Multiagent Coordination Techniques for Complex Environments: The Case of a Fleet of Combat Ships

Patrick Beaumont, Brahim Chaib-draa, *Senior Member, IEEE*

***Abstract*— The use of agent and multiagent techniques to assist humans in their daily routines has been increasing for many years, notably in Command and Control (C2) systems. In this context, we propose using multiagent planning and coordination techniques for resources management in real-time C2 systems. The particular problem we studied is the design of a decision-support for Anti-Air Warfare (AAW) on combat ships. In this paper, we refer to the specific case of several combat ships defending against incoming threats and where coordination of their respective resources is a complex problem of capital importance. Efficient coordination mechanisms between the different combat ships are then important to avoid redundancy in engagements and inefficient defence caused by the conflicting actions. To this end, we present four different coordination mechanisms based on task sharing. Three of these mechanisms are communication-based: central coordination, contract Net coordination and ∼Brown coordination, while the last one is a zone defence coordination and is based on conventions. Finally, we expose the results obtained while simulating these various mechanisms.**

***Index Terms*— Coordination mechanisms, agents, multiagent systems, command and control systems, military decision-making, Real time systems.**

I. INTRODUCTION

RECENTLY, we have seen the emergence of agent techniques as a new paradigm for software development. A simple way to describe agents is to say that they are entities perceiving their environment, and capable of acting rationally on it [1].

However, most environments are complex enough that no single agent can execute the entire task by itself. In this case, one can achieve such a task with the collaboration of individual agents, each having limited capabilities. This supports the recent research on collaborative agents (a subfield of Multiagent System (MAS) [2]), making possible distributed problem solving for some very complex distributed applications.

For sure, maritime environments are commonly known to be very complex where agent techniques can be helpful. In this specific context, modern AAW is an arduous problem, because of the sheer volume of data, usually imperfect, that needs be processed under time-critical conditions. In recent decades, operational crews were trained to operate within a force command structure where they can i) recognize a threat, ii) know how to react to that threat and, iii) employ measures to defeat that threat. Unfortunately, the problem has become increasingly intricate as threats and defence systems gained in sophistication. The scenarios are also gaining in complexity. This is mostly due to engagements occurring now more on the littoral rather than open sea and the ever-increasing need for cooperation among ships with different resources.

In the case of an aerial attack on a combat ship, which is an example of AAW, the operators in this ship have little time to observe, orient, decide and act. There is often less than a minute between detection of a threat and its impact on the ship. This calls for very fast decisions to be made by considering several important factors to make sure the best possible plan is carried out. Failing to do so might mean destruction of the ship and death of its crew. Because there is not much time to consider a great number of plans, the operators might overlook the most advantageous prospects, resulting in the choice of a suboptimal plan. Moreover, under such real-time constraints, the commander might make errors due simply to the complexity of the environment or the stress that such a situation generates, which will obviously result in dire consequences.

In a fleet of combat ships, some resources have more restrictions than usual, whilst there are synergic interactions that could potentially increase the overall survival of the fleet; the whole is greater than the sum of its parts. Of course, deciding on a course of action, communicating it to allies and adjusting the initial plan to the schedule still has to be done under time constraints, as stated earlier. This situation poses significant challenges to future shipboard C2 systems and the operators using these systems to defend the ship. Considering the complexity of the problem and the fact that computers can operate and communicate tremendously faster than humans, it appears that a reliable onboard Decision-Support System (DSS) (i.e., a software agent used to aid human operators in their decision-making) would really be welcome in modern AAW.

This led us to direct our attention to the particular problem of designing a DSS for AAW on combat ships. Since we address here more specifically, a fleet of combat ships with a DSS in each ship, we should then see the fleet as a multiagent system where the different DSS try to obtain and maintain the best coordination between them. The study of the fleet as a multiagent system was initiated through a project called Naval Environment for Resource Engagement in Unpredictable Situations (NEREUS),

NEREUS project is a collaborative project between the DAMAS laboratory[1], and Lockheed Martin Canada (LMC)[2]. The goal of this project was to efficiently manage all the

---

[1] For more information, see: `www.damas.ift.ulaval.ca`.

[2] For more information, see: `www.lockheedmartin.com/canada/`

resources (weapons, radars, electronic systems, etc.) present in a fleet to increase survival chances of combat ships at the time of attack by Anti-Ship Missile (ASM)s. Because resource managing for a fleet is a complex distributed problem, we rely on a collaboration between the onboard DSS agents (which form a multiagent system) to manage those resources while increasing the survivability of such a fleet [3]. One should first note that since the combat ships must respond in "real-time", they cannot adopt a classical complete plan coordination method which consists of : i) detecting, identifying and prioritizing threats; ii) creating individual plans; iii) resolving conflicts and managing positive and negative interactions; iv) effecting their respective portions of the joint plan. They should instead follow the *threat distribution coordination* (usually referred to as "task sharing" in the literature) where ships negotiate the allocation of threats and each ship only plan on threats it has been allocated.

Furthermore, coordination mechanisms are usually separated into three different types : (i) those based on communication where agents rely on exchange between them in order to manage the activities' interdependencies between them; (ii) those based on conventions where agents rely on some pre-compiled social laws given by designers in order to manage the activities' interdependencies between them; (iii) those based on learning where agents learn by trials-errors how to coordinate themselves.

In this paper, we present the coordination mechanisms that we have developed and experimented with in the context of a fleet of combat ships: Three of these mechanisms are communication-based: central coordination, contract Net coordination and ∼Brown coordination, while the last one is a zone defence coordination and is based on conventions. It is important to compare these mechanisms between them and to know in which situations such or such mechanism is the most efficient in terms of survival rate and resource utilization.

The paper is organized as follows. Section II introduces the command and control processes sustaining the NEREUS project as well as the the resources which are available for a combat ship. Section III specifies the way that we have simulated the combat ship fleet. Section IV presents in detail the different coordination mechanisms used to manage resources between the different ships, particularly when they are faced with attacks. Section V compares the different coordination mechanisms by listing and discussing their relative strengths and weaknesses. Finally, Section VI concludes and presents the main issues for future work.

II. THE COMBAT SHIP SURVIVABILITY AS COMMAND-CONTROL PROCESS

Before explaining how we have simulated the combat ship fleet, it is important to introduce the Command and Control (C2) processes which sustain the Anti-Air Warfare (AAW) and the resources available for a combat ship in the context of such AAW.

A. Command and Control

Command and Control (C2) is the exercise of authority and direction by a properly designated commander over assigned and attached forces in the accomplishment of a mission [4]. C2 functions are performed through an arrangement of personnel, equipment, communications, facilities, and procedures employed by a commander in planning, directing, coordinating, and controlling forces and operations in the accomplishment of a mission. C2 tasks usually include weapon and sensor systems control, tactical picture, compilation, situation interpretation and threat evaluation, weapon selection, engagement monitoring and mission planning and evaluation. In fact, the C2 tasks cover what is called the Observe, Orient, Decide, Act (OODA) loop [5], [6]. This theory, put forth by Colonel John R. Boyd, essentially states that, in a confrontation, whoever is quicker to react to changes will prevail. Boyd describes this readiness to react to changes as "having a tight OODA loop". Therefore, Boyd's theory can be expressed as "whoever has a tighter OODA loop, will prevail".

The development of a relevant C2 theory will have significant impact upon the analysis and design of both military and civil C2 systems. The major considerations are the following [4]:

- *C2 has a functional architecture:* Another key element of the C2 process is its functional decomposition. Indeed, the C2 process can be decomposed into a set of generally accepted C2 functions (see below) that must be executed in a reasonable time frame to ensure success.
- *C2 is a complex process:* The complexity of most C2 problems arises from the multitude, the heterogeneity and the interrelations of the resources involved.
- *C2 deals with large volumes of data under stringent time constraints:* Perceptual and cognitive processing is further complicated by the fact that the underlying information is derived by continuously integrating and merging data from a variety of sources to build a coherent situational picture.

