lui permettant (i) de calculer la valeur de son comportement et de la représenter sous forme de courbe, (ii) de déclencher une alerte si cette valeur est supérieure à un seuil fixé par les opérateurs impliqués dans la surveillance maritime, et (iii) de déclencher une phase d'apprentissage des valeurs d'anomalies si l'alerte déclenchée est incorrecte ou si une alerte n'a pas été déclenchée alors que c'était nécessaire.

4.2.1.1 Évaluation et représentation du comportement

La valeur de comportement d'un agent-navire, appelée SBv (*Ship Behaviour value*), représente la suspicion associée à son comportement. Elle est calculée en tenant compte des anomalies envoyées par le moteur de règles par la fonction suivante :

$$SBv = \sum_{i=0}^n (Init(a_i) \times nb('Init', a_i)) \\ + (Incr(a_i) \times nb('Incr', a_i)) - (Decr(a_i) * nb('Decr', a_i))$$

L'idée est de cumuler l'importance des différentes anomalies associées à un navire en s'appuyant sur les valeurs des paramètres des anomalies en question et en prenant en compte le temps et le nombre d'apparitions de ces anomalies. Comme indiqué précédemment, les paramètres de l'anomalie sont au nombre de trois : ils représentent la valeur initiale de l'anomalie lorsqu'elle apparaît, la valeur qu'elle prend lorsqu'elle persiste dans le temps, et la valeur qu'elle prend lorsqu'elle disparaît (cette valeur correspond à l'effet de dissipation de l'anomalie ; on garde ainsi trace un certain temps qu'une anomalie s'est produite). Plus précisément, pour chaque anomalie relative à un agent-navire, la fonction additionne : 1) le nombre de fois où l'anomalie s'est produite multiplié par la valeur de son paramètre Init (Init(a_i)) et 2) le nombre de pas de temps où l'anomalie perdure (nb('Incr', a_i)) multiplié par la valeur de son paramètre Incr (Incr(a_i)). Elle retranche ensuite au résultat de cette addition le nombre de pas de temps (nb('Decr', a_i)) où l'anomalie considérée n'est pas perçue multiplié par la valeur de son paramètre Decr (Decr(a_i)).

Le niveau d'anomalie d'un navire résultat de son comportement peut être représenté comme une fonction mathématique. Il est ainsi possible, à un instant donné, c'est-à-dire pour une situation donnée, (i) d'indiquer la valeur d'un point de la courbe et (ii) de donner l'inéquation relative au seuil d'alerte fixé par les opérateurs.

Par exemple, la figure 3 illustre l'exemple d'un navire dont le comportement est composé de l'anomalie a1 entre pas0 et pas13, puis des anomalies a1 et a2 entre pas14 et pas19. Les cercles représentent la valeur calculée du comportement du navire à un instant donné. Les cercles blancs représentent l'apparition d'anomalies tandis que les noirs ne sont pas liés à l'apparition d'événements suspicieux.

Ainsi, dans l'exemple de la figure 3, à pas5, nous avons pour valeur de SBv : $1 \times \text{Init}(a_1) + 5 \times \text{Incr}(a_1)$. En effet, à pas5, a1 est apparu 1 fois (un cercle blanc représente l'apparition de a1) et a duré 5 pas de temps (5 cercles noirs représentent le fait que a1 perdure). A pas11, a1 est apparu deux fois et a perduré de 9 pas de temps et a disparu pendant 2 pas de temps (avant de réapparaître). À pas18, à la fois a1 et a2 sont présentes et donc combinées. Au total, l'anomalie a1 est apparue 3 fois, a persisté durant 12 pas de temps et a disparue 4 pas de temps avant de réapparaître. De son côté, a2 est apparue 1 fois et a persisté durant 4 pas de temps. À chacun de ces pas de temps, l'agent-navire peut construire l'inéquation qui correspond à sa situation comportementale vis-à-vis du seuil d'alerte fixé par les opérateurs.

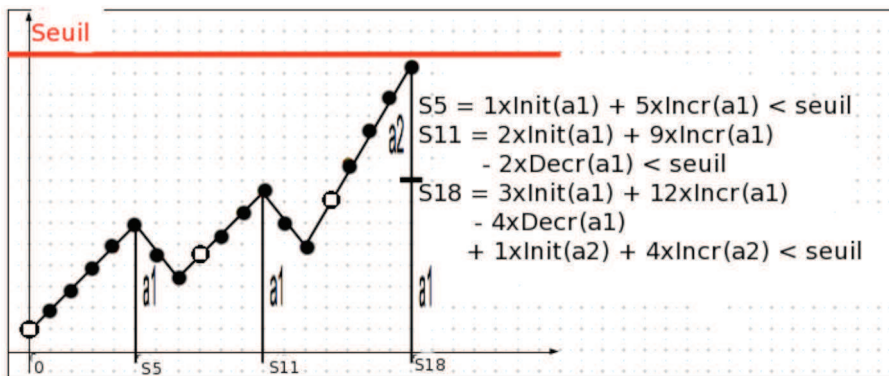


Figure 3. Calcul du comportement d'un navire

4.2.1.2 Déclenchement d'alerte et apprentissage

En plus de savoir évaluer et représenter la valeur de son comportement, un agent-navire a également la capacité de déclencher une alerte auprès des opérateurs si la valeur du comportement du navire qu'il représente est supérieure au seuil d'alerte (qui est en fait le seuil de déclenchement).

Un agent-navire a aussi la capacité de déclencher l'apprentissage par auto-ajustement des valeurs des anomalies qui lui sont associées. Plus précisément, un agent-navire peut solliciter les agents-anomalies concernés qui vont eux-mêmes déclencher l'apprentissage par auto-ajustement des valeurs de leurs paramètres Init, Incr et Decr. Les différents algorithmes d'apprentissage sont donnés en section 5.1.

