Autonomous Planning for Surface Engagement using a Multi-Agent System

J.M. de Gier MSc[a*], L.M.J. Nijsten MSc[a], H.J. Veltman MSc[a], T.H. de Groot PhD[b]

[a]*Intelligent Autonomous Systems Group (IAS), Dutch Organization for Applied Scientific Research (TNO);*
[b]*Maritime Systems Department, Defence Materiel Organisation (DMO), The Netherlands*
*Corresponding author. Email: jan.degier@tno.nl

Synopsis

Less personnel and the introduction of more complex systems on board of future navy vessels requires innovative systems to support the crew in making optimal command decisions. These decisions should result in an allocation of available resources that maximizes the probability to achieve the set of mission goals, provided the external and internal picture. The process of planning tasks and resources across subsystems is challenging, partly due to the heterogeneous nature of these, often proprietary, subsystems and the inter-dependencies between tasks and resource requirements. Currently, the coordination of systems-of-systems on board of navy vessels is predominantly performed by the human crew.

This article presents the use of the multi-agent paradigm for supporting the crew by automatically allocating the set of resources that are available on board a vessel to the vessel's systems. The presented solution leaves the crew in the loop as the ultimate decision maker but facilitates their decisions by providing pre-processed options void of unnecessary details. The solution works as follows: multiple goal-driven software agents cooperate to plan tasks and allocate resources. A single agent can be responsible for vessel functionalities such as sensor–, engagement–, and mobility–management. These agents collaborate autonomously with each other and under dynamically changing external and internal conditions, changing tactics, mission goals and resource availability. Agent decisions are driven by the set of mission goals expressed in a mathematical formulation. This ensures that system resources and assets are allocated in accordance with the mission without increasing the workload of the crew. Conflict resolution resulting from resource usage by agents is handled using a negotiation protocol.

The feasibility of the taken approach is demonstrated within an example scenario, where a vessel escorts a high-value asset in enemy territory. This article shows that with the multi-agent paradigm, the described conditions, and in the face of conflicting preferences, the most effective combination of actions is selected and planned.

Keywords: Multi-Agent Systems (MAS), Negotiation, Goal Driven, Utility, Planning, Resource Management

1 Introduction

In the next decade the Royal Netherlands Navy (RNLN) will introduce four new classes of ships to replace existing classes. This is an opportunity to get new solutions and innovations on board. The *Manning & Automation* program aims to maintain and enhance the operational capability of naval systems, given the challenges of reduced manning and the opportunities of automation. It is established with a vision and an ambition to bring the Royal Netherlands Navy into the next era of modern warfighting.

The challenge addressed in this paper is formulated as follows: *research and develop new ways of cooperation between humans and systems and between systems themselves to achieve the mission goals of future ships with reduced complements.* The focus of this paper is on the interaction between a subset of systems within a navy vessel. It is not centered around physical– / data– / information–integration but on a decomposition of tasks on a conceptual level, as is standard practice when these tasks are performed by the human crew. For example, engaging an incoming threat is decomposed in detecting / tracking / aiming / shooting.

Authors' Biographies

Jan de Gier obtained his master's degree in Applied Mathematics from the Delft University of Technology and is currently a Research Scientist at the IAS group of TNO. His research focus is on the modelling of and the goal-driven reasoning about information in dynamic and hostile environments. This results in novel reasoning frameworks and algorithms to optimise the decision making processes and the scheduling of resources in multi-agent systems.

Loek Nijsten earned his master's degree in Nanotechnology at the University of Twente and is now a Consultant at the Intelligent Autonomous Systems group of TNO. His research interests include autonomous systems, specifically planning and scheduling in the naval domain.

Kim Veltman has a master's degree in Artificial Intelligence from the university of Groningen, with a specialisation in multi-agent systems. Currently she works as a Research Scientist at TNO. Her research interests include multi-agent systems and modeling intelligent behaviour in agents, specifically with respect to planning and scheduling in the naval domain.

Teun de Groot is a Senior Naval Systems Integrator at DMO. He is a specialist in automatic system optimisation, guides various Manning & Automation projects, and uses his expertise to realise a new balance between humans and machines on board future navy ships.

 https://doi.org/10.24868/issn.2515-818X.2020.016

1.1 Contributions of this article

In this article, the authors address challenges in the field of naval engineering by combining established engineering approaches and computer science paradigms (Benaskeur et al., 2007; Arazy and Woo, 2002). Specifically, *distributed task and resource-allocation* are investigated, using a *goal-driven multi-agent systems paradigm*. The former is realised using *negotiation between agents*, the latter places emphasis on the *autonomous operation of the individual software components*. Using results obtained from *simulations* on an *implemented and tested software demonstrator* (cf. Figure 1), the authors *provide critical reflections* on the design choices made and *offer advice* for future work / projects addressing similar challenges.