In the case of AAW, the list of functions of the C2 architecture is as follows:

i) *Threat detection*: Based on data from several sensors.
ii) *Target tracking*: Usually based on data fusion.
iii) *Discrimination*: Results in the resolution of true threats from decoys.
iv) *Identification*: In this step, the threats are identified.
v) *Battle planning*: In this process, decisions are made on how to deal with the identified threats.
vi) *Resource assignments*: Resources are assigned to engage each threat.
vii) *Engagement control*: The process by which decisions in the two preceding steps are executed in real-time.
viii) *Damage assessment*: This process evaluates the outcome of the engagement control.

In our NEREUS project, we focused specifically on some particular aspects of the C2, in order to reduce the complexity of the domain. Our primary centers of interest, which have been discussed by many authors, are: *battle planning* [7]–[9], *resources assignments* [10]–[12], and *engagement control* processes [4], [13], particularly for the survivability of combat ships and specially when they are faced with attacks as a fleet.

B. Resources of a Combat Ship

A combat ship, can be a frigate or any other war vessel. In this paper however, we focus on the frigate case and consequently we will use the two words "combat ship" and "frigate" interchangeably. A frigate has two types of weapons: The AAW hardkill and the AAW softkill. The first are weapons that are directed to intercept a threat and actively destroy it through direct impact or explosive detonation in the proximity of the threat. The range of different types of hardkill weapons varies, and the effectiveness of these weapons depends on a variety of factors, like distance to the threat, type of threat, speed of the threat, environment, etc. The AAW hardkill weapons for a typical Frigate include surface-to air missiles (SAMs) that have the greatest range, an intermediate range Gun, and a Close-In Weapons System (CIWS) that is a short-range, rapid-fire gun. Closely allied to these weapons are two Separate Tracking and Illuminating Radars (STIRs) that are used to guide a SAM to a threat, and to point the Gun. This effectively provides two concurrent fire channels for the AAW hardkill weapons. The CIWS has its own pointing radar.

The AAW softkill weapons use techniques to deceive or disorient a threat to cause the threat to destroy itself, or at least lose its fix on its intended victim. Again, the range and effectiveness of these weapons varies considerably. The AAW softkill weapons for a typical Frigate include chaff and jamming systems. The chaff system launches a shell that produces a burst at a designated position. The resultant chaff cloud has a significant radar cross-section that can be used to screen the Frigate or produce an alternate target on which a radar-guided threat can fix. The jamming system uses electromagnetic emissions to confuse the threat's sensors to cause the threat either to lose its fix on its intended target, or to improperly assess the position of its target.

III. SIMULATION OF A FLEET OF COMBATS SHIPS

We have developed a simulator as depicted in Fig. 1 where each combat ship is considered as a sophisticated, "autonomous" agent where the DSS agent is just a part devoted to the decision making process. In fact, autonomy in our case is limited because our agents i) are members of a team (i.e., fleet), ii) are expected to be fully cooperative and iii) have to respect military doctrines and rules of engagement. Thus, all agents need to coordinate themselves to achieve an acceptable solution.

Since we work in the context where each combat ship is an "autonomous" agent, the fleet is then a multiagent system (MAS). Therefore, it would be useful to describe this MAS environment. In Table I, adapted from [14], we present the specific characteristics of the multiagent system environment in project NEREUS.

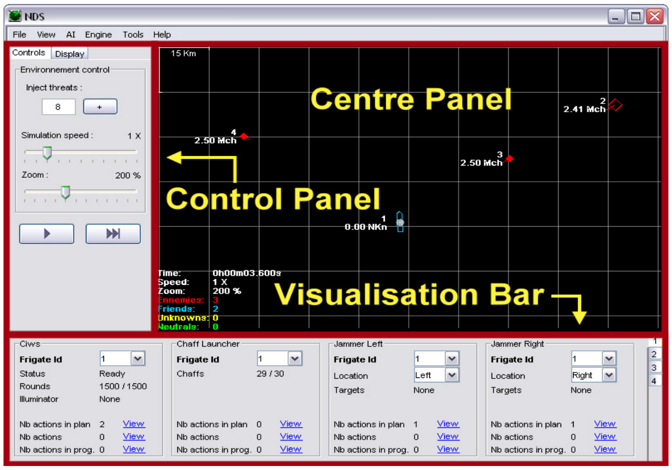
Agents

Note that the number of ships in a typical task group (or fleet) is four frigates and an important unit (typically a cargo vessel in our case). The frigates are responsible for their own defence and the defence of the cargo vessel, which has no defence systems. Considering only the frigate agents, they are clearly uniform. However, we plan to include diversity in the available resources in the future of the project. This would allow simulating ships from different classes and nationalities. The goal of each frigate is to maximize the survivability of the fleet as a whole. To simulate this attitude, we varied the relative importance of the escorted vessel; when there is a vessel of greater importance, the frigates will try to defend it, even to the detriment of their own survival. Finally, the abilities of the agents are diversified and relatively advanced; they can use all onboard systems to create a great number of different solutions.

TABLE I

CHARACTERISTICS OF A MULTIAGENT SYSTEM.

	Attribute	*Range*
Agents	Number Uniformity Goals Architecture Abilities	Usually 4 Homogeneous Complementary Mostly deliberative Somewhat advanced
Interactions	Frequency Persistence Level Pattern Variability Purpose	Low to Medium Middle-term Small but meaningful Decentralized or Hierarchical Changeable Cooperative
Environment	Predictability Accessibility Dynamics Diversity Resource availability	Stochastic Slightly limited Fixed for a scenario Limited Restricted



Simulation speed : 1 X
Zoom : 200 %

Time: 0h00m03.600s
Speed: 1 X
Zoom: 200 %
Ennemies: 3
Friends: 2
Unknowns: 0
Neutrals: 0

Ciws		Chaff Launcher		Jammer Left		Jammer Right	
Frigate Id	1	**Frigate Id**	1	**Frigate Id**	1	**Frigate Id**	1
Status	Ready	Chaffs	29 / 30	Location	Left	Location	Right
Rounds	1500 / 1500			Targets	None	Targets	None
Illuminator	None						
Nb actions in plan	2	Nb actions in plan	0	Nb actions in plan	1	Nb actions in plan	1
Nb actions	0	Nb actions	0	Nb actions	0	Nb actions	0
Nb actions in prog.	0	Nb actions in prog.	0	Nb actions in prog.	0	Nb actions in prog.	0

Fig. 1. A view of our simulator

Furthermore, each agent can be described in terms of its Performance, Environment, Actuators, Sensors (PEAS), as suggested by Russell and Norvig [1]. Table II presents these aspects.

Note that the resources mentioned in this table are those available on a typical frigate, and have been described succinctly in the previous section and in more detail in [15], [16]. Considering the attributes of environments presented in [1], we can describe the environment more profoundly. Since the agent has only a partial view (its radar range), we can deem the environment partially observable mainly because the agent lacks important information pertaining to the choice of actions:

TABLE II

PEAS OF A COMBAT SHIP AGENT.

	NEREUS Agent
Performance	The performance is evaluated by two factors: 1) the survivability of the ship in the fleet and 2) the utilization of resources. However the survivability is the most important performance measure of these two factors.
Actions	Available actions comprise: the use of every onboard resource (including hardkill and softkill weapon systems), and the use of steering and propulsion systems.
Environment	The environment is an unpredictable, naval environment. It might be situated on the coast or at sea.
Sensors	Everything detected by radars and sonars, including friends and foes, airplanes, ships, missiles, chaff clouds, etc. The status of every onboard weapons systems is also known.

it does not know *when* a detected airplane will launch its ASM. In addition, the environment is stochastic, since weapons have a *kill probability*. Finally, the environment is also sequential, dynamic and continuous, since a simulation is a scenario that stretches along a continuous timeline, with threats appearing at undeterminate times.