4.2.1.3 Capacités d'interaction

Enfin, pour pouvoir mettre en œuvre les capacités présentées précédemment (calcul de comportement, déclenchement d'alertes, phase d'apprentissage), un agent-navire doit pouvoir interagir avec trois composants de I2C à savoir :

- Le moteur de règles, pour recevoir les anomalies détectées ; ces anomalies sont utilisées pour calculer la valeur du comportement du navire (ces valeurs sont dans PAMAS).

- PAMAS, pour connaître les valeurs des anomalies mais aussi pour lancer une phase d'apprentissage par auto-adaptation de ces valeurs.

- Les opérateurs impliqués dans la surveillance maritime, qui reçoivent les alertes levées par OpMAS et qui, après vérification, les valident ou les invalident. Les opérateurs peuvent également directement solliciter les agents-navires d'OpMAS pour leur indiquer des alertes non identifiées.

4.2.2 Comportement collectif des agents-navires

Un agent-navire coopère de manière directe, par envoi de messages, avec d'autres agents-navires puisqu'il peut leur demander des informations telles que leur niveau de suspicion (*i.e.* leur valeur *SBv* de comportement), ou leur indiquer son niveau de suspicion. Il peut également indiquer s'il a déclenché une alerte afin que les navires qui partagent avec lui des anomalies ou qui naviguent dans son voisinage puissent recalculer leur valeur de comportement et éventuellement déclencher une alerte. Ce comportement coopératif de déclenchement d'alerte est important pour déterminer l'ensemble des navires impliqués dans une alerte.

Les agents-navires peuvent aussi coopérer de manière indirecte. Cela est important dans le contexte de la surveillance maritime afin de prendre en compte les activités illégales réalisées par plusieurs navires dont le comportement individuel est normal, mais dont le comportement collectif est anormal. Dans un tel cas de figure, le moteur de règles est inutile puisqu'il ne détecte aucune anomalie (individuelle). Pour pallier ce problème, OpMAS introduit la notion de stigmergie comme mécanisme de détection. Un agent-navire peut donc marquer sa position dans l'environnement dans le but d'informer les autres navires d'un événement qui peut les intéresser. Tout comme dans les cas des traces de phéromones des fourmis, ces marques sont peu coûteuses en ressources et s'avèrent être des sources d'informations utiles et précieuses (Tsankova, 2004). Une marque est déposée par un agent-navire ; elle possède une valeur critique et un taux de persistance (d'évaporation) traduisant sa durée de vie. Quand un autre agent-navire perçoit une marque, il peut à son tour l'utiliser si nécessaire dans le calcul de sa valeur de comportement et éventuellement déclencher une alerte.

4.3. Les agents-anomalies et agents-paramètres dans PAMAS

Cette section décrit les capacités individuelles et les comportements collectifs des agents-anomalies et des agents-paramètres. Ces capacités individuelles et ce comportement collectif sont les briques de base des algorithmes d'apprentissage introduits en section 5.1.

4.3.1. Capacités individuelles des agents-anomalies

Les agents-anomalies disposent des capacités individuelles relatives à l'ajustement des valeurs de leurs paramètres. Les agents-anomalies peuvent déclencher l'ajustement de tous leurs paramètres. Cette capacité est utile en phase d'apprentissage car les agents-anomalies concernés par une alerte invalidée vont être sollicités par l'agent-navire correspondant pour auto-adaptation de leurs valeurs. Les agents-anomalies vont alors solliciter leurs agents-paramètres pour qu'ils s'ajustent (*i.e.* changent leurs valeurs Init, Incr et Decr). Les agents-anomalies peuvent également déterminer s'il y a eu un ajustement de leurs paramètres. S'il n'y a pas eu d'ajustement, ils peuvent décider d'en imposer en fonction des ajustements réalisés par les autres agents-anomalies des alertes dans lesquels ils sont impliqués. Ils peuvent donc :

- demander l'ajustement de leurs agents-paramètres,
- calculer et communiquer leur importance, en tenant compte des poids de leurs agents-paramètres,
- calculer et communiquer leur criticité, qui résulte de la criticité des agents-paramètres qui les composent.

4.3.2. Capacités individuelles des agents-paramètres

Les agents-paramètres disposent des capacités individuelles suivantes :

- S'ajuster *i.e.* auto-ajuster leur valeur évoluant dans un intervalle définissant l'espace de recherche du paramètre défini a priori par l'opérateur. Ils utilisent pour cela un *Adaptive Value Tracker* (Lemouzy *et al.*, 2011). Un AVT est un système multi-agent capable d'estimer la valeur d'une variable donnée en fonction de retours perçus dans son environnement.

- Choisir de s'ajuster, en fonction de leurs ajustements passés (prise en compte de l'historique des ajustements) et de la demande formulée par l'opérateur qui peut être : (i) il faut monter au-dessus du seuil ou (ii) il faut descendre en dessous du seuil. À l'issue de ce choix, les agents-paramètres déclenchent ou non l'ajustement.

- Communiquer leur valeur à leur anomalie.

- Calculer et communiquer leur importance au sein de l'anomalie afin de mener au mieux l'auto-ajustement. Là, c'est le nombre de fois où une anomalie s'est produite, le temps qu'elle a duré ou le temps qu'il a fallu avant qu'elle ne se dissipe, qui est pris en compte.

– Calculer et communiquer leur criticité, qui est une valeur numérique représentant la latitude qu’ils ont à s’ajuster en accord avec leur espace de recherche. Cette latitude peut être très faible si, par exemple, la valeur du paramètre est très proche de la borne supérieure de l’intervalle définissant son espace de recherche et que le retour de l’opérateur lui demande de monter (au dessus du seuil).