Interactions

The interactions in our simulator happen in the form of asynchronous messages. Depending on the mechanisms used, there can be no message exchanged (such as in *Zone defence* with no need for backup) up to a number of messages in the order of the number of threats (as in Contract Net)[3]. The effects of these messages persist a while since they are used in the creation plan. The messages transmitted are reduced to the strict minimum, but they require knowledge to be constructed before passing and analyzed when received. We implemented both centralized and decentralized organizations. Whether an organization is centralized or not has an important impact on agent interactions.

Environment

We have said earlier that our environment is stochastic. In our case, this means that the environment is partially foreseeable, as we can determine the probability that specific events will occur. Furthermore, since we focus on resource management (RM), we do not consider the situation and threat assessment (STA) (see the previous section II-A for more details) for a fleet and the inter-agent coordination required to obtain a global view of the system. Thus, we consider that every agent has the same *combined* view. However, the environment is still partially globally observable, meaning that even when the different sensory inputs from agents are put together, the view is not necessarily completely observable. In our simulator, the dynamics (such as the communication environment, etc.) are fixed for any specific situation, but can be changed over the course of many scenarios. Furthermore, the diversity is intentionally limited, to focus on specific important aspects. Finally, the resources available to any agent are limited by the number and specification of weapon systems. Usually, the most constraining aspect in our simulations is the available time, as we rarely run out of any physical resource.

Knowing all these aspects, every single agent has to choose *when* to coordinate and *what* to do. This particular problem is complex, as has been expressed by many authors [17]–[21].

IV. COORDINATION MECHANISMS

A. Preliminaries

Firstly, it is important to define precisely the coordination process in the case of a multiagent environment. For us: *Coordination is the process by which the inter-dependencies between the agents' activities are managed* [22].

Coordination mechanisms are usually separated into three different types [23], as shown in Fig. 2: those based on communication, on conventions (or social laws) and on learning. Mechanisms based on learning can use communications, conventions or both. However, such mechanisms are distinguished by the fact that the strategies used are learned by the agents over the situations they encounter rather than decided at design time as in the two other classes of coordination mechanisms.

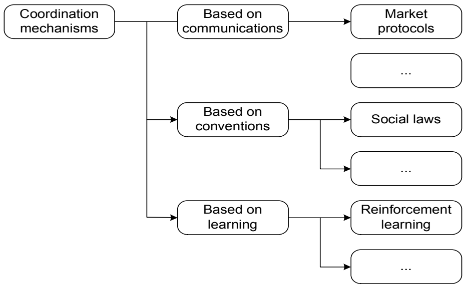


Fig. 2. Taxonomy of coordination mechanisms (after Boutilier [23]).

In most systems, coordination must be carried out in an environment constrained on time, available bandwidth, etc. In the case of AAW, the environment imposes strong real-time constraints. In a standard AAW situation, operators have few seconds available in which they must identify threats, choose and apply defence plans. In addition to these constraints, a fleet requires coordination between its members. As the reaction time is usually very short, it is often not possible for the ship operators members of this fleet to coordinate their actions with each other. This can result in i) redundancy in the engagements, using more resources than necessary, ii) inefficient defence, and iii) an increase in the cost of the global defence solution. Another impact of the lack of coordination is the negative interactions that can take place when certain resources are used in parallel, thus creating a degradation of the global solution. This prompts the need for an increasing cooperation between frigates. Indeed, good coordination mechanisms for the optimal use of the resources of a group of frigates become essential during a military deployment.

---

[3] These mechanisms are presented in Sections IV-C.1 to IV-C.4

Thus, in NEREUS, the coordination is cooperative and is intended to organize the individual actions toward a common goal, which is the efficient defence of the complete fleet.

B. The Threat Coordination-Allocation Method

In the light of those elements, the first step to successfully coordinate agents consists of developing efficient coordination mechanisms[4]. A first approach can be considered: the *complete plan coordination* method, which consists of:

i) Detecting, identifying and prioritizing threats.
ii) Creating individual plans.
iii) Ruling out conflicts and managing positive and negative interactions between frigates' plans.
iv) Effectuating the determined plan.

However, this coordination method has a problem: it takes much time to rule out the conflicts emerging from the individual planning, since many interactions are required to do so. This aspect of time is very detrimental since the reaction time must be short in most situations.

To resolve this problem, we consider a second approach: Interleaving allocation and coordination of a set of tasks among agents before starting the planning process. We call this approach the *threat coordination-allocation method*. In this method, each threat will be allocated to a specific agent, then each agent will only plan on threats it has been allocated. Such a method can be decomposed into:

i) Detecting, identifying and prioritizing threats.
ii) Coordinating and allocating threats among frigates, using one approach among the following:
  - *Zone Defence coordination.*
  - *Centralized coordination.*
  - *Contract Net coordination.*
  - *∼Brown coordination.*
iii) Creating individual plans between frigates' plans.
iv) Managing positive and negative interactions.
v) Effectuating the determined plan.

The *coordination-allocation method* alleviates the burden of the agents, as it significantly reduces the complexity of the planning process. Indeed, since the threats are assumed independent, it is easier to plan for a few threats than for the complete list of threats. Another advantage of this approach is that we eliminate conflicting actions where two agents engage the same threat while another is left unimpeded. Furthermore, reducing the conflicts greatly accelerates the coordination process since many fewer messages will be exchanged. Indeed, since we are not in a system with free communications, it takes several milliseconds for a message to reach its recipient.

In the case of the *complete plan coordination*, the coordination happens at Step IV-B, *after* the creation of individual plans, while in the *coordination-allocation method*, coordination occurs at Step IV-B, *before* the creation of individual plans. We developed coordination mechanisms for the *coordination-allocation method*, inspired by approaches commonly used in task sharing problems (market mechanisms, Contract Net, multiagent planning and organizational structure, etc.).

However, before going on, we need to define what the probability of success (P_S) is, as it will be used in the presented coordination mechanisms. The P_S of a frigate for a threat is a value that represents the probability, evaluated by the frigate, to destroy this threat. A P_S list (LP_S) for a specific frigate contains its P_S evaluation for each threat, while a P_S matrix (MP_S) contains the LP_S of every frigate.

Obviously, it is impossible to compute the exact P_S of a frigate against a threat only *before* planning. Thus, it is imperative that we use heuristics to determine probabilities of success.

The P_S evaluation heuristic implemented is inspired by [24] and is based on two factors: the *probability of kill* (P_K) and the closest point of approach (CPA). The P_K is relative to the prior number of threat engagements contracted, while the CPA is the shortest possible distance between the trajectory described by a threat and the evaluation point. The P_S evaluation is further detailed in [16].

C. Four Mechanisms for the Threat Coordination-Allocation Method

1) Zone Defence Coordination: This mechanism is a type of *convention-based* coordination, where the notion of role [25] of an agent is a key concept. Each role has associated responsibilities and preferences, which can be defined at design time or dynamically. The zone defence mechanism defines an agent role as the defence of a particular azimuthal sector around the fleet. These sectors are determined in the following way:

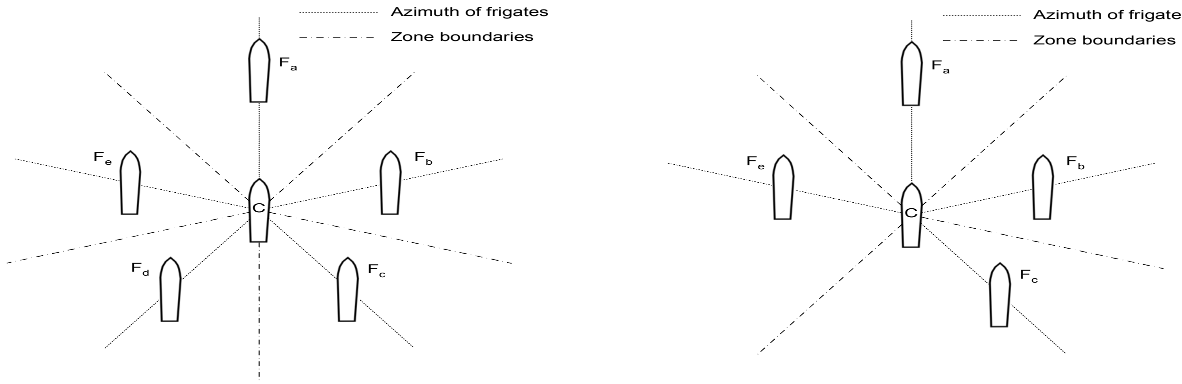
i) Determine the *center* of the fleet (in our case, it is the protected unit (a cargo vessel) that is situated in the center of the fleet).
ii) Determine the azimuth of every ship from the center unit.
iii) Determine the boundaries, which divide exactly in two each pair of neighboring ship azimuths.

When the boundaries are determined, each agent-ship knows that it is responsible for engaging any threat detected in the zone delimited by its boundaries, which are determined and maintained dynamically. Thus, if a ship is destroyed, the boundaries are resolved again to determine the new zones of defence. Fig. 3(a) illustrates a simple formation for five frigates and a cargo vessel (noted C in this figure); Fig. 3(b) represents the sector redivision if frigate F_d is destroyed.

Another important concept of the *zone defence* is the defence threshold. If an agent, estimating a P_S, realizes that this P_S is below the threshold, it will seek assistance in the engagement of this particular threat. At first, it demands help from another agent closest to the threat. Then, if the latest agent does not reply or refuses to engage the threat, the agent will seek assistance with its other neighbor. In the case of another refusal, it will add the threat to its plan, even though its estimated P_S value for this threat is below the threshold.

---

[4] Note that in the approaches we present, even though the steps seem to be sequential, there can be a certain amount of concurrency between them. For example, carrying out step 4 may introduce new interactions that would need to be dealt with (step 3).



(a) Initial zone division.

(b) Dynamic zone redivision.

Fig. 3. Zone defence responsibility division.

Since we developed completely cooperative agents, an agent will refuse to help its neighbor if and only if its personal P_S evaluation for this threat is worse than the P_S evaluation of the asking agent.

Fig. 4 presents the communication protocol for the *zone defence* mechanism. As with every other protocol presented in this chapter, we use the formalism of Agent UML (AUML) introduced by [26].Notice that we refer to a “Neighbour” of agent X, that is, to any agent who is located near to X.

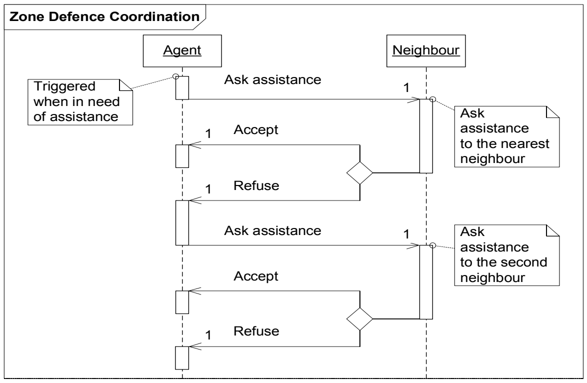


Fig. 4. Zone defence protocol.

We should note that zone defence has an advantage over the other mechanisms we will present later on: it uses almost no communications. Therefore, in the case where the communication channels have a very low output or are completely disabled, this mechanism stands out from the others.

2) Central Coordination: The *central coordination* mechanism is based on communications, with a centralized coordinator. The concept of the central coordination is that a central agent is responsible collecting the information, and deciding a task distribution according to this information. In this case, the information collected is the LP_S of every agent.

The central coordination process can be described as follows:

i) The fleet chooses a coordinator.
ii) When one or more threats are detected, every agent-ship computes its LP_S and sends it to the coordinator.
iii) The central coordinator constructs a capability matrix (which is simply the MP_S).
iv) The coordinator decides how to assign the incoming threats to frigates, using the capability matrix and an optimization algorithm.
v) The coordinator sends notification messages to chosen agent-ships.

Note that the choice of an appropriate coordinator is done prior to detecting any threat. In our case, the coordinator is the frigate with the highest ranking. The *rank* is a simple value of the relative importance of each ship. The process of choosing the coordinator is facilitated by the assumption that every agent has the same common view, as we explained in Section IV. This, combined with the fact that our agents are fully cooperative, makes it possible for any agent to determine which one is the coordinator without the need for a complex negotiation.

The assignation of tasks by the coordinator can be done with any optimization algorithm such as a greedy algorithm or even a complete lookup of the solution set. The greedy algorithm is a fast heuristic, while the complete solution lookup takes more time, since it looks over every possible solution to choose the best one. Usually, the central coordinator allocates only one threat per frigate. However, if there are more threats than frigates, the coordinator first allocates one threat per frigate as usual and then starts the process over by demanding another P_S evaluation for the remaining threats. A second option, not implemented yet, would be to allocate more than one threat per frigate, using a heuristic to predict the degradation of P_S.

A disadvantage of the central coordination mechanism is that it is a centralized mechanism. This can cause problems, since a failure of the coordinator can make the whole coordination process abort and put the fleet in peril. Indeed, if the coordinator becomes unresponsive for any reason (damaged communication system, the ship is sunk, etc.), the coordination process must be started over with a new coordinator, at the expense of several important seconds. Of course, the use of backup coordinators, which would receive the same information as the central coordinator, can alleviate this problem, but at the expense of other problems: overload of communication channels, security, etc.

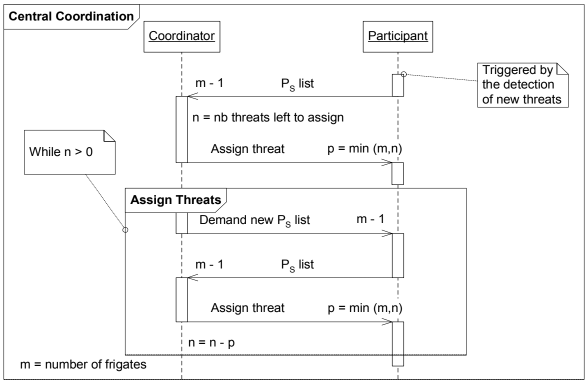


Fig. 5. Central coordination protocol.

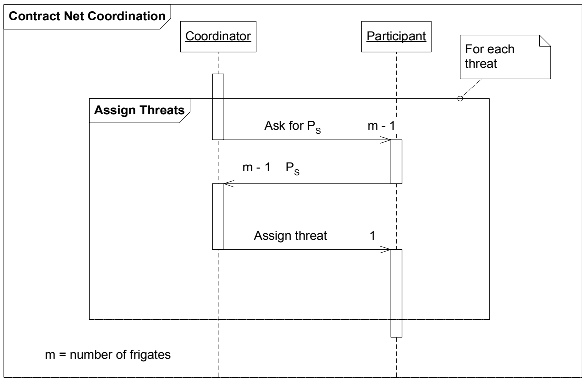


Fig. 6. Contract Net protocol for the NEREUS problem.

Fig. 5 presents the *central coordination* mechanism protocol formalized with AUML. In this figure, we see the messages exchanged between the coordinator and the other participating agents. We also see the iterative process (Assign Threats) which we described earlier. In this process, the coordinator asks the participants to send their LP_S. When it has received every P_S list, it then assigns the threats to the agents. The process is iterated as long as there are threats left unassigned.

3) Contract Net: The Contract Net mechanism, introduced by Smith [27] and standardized by FIPA [28], is similar to the central coordination mechanism, as it relies on a central coordinator and the use of communications. The difference between the two mechanisms is the number of threats assigned at one time. In the central coordination, we want to assign all threats at once, while we will allocate one threat at a time in the Contract Net mechanism. The following describes the Contract Net process:

i) The fleet chooses a coordinator.
ii) For each threat detected, one at a time:
    a) The coordinator asks every ship to send an estimated P_S value for this threat.
    b) Each ship returns its estimated P_S value for this specific threat, considering already assigned threats.
    c) The central coordinator chooses the best frigate to engage this threat and informs the agent.