4.3.3. Comportement collectif des agents-anomalies et agents-paramètres

Au même titre que les agents-navires coopèrent entre eux de manière directe, par envoi de messages, pour se communiquer leur niveau de suspicion, les agents-anomalies coopèrent entre eux pour se communiquer leur criticité. Ainsi, lorsque les agents paramètres de toutes les anomalies concernées par une situation d’apprentissage refusent de s’ajuster, les agents-anomalies vont désigner collectivement, à partir de l’échange de leur criticité, celles(s) qui va (vont) se dévouer pour ajuster leur paramètres : c’est (ce sont) l’ (les) agent-anomalie(s) le (les) moins critique(s) qui demandera (demanderont) l’ajustement de leur agents-paramètres.

De manière analogue, les agents-paramètres coopèrent aussi entre eux de manière directe, par envoi de messages, pour se communiquer leur importance et leur criticité. L’importance est prise en compte lors de l’ajustement d’un agent-paramètre car plus un agent est important, plus une modification de sa valeur va avoir des conséquences significatives. La criticité est également prise en compte lors d’un ajustement, notamment lorsque les agents paramètres de toutes les anomalies concernées par une situation d’apprentissage refusent de s’ajuster. Dans ce cas, les agents-paramètres vont désigner collectivement, à partir de l’échange de leur criticité, celui (ceux) qui va (vont) se dévouer pour auto-ajuster sa (leur) valeur(s) : c’est (ce sont) l’ (les) agent(s)-paramètres(s) le (les) moins critique(s) qui s’auto-ajustera (s’auto-ajusteront).

5. Apprentissage dans MAS4AT : mise en œuvre et évaluation

Cette section présente d’abord les algorithmes d’apprentissage développés dans le système multi-agent MAS4AT. Elle explicite ensuite comment ce SMA a été mis en œuvre. Elle décrit enfin les premières expérimentations effectuées en présentant deux principaux cas d’étude et les résultats obtenus. Cette dernière section analyse plus particulièrement les résultats de l’apprentissage et notamment la convergence et l’ajustement.

5.1. Algorithmes d’apprentissage

Un des objectifs de MAS4AT est l’apprentissage par les agents de PaMAS des valeurs des anomalies détectées et transmises par le moteur de règles. Cet apprentissage se fait lorsqu’une alerte levée par OpMAS est invalidée par un opérateur (alerte erronée) ou lorsque l’opérateur détecte une alerte non levée par MAS4AT. L’ajustement consiste à ajuster la valeur des paramètres de chaque

anomalie impliquée dans l'alerte correspondante. Plus précisément, les agents-paramètres de PaMAS s'auto-ajustent pour que leurs valeurs soient en adéquation avec le retour de l'opérateur. Cependant, même si l'ajustement repose essentiellement sur les agents-paramètres, chaque type d'agent de MAS4AT participe à l'ajustement, chacun avec un rôle particulier.

5.1.1. Les agents-navires dans l'apprentissage

Comme nous l'avons vu en section 4, le rôle des agents-navires est de représenter, dans l'environnement de MAS4AT, les navires réels surveillés par le système I2C. Ces agents reçoivent des anomalies en provenance du moteur de règles, calculent la valeur SBv du comportement des navires qu'ils représentent et déclenchent une alerte lorsque cela est nécessaire, *i.e.* lorsque que SBv dépasse un seuil d'alerte. Lorsqu'une alerte est erronée ou non levée et qu'un retour négatif (*i.e.* l'alerte levée est invalidée ou une alerte non levée est détectée) est renvoyé par l'opérateur, les agents-navires provoquent alors un ajustement selon l'algorithme suivant.

```

Algorithme Ajustement
Début
  Paralléliser
  | appeler AjustementAnomalies
  Fin parallélisation
   $SBv = \text{somme}(\text{valeur\_agents-paramètres})$ 
  Si  $SBv == SBv_{\text{Anterieur}}$  Alors
  | appeler AjustementAnomaliesMinimal
  Fin Si
Fin

```

Algorithme 1. Les agents-navires dans l'ajustement

Cet algorithme indique que lorsqu'un retour négatif est reçu, l'agent-navire l'ayant reçu diffuse ce retour aux agents-anomalies concernés en leur demandant de s'ajuster. Les agents-anomalies s'ajustent (*cf.* section 5.1.2.) et renvoient leurs nouvelles valeurs à l'agent-navire qui va vérifier si la nouvelle valeur SBv calculée est conforme au retour opérateur, c'est-à-dire au-dessous du seuil d'alerte si une alerte non valide a été déclenchée, ou au-dessus du seuil d'alerte si l'opérateur a indiqué à MAS4AT qu'il aurait dû déclencher une alerte. Pour éviter le sur-apprentissage, les agent-navires prennent en compte la nouvelle valeur de SBv et la transmettent aux opérateurs. Un nouvel ajustement sera demandé aux agents-anomalies uniquement après réception d'un nouveau feedback. Cependant, si cette nouvelle valeur de comportement est égale à la valeur de comportement avant ajustement, les agents-navires vont forcer l'ajustement d'au moins un des paramètres des anomalies impliquées. Ce cas de figure correspond au cas où l'ensemble des agents-paramètres des anomalies concernées par une situation d'apprentissage ont refusé de s'ajuster (*cf.* 4.3.2 : capacité *choisir de s'ajuster*).

5.1.2. Les agents-anomalies et les agents-paramètres dans l'apprentissage

Les agents-anomalies représentent la valeur des anomalies envoyées par le moteur de règles aux agents-navires. Plus précisément, cette valeur est décrite comme la combinaison des trois agents-paramètres *Init*, *Incr* et *Decr* (cf. section 4.3.). Lors de la phase d'apprentissage, les agents-anomalies vont utiliser l'algorithme ci-dessous.