In Fig. 6, which illustrates our adaptation of Contract Net to NEREUS, there are three types of messages exchanged between the coordinator and the other participating agents, corresponding to the three steps previously presented and noted a), b) and c). It is also interesting to compare the iterative process in this protocol to the one in the central coordination. We find that the only difference is the number of threats we allocate at a time.

The drawbacks associated to the central coordination mechanism also apply to the Contract Net protocol, since both mechanisms are centralized coordination methods. However, the communication channels are more solicited in the Contract Net protocol than in the central coordination protocol, since more messages are exchanged. In this case, if the communications are unsafe or unstable in Contract Net, the chances increase that they become elements of failure. Therefore, the central coordination will give more "reactive" (i.e., faster) responses, while the Contract Net approach will give better results but will also take longer. In addition, as the communication bandwidth decreases, the expected quality and timeliness of the coordinated solution obtained with Contract Net will decrease faster than with the central coordination method. Therefore, Contract Net is applicable only when there is sufficient time to coordinate. A more detailed comparison of these two mechanisms is provided in Section V.

4) ∼Brown (Similarly Brown): Another mechanism based on communications is the mechanism proposed by Brown [24], [29].

This mechanism closely resembles the central coordination with added parameters. The main difference between the central coordination and the *∼Brown* mechanism is that the threats are ordered by a priority evaluation before being distributed. This priority is based on three factors: the certainty that a threat is aimed at a specific ship, the relative importance of each ship and the fleet engagement capability for each threat.

Fig. 7 illustrates the ∼Brown coordination protocol. This figure illustrates the broadcasts between agents. As in the central coordination, we see that the *Assign Threats* process is iterated as long as there are still unassigned threats.

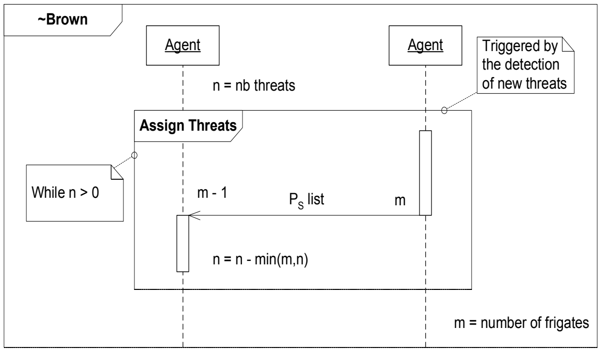


Fig. 7. ∼Brown coordination protocol.

∼Brown can be used in a centralized or decentralized fashion. However, to use ∼Brown mechanism in a decentralized way, the following assumptions must hold: i) the agents must be entirely cooperative, ii) the agents must be homogeneous, iii) the protocol must be used in the same way for every agent and iv) the agents must be aware that the three preceding assumptions hold. These assumptions are required to make sure that every agent, receiving the same information, will evaluate the situation in the same way. Firstly, if the agents are not entirely cooperative (e.g., if they are from different nationalities), there is the possibility that one agent might defect, which is unacceptable in our case. Furthermore, if the agents are not homogeneous, or if any part of the protocol uses information specific to a ship (as ship's own ranking), the allocation might be evaluated differently among different ships. The fourth assumption is self-explanatory.

Thus, if these assumptions hold, a simple way to use the $\sim$Brown mechanism in a decentralized fashion is to broadcast all the information to every agent. Thus, since each agent receives the same information and reasons the same way, it can deliberate and come up with an allocation solution.This solution does not need to be sent to other agents since each one is supposed to find the same solution. Therefore, once a solution is obtained, an agent only needs to act on its assigned threats, as it is sure that the other agents will take care of their own threats.

The following process describes the steps of the decentralized version of $\sim$Brown mechanism:

i) Before any threat is detected, each agent determines the relative weight of each ship and puts in a list of weights (*W*). The generic formula to compute the weights is:

$$weight = rank \cdot x + y$$

where we need to find *x* and *y*. The *rank*, as explained earlier, is a simple value of the relative importance of each ship. Thus, a ship with a ranking of 10 is more important than a ship with a ranking of 5. Since any agent knows the ranking of every ship, it also knows what the lowest and highest ranks are. In addition, an adjustable parameter is known: the *maximum weight deviation* (Dev_{max}), which is the desired difference between the highest and lowest ranking frigates' weights. Knowing this, we have:

$$1 - Dev_{max} = lowest \cdot x + y$$

$$1 = highest \cdot x + y$$

Thus,

$$x = \frac{Dev_{max}}{highest - lowest}$$

$$y = 1 - \frac{highest \cdot Dev_{max}}{highest - lowest}$$

ii) When threats are detected, a matrix of threats' targeting probability (*T*) is created, representing our confidence that a threat is aimed toward a specific ship.

iii) Each agent determines its P_S list and broadcasts it to the other agents.

iv) A threat-weight matrix ($T \cdot W$) is calculated, which is a multiplication of the weight list with the targeting matrix.

v) Once each LP_S is received, the fleet engagement capability (P_F) for each threat is determined. The fleet engagement capability (P_F) for a threat is computed by multiplying every P_S for this threat.

vi) Each agent computes the fleet engagement priority matrix, which is $\frac{T \cdot W}{P_F}$, for each threat.

vii) Each agent constructs a capability matrix which is the multiplication of the matrix composed of the LP_S of every frigate by the *fleet engagement priority* matrix: $(T \cdot W/P_F) \cdot MP_S$. Thus, this matrix represents how the threats should be engaged. It considers the relative weight of each ship, the possible targets of each threat and the different capabilities of each frigate against the incoming threats.

viii) Using an allocation algorithm, as in the central coordination mechanism, each agent determines the assignment of threats to ships.

A simple example shown in Fig. 8 and 9 illustrates the calculations required to determine the fleet engagement priority for this specific example. Looking at the weights list, we see that the Dev_{max} is 0.5. In this example, uncertainties in the *T* matrix are also showed. Once the MP_S is constructed (with every LP_S received), the P_F list is computed for each threat. Then, we use the results of *T*, *W*, P_F and MP_S to compute the final capability matrix. In this particular example, the complete lookup (i.e., best answer) would give the following assignments: (F_A-Th_1, F_B-Th_2, F_C-Th_3).

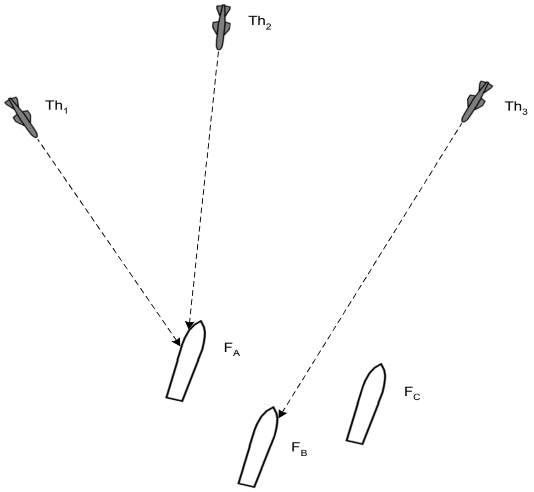


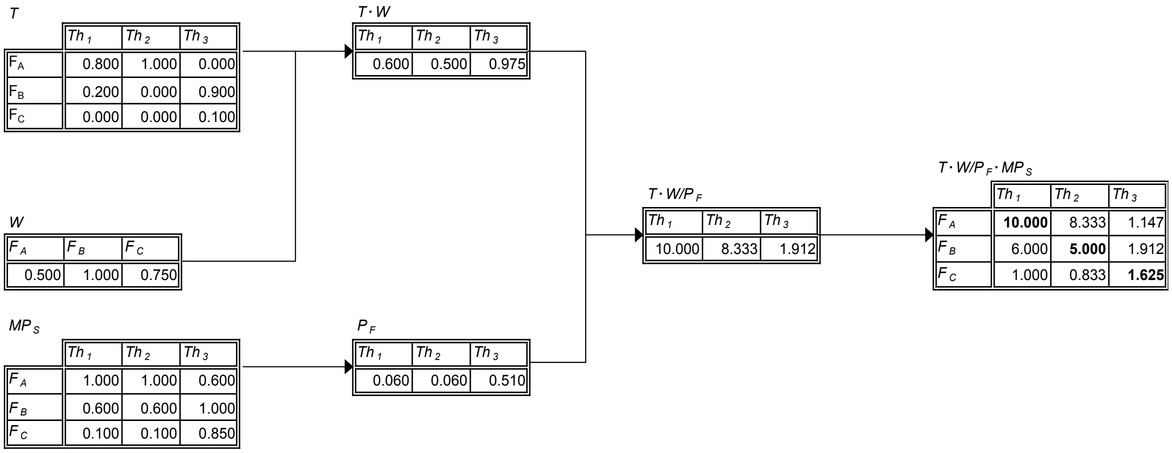
Fig. 8. A simple AAW scenario.