Algorithme AjustementAnomalies

Début

Paralléliser

| appeler **AjustementParamètres**

Fin parallélisation

Fin

Algorithme 2a. Les agents-anomalies dans l'ajustement

Suite à l'ajustement des agent-paramètres, les agent-anomalies transmettent leurs valeurs aux agent-navires qui vont calculer la nouvelle valeur de *SBv*. Si cette valeur n'a pas changé, l'agent-navire concerné va considérer qu'aucun ajustement n'a été fait. Il va donc déclencher un ajustement minimal et forcer l'ajustement d'au moins un des agents-anomalies impliqués, selon l'algorithme suivant.

Algorithme AjustementAnomaliesMinimal

Début

Calculer criticité agents-anomalies

Désigner l'(s) agent(s)-anomalie(s) le(s) moins critique(s)

*Appeler **AjustementParamètresMinimal***

Fin

Algorithme 2b. Ajustement minimal des agent-anomalies

Lorsque les agent-paramètres reçoivent une demande d'ajustement de la part de leur agent-anomalie, l'algorithme suivant est utilisé.

Algorithme AjustementParamètres

Début

Calculer $\text{SensAdjust} = \text{retourOpérateur} * \text{dynamique} * \text{importance} +$
 $(1 - \text{dynamique}) * \text{SensAdjustPrécédent}$

Si (*signe*(*SensAdjust*) == *retour*) *Alors*

| **Ajuster**

Fin Si

Fin

Algorithme 3a. Ajustement des agents-paramètres

Les agents-paramètres se basent sur le retour reçu de l'opérateur et sur l'historique de leur évolution pour décider de procéder ou non à l'ajustement. Si aucun retour n'est fait par l'opérateur, les agents-paramètres vont considérer un retour positif implicite. Cela leur permet de renforcer leur confiance dans la valeur qu'ils attribuent à leur paramètre propre. En revanche, lors de la réception d'un retour négatif, les agents-paramètres vont consulter l'historique de leur évolution pour prendre la décision d'ajustement ou non.

L'historique de l'évolution d'un agent-paramètre est un indicateur des dernières actions de l'agent. Cela peut être vu comme la préférence de l'agent à augmenter ou diminuer sa valeur en fonction des retours reçus. Cet historique est représenté par le terme $(1 - \text{dynamique}) * \text{SensAdjustPrécédent}$. La variable *dynamique* représente la dynamique de l'environnement et va permettre de pondérer l'importance que l'on souhaite donner à la situation courante par rapport aux situations passées. La variable *SensAdjustPrécédent* est la préférence de l'agent à augmenter ou abaisser sa valeur. On associe à ce résultat le calcul $\text{retourOpérateur} * \text{dynamique} * \text{importance}$. La variable *retourOpérateur* est l'action demandée à l'agent-paramètre, *i.e.* augmenter ou abaisser la valeur globale de l'anomalie. La variable *importance* indique la part prise par l'agent dans la situation courante ; cette valeur pondère son action. Le résultat de cette expression permet donc à l'agent-paramètre de prendre une décision sur l'action à effectuer. En effet, si le signe de *SensAdjust* est celui du retour perçu, alors l'agent ajuste sa valeur. Dans le cas contraire, l'agent ne fera rien car cela va à l'encontre de sa préférence. Ainsi, si un retour négatif demandant d'abaisser la valeur globale est reçu par un agent qui a eu tendance à augmenter sa valeur, cet agent préférera ne pas agir et laisser faire les agents plus aptes à aller dans le sens demandé. Une fois que tous les agents-paramètres ont pris une décision, et participé ou non à l'ajustement de l'anomalie, les nouvelles valeurs sont envoyées à l'agent-anomalie correspondant.

Algorithme AjustementParamètresMinimal

Début

Calculer criticité agents-paramètres

Désigner l'(s) agent(s)-paramètre(s) le(s) moins critique(s)

Ajuster

Fin

Algorithme 3b. Ajustement minimal des agent-paramètres

Si chaque agent-paramètre de l'anomalie décide de ne pas s'ajuster car le retour demande un ajustement inverse à leurs préférences, on est dans le cas d'un ajustement minimal. Les agents-paramètres vont alors échanger leurs criticités. Cet indicateur, géré par l'*Adaptive Value Tracker*, va permettre aux agents-paramètres (i) de déterminer celui (ou ceux) qui est (sont) le(s) moins contraint(s) et (ii) de forcer son (leur) ajustement(s). Cette technique permet d'éviter une boucle infinie au niveau des agents-navires. En effet, si aucun paramètre d'aucune anomalie ne s'ajuste, la nouvelle valeur *SBv* calculée ne sera pas différente de sa valeur

précédente. L'agent-navire demandera alors à nouveau aux agents-paramètres, et donc aux agents-anomalies, de s'ajuster car la valeur SBv n'a pas évolué et n'est toujours pas conforme au retour reçu. Ainsi, on évite de déstabiliser le système. L'algorithme d'ajustement minimal des paramètres est l'algorithme 3b.

5.2. Exploitation de MAS4AT

Le fonctionnement de MAS4AT prend en compte différents éléments. Dans un premier temps, il faut un environnement et des navires qui agissent dans cet environnement. De plus, un moteur de règles doit être capable d'observer ces navires et d'analyser leurs comportements pour envoyer des anomalies au SMA. Enfin, un opérateur doit être capable de comparer les alertes levées par le SMA avec ce qu'il se passe dans l'environnement afin de valider ou invalider les alertes levées, et dans ce dernier cas, retourner une erreur à MAS4AT. Nous avons donc construit une plateforme de simulation prenant en compte ces différents éléments pour nous permettre de tester et d'évaluer le fonctionnement de MAS4AT. Comme nous pouvons le voir dans la figure 4, cette plateforme de simulation est constituée de trois composants en plus du système MAS4AT : le *Générateur de Scénarii*, le *Moteur de Règles* et l'*Opérateur Humain*.

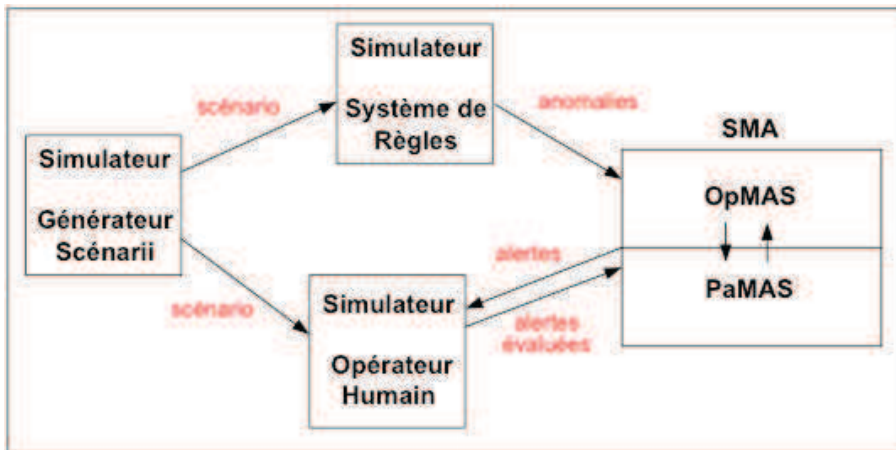


Figure 4. Architecture globale de la plateforme de simulation

Le Générateur de scénarii construit un nombre donné de navires et d'anomalies différents. Ensuite, un scénario aléatoire est créé et associé à chaque navire. Un scénario est une suite non prévisible d'anomalies qui se produisent sur un navire. Le générateur est cependant construit de manière à éviter des scénarios improbables. Toutes les données créées sont aussi des connaissances partagées avec les deux autres composants de la plateforme de simulation en tant que connaissance partagée.

Le *Moteur de Règles* simule le fonctionnement du moteur de règles du système I2C. Pour chaque navire, ce composant va lire séquentiellement le scénario associé et transmettre les anomalies perçues au système MAS4AT. Chaque anomalie représente dans I2C le résultat d'une analyse par le simulateur du moteur de règles et le déclenchement d'une règle particulière. Ce sont les valeurs de ces anomalies que devront apprendre les agents de MAS4AT et qui seront utilisées pour calculer la valeur du comportement des navires.

L'*Opérateur* est le composant qui simule l'analyse et le retour des opérateurs humains impliqués dans la surveillance maritime. Au sein de la plateforme de simulation, lorsqu'une alerte est levée par MAS4AT, elle est envoyée au composant *Opérateur* qui va comparer les valeurs des anomalies qui constituent l'alerte avec les valeurs réelles. Dans le cas où l'alerte a été levée de manière erronée, l'*Opérateur* envoie un retour négatif à MAS4AT. Sinon, aucun retour n'est envoyé et le SMA l'interprétera comme un retour positif. Le composant *Opérateur* est aussi capable de signaler au SMA qu'une alerte aurait dû être levée à un instant donné et que rien n'a été fait, ce qui est aussi interprété comme un retour négatif par MAS4AT.

Cette plateforme a été conçue pour remplacer les différents éléments qui constitueront le système I2C final et qui seront en lien direct avec le système multi-agent. Ainsi, outre les entrées et les sorties nécessaires à son fonctionnement, chaque composant est un module indépendant qui peut être aisément retiré et remplacé dans la plateforme de simulation. À terme, chaque module pourra alors être supprimé et le système multi-agent sera interfacé avec l'équivalent au sein de I2C.

5.3. Évaluation de l'apprentissage

5.3.1. Indicateurs pour l'évaluation

L'objectif principal de MAS4AT est de lever des alertes pertinentes à l'attention des opérateurs du système I2C. Pour ce faire, les agents-anomalies sont capables d'apprendre les valeurs des anomalies qu'ils représentent en accord avec les retours des opérateurs. À la lumière de cet objectif, nous avons choisi de mesurer l'apprentissage de MAS4AT en observant la convergence du système vers un état stable. Cette convergence, que l'on pourrait qualifier de globale, est le résultat de la convergence locale des agents-anomalies vers les valeurs qu'ils ont à apprendre. Un état stable du système est un état dans lequel l'opérateur n'envoie plus de retours négatifs à MAS4AT. Pour avoir une évaluation plus précise, nous allons nous baser sur le nombre de retours mais aussi sur l'écart entre la valeur apprise par chaque agent-paramètre et la valeur à apprendre. Nous allons pouvoir ainsi évaluer la vitesse d'ajustement et la justesse de l'apprentissage effectué.

Lors des tests que nous avons effectués, nous avons spécifié les bornes inférieure et supérieure de l'espace de recherche pour les valeurs des agents-paramètres. Ainsi les valeurs à apprendre par les agents-paramètres sont comprises entre 0 et 100.

5.3.2. Cas d'études

Deux principaux cas ont été étudiés. Dans un premier temps, nous avons testé MAS4AT dans un environnement contenant 1 anomalie et 10 navires. Ainsi, les scénarios construits pour chacun des dix navires ne contiennent qu'une seule anomalie pouvant se produire plusieurs fois. La figure 5 montre le résultat de l'apprentissage des agents-paramètres correspondant.