V. PERFORMANCES

A. Analysis of Protocols in Terms of Solution Quality

We now analyze the different protocols in terms of the following attributes:

- *Type of coordination method:* As previously, the coordination mechanisms can be separated into three types: communication-based mechanisms, convention-based mechanisms and learning-based mechanisms. Knowing the type of each one is useful to determine which factors are more important to this mechanism.
- *Communications:* As has been discussed earlier, the communications are important in many coordination mechanisms. Thus, knowing the number and importance of



T

	Th_1	Th_2	Th_3
F_A	0.800	1.000	0.000
F_B	0.200	0.000	0.900
F_C	0.000	0.000	0.100

W

F_A	F_B	F_C
0.500	1.000	0.750

MP_S

	Th_1	Th_2	Th_3
F_A	1.000	1.000	0.600
F_B	0.600	0.600	1.000
F_C	0.100	0.100	0.850

$T \cdot W$

Th_1	Th_2	Th_3
0.600	0.500	0.975

P_F

Th_1	Th_2	Th_3
0.060	0.060	0.510

$T \cdot W/P_F$

Th_1	Th_2	Th_3
10.000	8.333	1.912

$T \cdot W/P_F \cdot MP_S$

	Th_1	Th_2	Th_3
F_A	**10.000**	8.333	1.147
F_B	6.000	**5.000**	1.912
F_C	1.000	0.833	**1.625**

Fig. 9. Determining the fleet engagement priority.

communication is important to know which mechanisms will be more sensitive to degradation in the communication environment.

- *Centralized:* Some mechanisms use centralized information and decision-making. Obviously, a centralized method is less robust than an equal but decentralized method. Indeed, in centralized mechanisms, the failure of a single agent (the coordinator) can make the process abort, or at least significantly reduce the quality of the solution. Furthermore, a centralized mechanism implies a certain hierarchy and authority structure. While this structure is present in most military contexts, there are systems where having an authoritative agent might be unwelcome.
- *Ship importance:* We have seen that in some cases some ships are more important than others. A ship with higher importance could be a commanding ship, an escorted supply ship, a coalition ship, etc. Some mechanisms deal with the relative importance of ships, while others do not take into account these ranking methods.
- *Backup plan:* This attribute offers the possibility of integrating backup plans in the mechanism because in a stochastic environment such as project NEREUS, it is important to be able to implement and use backup plans, in the case where an agent is unable to take care of its assigned tasks.
- *Completeness:* In project NEREUS, it is unacceptable to let threats reach ships unimpeded. Therefore, the completeness of a solution (whether every task is distributed) is important.

The evaluation of these factors for each coordination mechanism is presented in Table III, which compares the different coordination mechanisms. In this table, *m* is the number of frigates, n is the number of threats and p is the number of times a contingency arises. Thus, in the case of the *zone defence*, p is the number of times the P_S valuation is below the threshold.

B. Performances in Terms of Survival Rate and Resource Utilization

In the following figures, we compare the results for different coordination mechanisms. We used many different values for the various parameters described earlier, and we averaged the results to get a good idea of the performances of the mechanisms. However, since the frigate per threat parameter had a too great impact on the results of the Contract Net mechanism, we also included the results for the Contract Net protocol with only one frigate per threat, which is noted "Contract Net*" in the figures. Notice that that the Contract Net protocol has been tested with 1 to 4 frigates/threat.

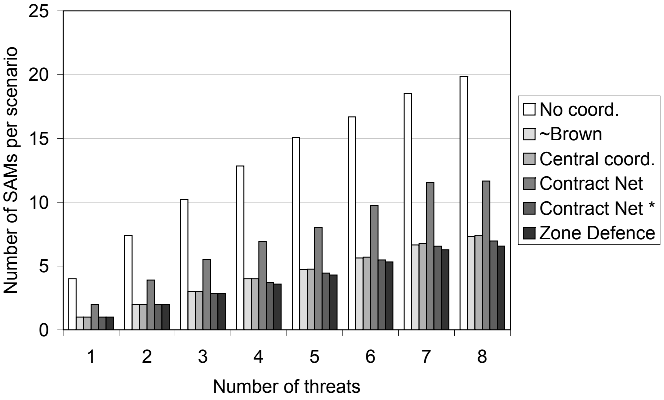


Fig. 10. Number of SAM planned per scenario, considering different coordination mechanisms.

Fig. 10 presents the number of SAMs used in the different mechanisms. We see that most mechanisms will fire about the same number of SAMs, except for the Contract Net used with more than one frigate per threat. Scenarios where no coordination mechanism is used will also expend more SAMs, which is normal since in this case, every frigate considers that it must defend the fleet alone.

Fig. 11 illustrates the size of communications, depending on the number of threats and mechanisms used. Since the messages are roughly the same size, the graph is also representative of the *number* of communications that occurs. It is also

TABLE III

COMPARING COORDINATION MECHANISMS.

	Zone defence	Central coordination	Contract Net	∼Brown
Coord. type	Social laws	Communications	Communications	Communications
Number of comm.	$4p$	$m + n\%(m+1) + 2(m-1) \cdot \lceil \frac{n}{m} - 1 \rceil$	$n \cdot (2m-1)$	$m(m-1) \lceil \frac{n}{m} \rceil$
Comm. importance	Useful	Very Important	Very Important	Very Important
Centralized		✓	✓	
Ship Importance				✓
Backup plans	✓	✓	✓	✓
Completeness	✓	✓	✓	✓

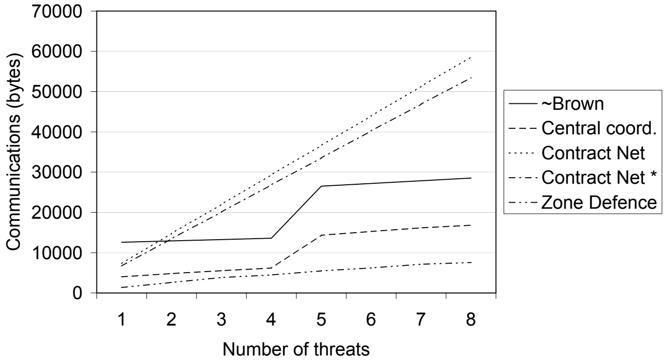


Fig. 11. Total size of communications per scenario, considering different coordination mechanisms.

interesting to see that the results are coherent with the analysis provided in Table III. Note that there are no communications in uncoordinated scenarios.

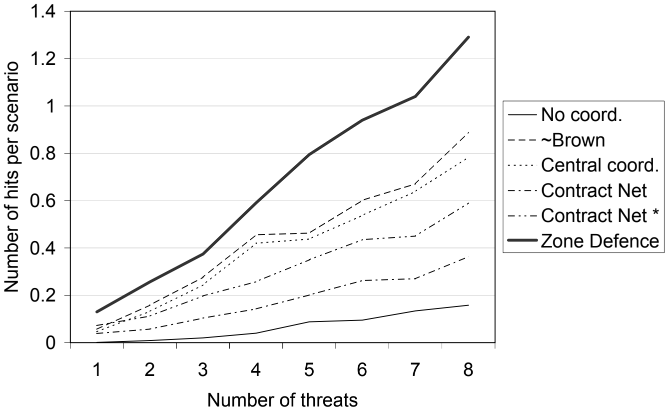


Fig. 12. Number of hits per scenario, considering different coordination mechanisms.