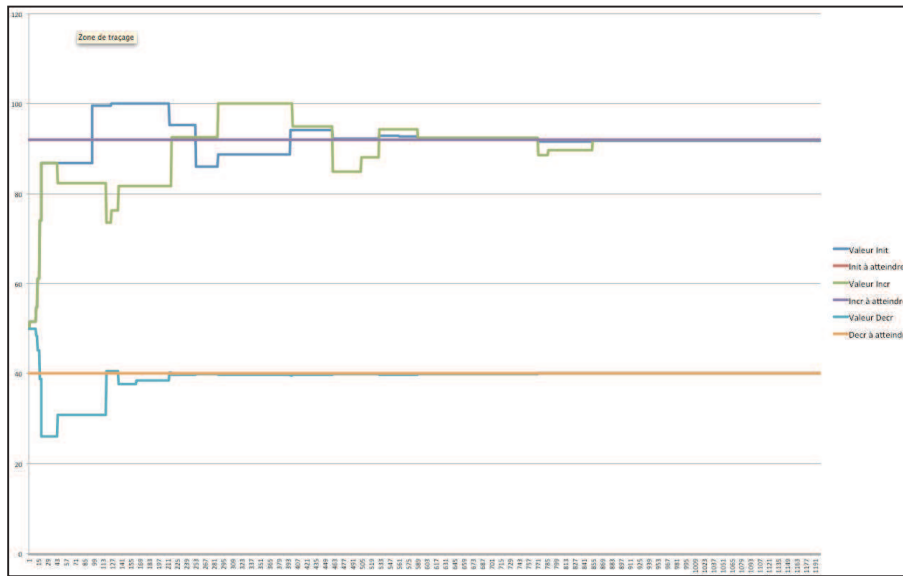


Figure 5. Convergence locale des agents-paramètres

Sur cette figure, nous pouvons apercevoir la convergence des valeurs apprises par chacun des paramètres vers la valeur réelle à apprendre. Cette première observation simple nous a permis de vérifier que les algorithmes d'ajustement permettaient de trouver la valeur souhaitée pour un agent-anomalie. En moyenne, sur une série de 50 jeux de tests, entre 20 et 40 retours de l'Opérateur sont nécessaires pour que l'ajustement des agents-paramètres mène aux valeurs correctes des agents-anomalies. Ces résultats sont intéressants car ils nous permettent de valider l'ajustement et nous pouvons nous assurer du fonctionnement de l'auto-ajustement des agents de MAS4AT.

Dans un second temps, nous avons évalué le système avec 10 anomalies et 10 agents. La figure 6 indique la convergence globale du système vers un état stable. Nous pouvons observer la stabilisation du système, c'est-à-dire que les valeurs des anomalies sont suffisamment proches des valeurs réelles.

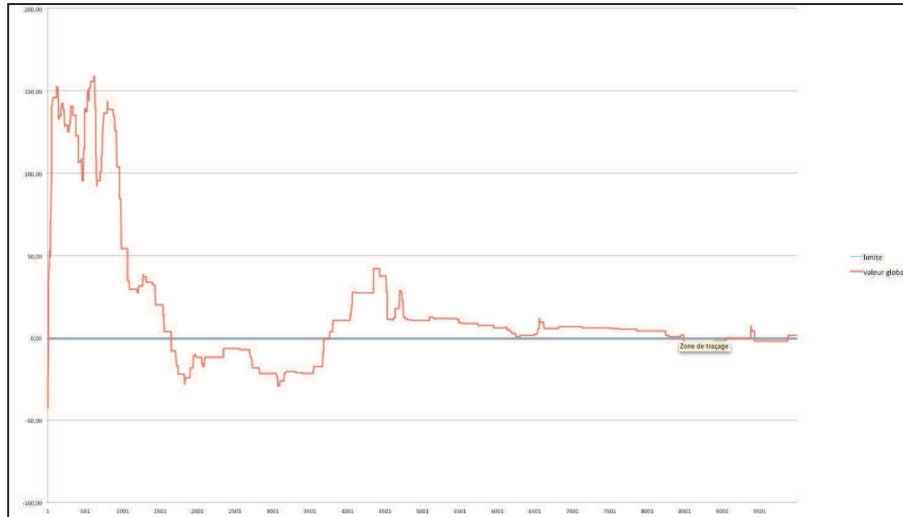


Figure 6. Convergence globale des agents-anomalies et agents-paramètres

Cet état se traduit par l'absence de retours négatifs en provenance des opérateurs. Dans le cas des simulations avec 10 navires et 10 anomalies différentes, le système converge vers un état stable avec entre 50 et 200 retours négatifs (sur un jeu de 50 tests). Le nombre de retours nécessaires est relativement élevé, mais cela s'explique par les algorithmes d'ajustement utilisés. En effet, ceux-ci doivent encore être optimisés pour prendre en compte des situations comme celle qui a eu lieu au pas de temps 4000. On peut voir sur la courbe de la figure 6 que les agents-anomalies étaient relativement proches des valeurs à trouver mais qu'un retour négatif a provoqué un ajustement qui a augmenté l'écart entre les valeurs estimées par les agents-paramètres et les valeurs à trouver, au lieu de le réduire. Cela s'explique par les valeurs initiales des paramètres, éloignées des valeurs cibles. Ainsi, les premiers retours reçus par les agents provoquent de grands pas d'ajustement. Ces résultats nous permettent cependant de valider le fonctionnement des algorithmes d'ajustement des agents de MAS4AT dans des situations où on trouve plusieurs navires et plusieurs anomalies. Il nous faut maintenant nous concentrer sur l'optimisation de ces algorithmes, notamment pour prendre en compte le début de l'apprentissage.

6. Conclusion

Cet article a présenté MAS4AT, un système multi-agent (SMA) pour le déclenchement d'alertes dans le cadre de la surveillance maritime. MAS4AT fonctionne en couplage avec un moteur de règles détectant les anomalies de comportement de navires évoluant dans une zone maritime sous surveillance.

MAS4AT calcule le comportement des navires en cumulant l'importance associée aux différentes anomalies qui leur sont associées et déclenche une alerte à destination d'opérateurs humains impliqués dans la surveillance maritime lorsque ce comportement est évalué comme menaçant. Les opérateurs valident ou invalident ces alertes qui sont documentées et transmises aux autorités pour intervention, ou bien retournées au SMA pour une phase d'apprentissage. Dans ce cas, le SMA ajuste par auto-adaptation les valeurs des anomalies afin de prendre en compte les retours des opérateurs humains. Ce système est conçu et développé dans le cadre du projet européen I2C.

Deux raisons principales nous font penser que l'approche adoptée dans I2C pour la surveillance maritime est pertinente. La première repose sur le couplage du moteur de règles et du SMA. Le moteur de règles permet de modéliser les comportements anormaux à travers un ensemble de règles représentant la législation maritime. Se limiter à utiliser un moteur de règles est cependant insuffisant car (i) il n'est pas possible de décrire toute la législation maritime à travers des règles, (ii) il n'est pas possible de faire une analogie avec anomalie et alerte, et (iii) tous les comportements anormaux ne sont pas connus ou modélisables sous forme de règles. Le SMA est introduit pour pallier ces insuffisances et notamment pour déclencher une alerte si le cumul des anomalies est supérieur à un seuil d'alerte donné, mais aussi pour découvrir des comportements collectifs anormaux alors que les comportements individuels qui les composent sont considérés comme normaux. La deuxième raison repose sur le type d'apprentissage utilisé dans I2C. En effet, cet apprentissage par renforcement qui tient compte des retours effectués par les opérateurs humains impliqués dans la surveillance maritime est reconnu comme étant adapté dans le cas d'un SMA (Tan, 1993).

Plus précisément, les contributions de l'article sont les suivantes :

- La spécification des capacités individuelles et comportements coopératifs des agents-navires, agents-anomalies et agents-paramètres de MAS4AT. Nous avons plus particulièrement mis l'accent sur les propriétés et les capacités spécifiques aux agents en leur permettant d'ajuster les valeurs des paramètres.

- Les algorithmes d'apprentissage de ces différents agents. Nous avons développé la méthode de décision des agents lors de la réception d'un feedback et explicité en quoi l'auto-ajustement des agents-paramètres permet d'apprendre les valeurs cibles.

- Une implémentation de MAS4AT couplée à un simulateur de scénario, d'anomalies et de retours effectués par des opérateurs humains. La plateforme SIM4AT a été développée pour pallier au manque de données réelles et son architecture est expliquée dans l'article, ainsi que ses interactions avec MAS4AT.

- Une première évaluation de l'apprentissage des agents de MAS4AT. Les premiers résultats obtenus lors de l'évaluation de l'apprentissage montrent que les algorithmes d'ajustement conçus et développés pour les différents agents permettent d'apprendre les valeurs des anomalies de manière satisfaisante, c'est-à-dire qu'ils

permettent la diminution de retours négatifs des opérateurs au cours de la simulation. Ces résultats nous permettent aussi de valider la modélisation et la méthode d'ajustement déterminée. En effet, les agents-paramètres sont capables de s'auto-ajuster en se basant uniquement sur les retours des opérateurs. Chaque action des différents agents aura un impact sur la valeur de comportement des agents-navires et provoquera un retour, qu'il soit négatif ou positif. Nous pouvons ainsi valider l'utilisation d'un apprentissage par renforcement en utilisant des SMA.

Nos travaux futurs vont dans un premier temps, c'est-à-dire à court terme porter sur l'optimisation des algorithmes d'ajustement pour réduire le nombre de retours nécessaires à l'apprentissage des valeurs des anomalies. Nous allons pour cela étudier la possibilité d'associer des poids différents aux anomalies. L'implantation actuelle ne considère en effet que des anomalies ayant le même poids, *i.e.* la même importance. Nous allons réfléchir à la pertinence de prendre en compte des anomalies d'importance différente et voir si ce nouveau critère améliore l'apprentissage dans MAS4AT. Nous allons également confronter l'apprentissage dans MAS4AT à des cas réels de surveillance maritime. Cela sera probablement possible dans le courant de l'année 2013 lorsque les différents capteurs fixes ou mobiles déployés dans I2C seront complètement opérationnels. Cela permettra de valider le passage à l'échelle de MAS4AT.

Enfin, nous allons à plus long terme essayer de voir dans quelle mesure le SMA pourrait faire des retours au moteurs de règles en lui proposant de nouvelles anomalies ou de nouvelles règles qu'il pourrait intégrer.

Bibliographie

- Andrade M.A., Chacon P., Merelo J.J., Moran F. (1993). Evaluation of Secondary Structure of Proteins from UV Circular Dichroism Spectra Using an Unsupervised Learning Neural Network. *Protein Engineering*, vol. 6, Oxford University Press, p. 383-390.
- Auslander B., Gupta K.M., Aha D.W. (2011). A comparative evaluation of anomaly detection algorithms for maritime video surveillance. *Society of Photo-Optical Instrumentation Engineers*, vol. 8019, p. 3.
- Barlow H.B. (1989). Unsupervised Learning. *Neural Computation*, vol. 1, n° 3, MIT-Press, p. 295-311.
- Capera D., Georgé J.P., Gleizes M.P., Glize P. (2003). The AMAS Theory for Complex Problem Solving based on Self-Organizing Cooperative Agents. *Enabling Technologies Infrastructure for Collaborative Enterprises*, WETICE, p. 383-388.
- Celik M., Dadaser-Celik F., Dokuz A.S. (2011). Anomaly Detection in Temperature Data Using DBSCAN Algorithm. *IEEE Innovations in Intelligent Systems and Applications (INISTA)*, p.91-95.
- Cohen C.J., Morelli F., Scott K.A. (2008). A Surveillance System for the Recognition of Intent Within Individuals and Crowds. *Technologies for Homeland Security, 2008 IEEE Conference on*, IEEE, p.559-565.