Fig. 12 shows the number of hits per scenario, comparing the coordination mechanisms. We see that the number of hits is greater in the *zone defence* mechanism, which can be explained by the fact that the division in sectors around the frigate generates many situations where the plan is suboptimal[5]. However, we see that this mechanism also uses far fewer communications than the other coordination mechanisms. Therefore, approaches based on conventions are still to be considered, but there is room to find another approach that would gives better results. On the other hand, we see that using no coordination gives a low number of hits per scenario. This is normal since each frigate engages every threat. However, as seen in Figure 10, there are many resources used when not coordinating.

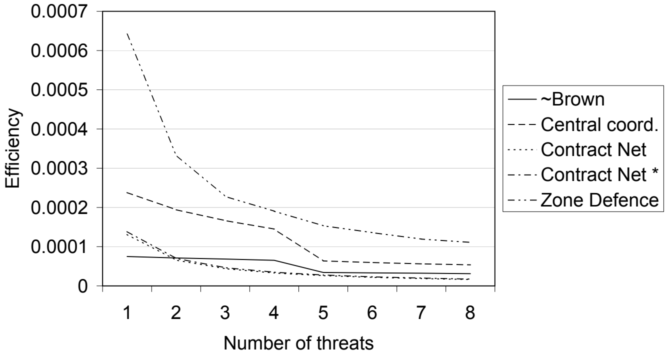


Fig. 13. Efficiency relative to communications use, considering different coordination mechanisms.

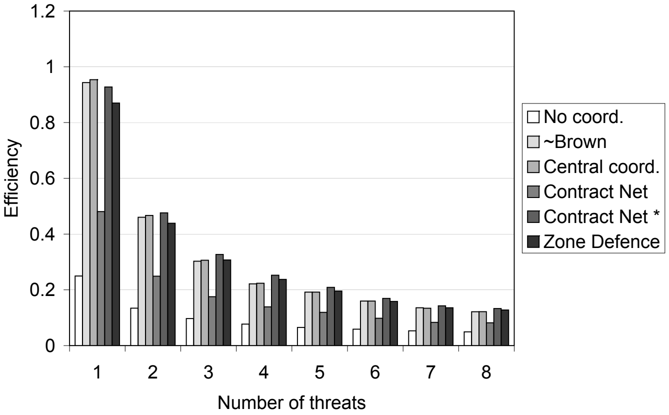


Fig. 14. Efficiency relative to SAM use, considering different coordination mechanisms.

To evaluate the different coordination mechanisms in terms of efficiency, we need to define *performance measures*. The *performance* metrics used to evaluate the coordination mechanisms are: the *survival rate* and the *utilization of resources*. Thus, using these two parameters, we determine the *efficiency*, which is the percentage of threats destroyed considering the resources used to destroy them $(1 - \%$ *of hits* $/$ *nb resources used*$)$. However, in our case, the survivability is far more important than the total of resources used. Thus, if we transform

---

[5] Further discussions of the results of the zone defence mechanism are provided in [16].

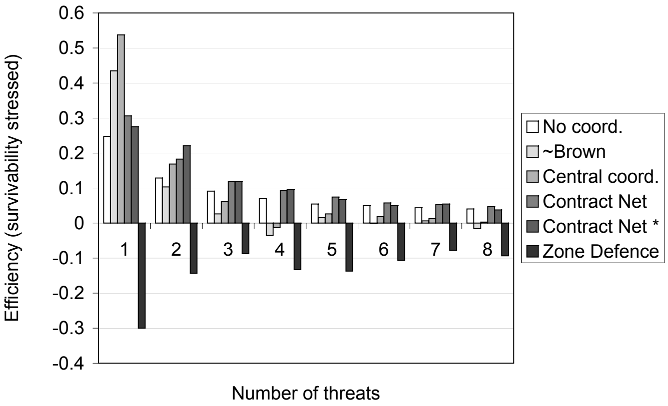


Fig. 15. Efficiency (with survivability stressed) relative to SAM use, considering different coordination mechanisms.

the efficiency calculus by adding an α parameter to stress the fact that it is critical not to be hit by threats, the efficiency becomes: $(1 - \alpha \cdot \%$ *of hits* / *nb resources used*$)$. We looked at two different efficiency measures: the efficiency according to the SAMs launched and according to the total size of sent messages.

Fig. 13 presents the efficiency according to the utilization of communications. In this figure, we see that the zone defence mechanism is very efficient for this metric, which can be explained by the fact that it does not use many communications. On the other hand, Fig. 14 presents the efficiency of the mechanisms, according to the utilization of SAMs. In this figure, we see that the Contract Net with more that one frigate engaging each threat and the scenarios where no coordination is used are less efficient than the other mechanisms. This corresponds to the fact that in these two cases, more SAMs are expanded to counter incoming threats. However, we should introduce an α parameter in the calculus of efficiency, as presented earlier, since these results are not biased enough toward the importance of survivability. Therefore, we present the efficiency (according to SAM use) with an α of 10 in Fig. 15. We see that, overall, the Contract Net gives very good results.

VI. CONCLUSION AND FUTURE WORK

We first presented the complex problem of managing resources in a fleet of combat ships. We then detailed the *threat allocation-coordination* problem and detailed four different mechanisms which contribute to solving it: the *zone defence* coordination, *central coordination*, the *Contract Net* approach and the $\sim$*Brown* coordination mechanism.

Experiments of these mechanisms in the context of a simulated fleet of combats ships considered as a team of “autonomous” agents show interesting results. Firstly, the results showed that an increase in survivability leads to an increase in the number of frigates engaging the same threat. However, work is still to be done on the redundancy in engagements. It is important to prioritize the threats differently when constructing plans, so as to make sure that the threats are engaged uniformly.

The results also showed that the communications channels were not much as used as we first expected them to be, and therefore, the communications do not seem to be as problematic as we first imagined. However, communication-based mechanisms are especially hard to scale up. Will these mechanisms be as successful if we double the number of threats and frigates? On the other hand, it is usually easier to scale up mechanisms with convention-based and learning-based approaches; thus, we could develop new coordination mechanisms based on these approaches. For example, coordination based on learning is often seen in small teams requiring quick reactions such as SWAT teams. Agents using such Special Weapons And Tactics (SWAT)[6] coordination mechanisms should predict the choices of other members of the team and select their actions to complement the team plan. In such mechanisms, the agents would learn their role in the team and the appropriate reactions according to various situations. SWAT is scheduled as future work in the context of NEREUS.

Furthermore, we have seen that the coordination mechanisms proposed each has its strengths and weaknesses. In the case where the communications are unavailable, the zone defence can be used with appreciable results. However, when there is enough time left and the communications channels are fully functional, a Contract Net approach gives very good results. We see that different situations lead to different responses. Therefore, we suggest the use of a metalevel decision agent to decide which movement method, planning algorithm and coordination mechanism use, depending on the situations. This could be implemented using the meta-deliberation technique proposed by Dean and Boddy [30].