- Coulom R. (2002). *Reinforcement Learning using Neural Networks, with Applications to Motor Control*. Thèse de Doctorat, Institut National Polytechnique de Grenoble.
- Cram D. (2010). *Découverte interactive et complète de chroniques : application à la co-construction de connaissance à partir de traces*. Thèse de Doctorat en Informatique, Université Claude Bertrand Lyon 1.
- Direction des Constructions Navales, Systèmes et services et partenaires. *Description of Work*, Projet I2C, 2010.
- Dousson C. (1994). *Suivi d'évolution et reconnaissance de chroniques*. PhD thesis, Université Paul Sabatier Toulouse 3.
- Fok V., Lingard D.M. (2011). Using a Genetic Algorithm to Optimise Maritime Surveillance Performed by Space-based Sensors. *National Committee for Space Science & National Space Society of Australia*.
- García J., Molina J.M., Singh T., Crassidis J., Llinas J. (2011). Research Opportunities in Contextualized Fusion System. The Harbor Surveillance Case. *Advances in Computational Intelligence*, LNCS vol. 6692, p. 621-628.
- Georgé J.P., Gleizes M.P., Glize P. (2003). Conception de systèmes adaptatifs à fonctionnalité émergente : la théorie des AMAS. *Revue d'Intelligence Artificielle*, vol. 17, n° 4, p. 591-626.
- Gupta K.M., Aha D.W., Moore P. (2009). Case-based Collective Inference for Maritime Object Classification. *Eighth International Conference on Case Based Reasoning*, p.443-449.
- Gupta S., Hossain L. (2003). Towards Near Real-Time Detection of Insider Trading Behavior Through Social Networks. *Computer Fraud & Security*, vol. 1, p. 7-16.
- Hébrard P. (2011). Politique maritime : La surveillance maritime, enjeu stratégique pour l'UE. *Revue Maritime*, vol. 490, p. 68.
- Hinton G., Sejnowski T.J. (editors) (1997). *Unsupervised Learning : Foundation of Neural Computation*, MIT-Press.
- Jakob M., Vaněk O., Urban Š., Benda P., Pěchouček M. (2010). AgentC : Agent-based Testbed for Adversarial Modeling and Reasoning in maritime Domain. *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, vol. 1, p. 1641-1642.
- Juradsky D., Martin J.H., Kehler A., Vander-Linden K., Ward N. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, Prentice-Hall New Jersey,
- Kaelbling L.P., Littman M.L., Moore A.W. (1996). Reinforcement Learning: a Survey. *Journal of Artificial Intelligence Research*, vol. 4, p.237-285.
- Lauer M., Riedmiller M. (2000). An Algorithm for Distributed Reinforcement Learning in Cooperative Multi-Agent Systems. *Proceedings of the 17th International Conference on Machine Learning*, Citeseer.
- Lemouzy S. (2011). *Systèmes interactifs auto-adaptatifs par systèmes multi-agents auto-organisateur : applications à la personnalisation de l'accès à l'information*. Thèse de Doctorat, Université Paul Sabatier, Toulouse 3.

- Lemouzy S., Camps V., Glize P. (2011). Principles and Properties of a MAS Learning Algorithm: a Comparison with Standard Learning Algorithms Applied to Implicit Feedback Assessment. *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, CPS, p. 228-235.
- Li X., Xue Y., Chen Y., Mali B. (2011). Context-aware Anomaly Detection for electronic Medical Record Systems. *Proceedings of the 2nd USENIX Conference on Health Security and Privacy*, p. 812.
- Mano J.P., Georgé J.P., Gleizes M.P. (2010). Adaptive Multi-Agent System for Multi-Sensor Maritime Surveillance. *International Conference on Practical Applications of Agents and Multiagent Systems*.
- Nilsson M., van Laere J., Ziemke T., Edlund J. (2008). Extracting Rules from Expert Operators to Support Situation Awareness in Maritime Surveillance. *Proceedings of the 11th International Conference on Information Fusion*, IEEE.
- Pao Y. (1989). *Adaptive Pattern Recognition and Neural Networks*. Addison-Wesley Publishing Co.
- Sadri F. (2010). Logic-based approaches to intention recognition. *Handbook of Research on Ambient Intelligence: Trends and Perspectives*.
- Shi Y., Tian Y., Kou G., Peng Y., Li J. (2001). Network Intrusion Detection. *Optimization-based Data Mining: Theory and Applications*, p. 237-241.
- Shoham Y., Powers R., Grenager T. (2003). Multi-Agent Reinforcement Learning: a Critical Survey. *Unpublished Survey* <http://robotics.stanford.edu/shoham>.
- Sutton R.S. (1988). Learning to Predict by the Method of Temporal Differences. *Machine Learning*, vol. 3, p. 9-44.
- Sutton R.S., Barto A.G. (1998). *Reinforcement Learning: an Introduction*, The MIT-Press Cambridge.
- Tan K. (2005). *A Multi-Agent System for Tracking the Intent of Surface Contacts in Ports and Waterways*. Naval Postgraduate School Monterey CA.
- Tan M. (1993). Multi-agent Reinforcement Learning: Independent vs. Cooperative Agents. *Proceedings of the 10th International Conference on Machine Learning*, p. 330-337.
- Venturini G., Kodratoff Y. (1994). *Apprentissage adaptatif et apprentissage supervisé par algorithme génétique*. Thèse de Doctorat, Université de Paris 11.
- Watkins C.J.C.H., Dayan P. (1992). Q-Learning. *Machine Learning*, vol. 8, p. 279-292.
- Zander S., Nguyen T., Armitage G. (2005). Automated Traffic Classification and Application Identification Using Machine Learning. *The IEEE Conference on Local Computer Networks*, IEEE, p. 250-257.