Finally, current planning algorithms do not take into account the possibility that actions can be refused by operators. New agents could be designed that correspond to human operators and interact with DSS agents in a cooperative way. This would allow i) exploring the interactions between human and agents in real-time systems, ii) improving current algorithms to take into account the choices of human operators and iii) modeling the complexity of negotiation and coordination between human operators in a hierarchical structure. To achieve that, it is important to design interfaces between operators and DSS which is based on skills, rules, knowledge taxonomy of cognitive control as suggested by Rasmussen [31]. Thus, one can design ecological interface design (EID) [32] based on this taxonomy of which the goal is twofold: (1) not to force processing to a higher level than the demands of the task require and, (2) to support each of the three levels of cognitive control. These three levels are:

1) *Routine.* These are very familiar events which occur frequently and for which operators have accumulated skills, required to deal with, as a result of considerable amount of experience and training.
2) *Familiar events.* These are anticipated events which occur infrequently and for which operators will not have

---

[6] The SWAT team is responsible for crisis resolution which is appropriate and it aims to maximize safety, and deployment in lieu of other patrols when specialized coverage is required.

a great deal of experience to rely on. However, this kind of events have been anticipated by plant designers, who have built in means to deal with them (under the form of procedures, DSS, automatic controllers, etc.). These anticipated solutions provide operators with the help they need to cope with this kind of events.

3) *Unfamiliar events.* These are unanticipated events which are unfamiliar to operators because they rarely occur. Unlike the previous category, however, they have not been anticipated by designers and consequently, operators cannot rely on any anticipated solution, but must improvise one themselves.

ACKNOWLEDGMENT

We would like to thank the anonymous referees for their valuable comments and suggestions that helped to improve the presentation and the quality of this paper. We are also grateful to Mme Phyllis Daly for patiently polishing our English. This work has been supported by the National Sciences and Engineering Research Council of Canada (NSERC) and in part by Lockheed Martin Canada.

REFERENCES

[1] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach (second edition)*. Englewood Cliffs, NJ: Prentice-Hall, 2003.

[2] M. Wooldridge, *An introduction to multiagent systems*. Chichester, England: John Wiley & Sons, Inc., February 2002.

[3] P. G. Kropf, B. Chaib-draa, and B. A. Chalmers, “Resource management in socio-technical systems: a multi-agent coordination framework,” in *HMS 2000*, Portofino, Italy, October 2000, p. pp. 6570.

[4] M. Athans, “Command and control (c2) theory : a challenge to control science,” *IEEE Transactions on Automatic Control*, vol. AC-32, no. 4, pp. 286–293, April 1987.

[5] J. R. Boyd, “Organic design for command and control,” May 1987, briefing slides.

[6] ——, “The essence of winning and losing,” January 1996, briefing slides.

[7] C. Applegate, C. Elsaesser, and J. Sanborn, “An architecture for adversarial planning,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 1, pp. 186 – 194, January-February 1990.

[8] D. Bertsekas, M. Homer, D. Logan, S. Patek, and N. Sandell, “Missile defense and interceptor allocation by neuro-dynamic programming,” *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 30, no. 1, pp. 42–51, January 2000.

[9] R. Kewley and M. Embrechts, “Computational military tactical planning system,” *IEEE Transactions on Systems, Man and Cybernetics, Part C*, vol. 32, no. 2, pp. 161–171, May 2002.

[10] D. W. Oard, A. Ephremides, and S. I. Wolk, “On the integrated scheduling of hardkill and softkill assets using dynamic programming,” Naval Research Laboratory, Tech. Rep. NRL-FR-5750-94-9721, 1994.

[11] D. Blodgett, P. Plamondon, B. Chaib-draa, P. Kropf, and E. Bossé, “A method to optimize ship maneuvers for the coordination of hardkill and softkill weapons within a frigate,” in *7^{th} International Command and Control Research and Technology Symposium (7^{th}ICCRTS)*, Québec, QC, September 2002.

[12] Z.-J. Lee, S.-F. Su, and C.-Y. Lee, “Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics,” *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 33, no. 1, pp. 113– 121, February 2003.

[13] D. Blodgett, S. Paquet, P. Plamondon, B. Chaib-draa, and P. Kropf, “Coordinating plans for agents performing AAW hardkill and softkill for frigates,” in *Proceedings of The 2001 AAAI Fall Symposium Series*, North Falmouth, MA, July 2001, pp. 1–8.

[14] G. Weiss, “Prologue,” in *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, G. Weiss, Ed. Cambridge, MA: The MIT Press, 1999, pp. 1–23.

[15] P. Plamondon, “A frigate survival approach based on real-time multi-agent planning,” Master’s thesis, Computer Science & Software Engineering Department, Laval University, 2002.

[16] P. Beaumont, “Multi-platform coordination and resource management in command and control,” Master’s thesis, Computer Science & Software Engineering Department, Laval University, 2004.

[17] R. Becker, S. Zilberstein, and V. Lesser, “Decentralized markov decision process with even-driven interactions,” in *Proceedings of the 3nd International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS’03)*, N. Jennings, C. Sierra, L. Sonenberg, and M. Tambe, Eds. ACM Press, 2004, pp. 302–309, 19–23 July 2004, NY, USA.

[18] R. T. Maheswaran, M. Tambe, and E. Bowring, “Taking dcop to the real world: efficient complete solutions for distributed multi-event scheduling,” in *Proceedings of the 3nd International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS’03)*, N. Jennings, C. Sierra, L. Sonenberg, and M. Tambe, Eds. ACM Press, 2004, pp. 310–317, 19–23 July 2004, NY, USA.

[19] E. H. Durfee, “Scaling up agent coordination strategies,” *IEEE Computer*, vol. 34(7), pp. 39–46, 2001.

[20] N. R. Jennings, “Coordination techniques for distributed artificial intelligence,” in *Foundations of Distributed Artificial Intelligence*, G. M. P. O’Hare and N. R. Jennings, Eds. John Wiley & Sons, 1996, pp. 187–210.

[21] S. Lizotte, “Coordination dans les sociétés d’agents: une approche basée sur la gestion des dépendances,” Master’s thesis, Computer Science & Software Engineering Department, Laval University, 1996.

[22] T. W. Malone and K. Crowston, “The interdisciplinary study of coordination,” *ACM Comput. Surv.*, vol. 26, no. 1, pp. 87–119, 1994.

[23] C. Boutilier, “Planning, learning and coordination in multiagent decision processes,” in *Theoretical Aspects of Rationality and Knowledge*, 1996, pp. 195–201.

[24] C. Brown, P. Fagan, A. Hepplewhite, B. Irving, D. Lane, and E. Squire, “Real-time decision support for the anti-air warfare commander,” in *6^{th} International Command and Control Research and Technology Symposium (6^{th}ICCRTS)*, U.S. Naval Academy, Annapolis, Maryland, June 2001.

[25] B. J. Clement and A. C. Barret, “Continual coordination through shared activities,” in *Proceedings of the 2nd International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS’03)*, J. S. Rosenschein, T. Sandholm, M. Wooldridge, and M. Yokoo, Eds. ACM Press, 2003, pp. 57–64, 14–18 July 2003, Melbourne, Australia.

[26] J. Odell, H. Van Dyke Parunak, and B. Bauer, “Representing agent interaction protocols in uml,” in *First international workshop, AOSE 2000 on Agent-oriented software engineering*. Springer-Verlag New York, Inc., 2001, pp. 121–140.

[27] R. G. Smith, “The contract net protocol. high-level communication and control in a distributed problem solver,” *IEEE Transactions on Computers*, vol. C-29, no. 12, pp. 1104–1113, December 1980.

[28] The Foundation for Intelligent Physical Agents, “Fipa contract net interaction protocol specification,” FIPA, Component 00029, December 2002, interaction Protocols.

[29] C. Brown and D. Lane, “Anti-air warfare co-ordination - an algorithmic approach,” in *Command and Control Research and Technology Symposium*, Monterey, CA, USA, October 2000.

[30] T. Dean and M. Boddy, “An analysis of time-dependant planning,” in *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI-88)*. Menlo Park, California: AAAI Press, 1988, pp. 49–54.

[31] J. Rasmussen, *Information processing and human-machine interactions: an approach to cognitive approach*. New-York: North Holland, 1986.

[32] K. J. Vicente and J. Rasmussen, “Ecological interface design: theoretical foundations,” *IEEE Trans. on Syst., Man and Cybernetics*, vol. 22, no. 4, pp. 589–606, 1992.