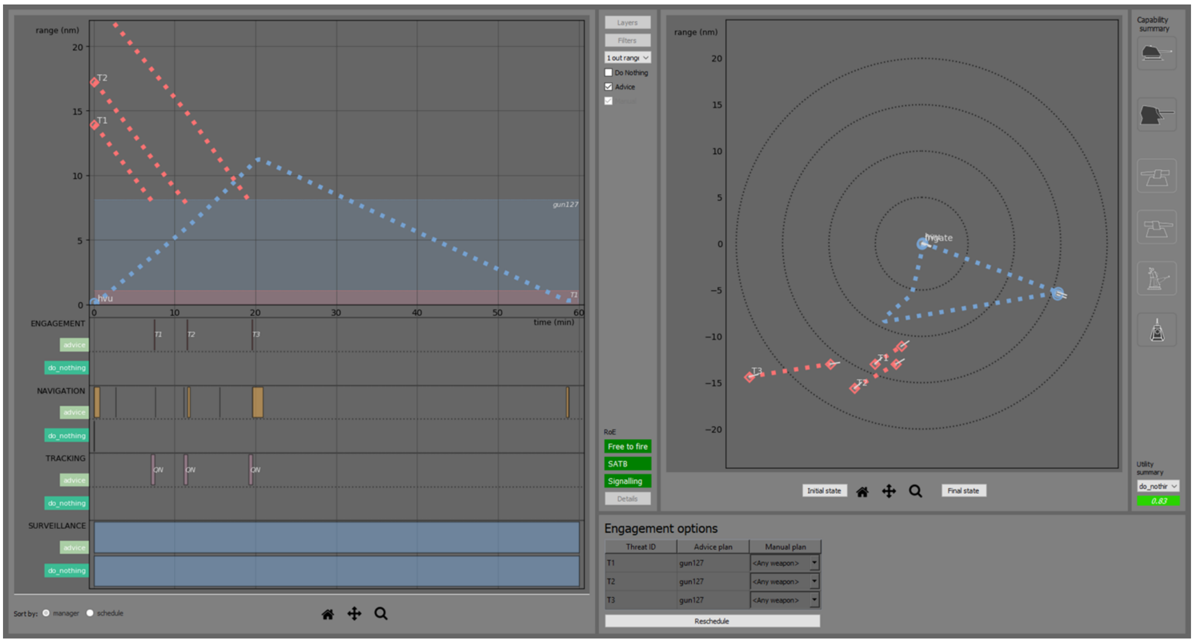

Figure 1: Visual output of the implemented demonstrator: on the right the courses of our vessel, a High-Value Unit (HVU) (blue), and three hostile contacts (red) are plotted. On the left (upper section) the distances of all vessels (dotted lines) relative to our vessel are displayed (HVU in blue, the hostiles in red). In the same window, the vessels' respective effective firing ranges are shown as the colored areas. In the shown simulation, the attacking vessels are disabled before coming within their own weapons ranges. The dotted blue line is our distance to the HVU, which first increases as our vessel engages the targets and then approaches zero as the blue vessels reunite. The corresponding plan (the interaction between the relevant vessel's subsystems) is shown as a Gantt chart on the lower left section. For comparison, see Figures 5 and 8 where other outcomes are shown.

1.2 Overview and outline of this article

The flow of this paper is as follows: in Section 2 three challenges are formulated (§2.2) and related to the a practical example scenario (§2.3). This paper then (Section 3) focuses on five design choices (§3.1 – §3.5) addressing these challenges, with each subsection providing motivation for as well as critical reflections on these choices. The validity of the provided insights is motivated in Section 4 by a brief presentation of the realised demonstrator, which simulates operations aboard a vessel of the RNLN in the example scenario. The article closes with a conclusion (Section 5) and bibliography.

2 Main Challenges and our Use-case

A navy vessel is a system of proprietary, highly specialized and distributed subsystems (cf. §3.1). Often these subsystems are commercial products using specific, and system-dependent interfaces and data representations (Janssen, 2019). When specifics of internal operations are confidential the system may not share all engineering information to the system integrator.

Due to this, any integration effort is bound to treat this population of systems as individuals with their own properties and internal states that are hidden from the integrating engineer. In computing science, this view can be represented by the Multi-Agent Systems (MAS) paradigm (cf. §3.2). However, when these subsystems are required to cooperate towards a greater goal, sharing of information is crucial.

 https://doi.org/10.24868/issn.2515-818X.2020.016

2.1 Brief overview of the application domain

For a complex system-of-systems such as a naval vessel, the process of planning tasks and resources is challenging (cf. §3.4). Vessels consist of interacting subsystems, with many interdependent tasks. In such systems, humans and intelligent machines are required to work together and to be able to adapt their behaviour under changing conditions and situations to reach their shared goals (Kester, 2010). Currently, this coordination is mostly a human task relying heavily on the operator's experience and inside knowledge of the subsystems. Timely consideration by the operator of all parameters and constraints is stressful and as the number of threats increases, impossible. The ultimate goal of the approach presented in this paper is to appropriately allocate system resources to achieve equal or even better operational effectiveness with less personnel in a shorter amount of time.

2.2 Description of challenges

Creating a MAS that allows agents to obtain their own world views and endows them with individual goals and preferences, can translate to opposing priorities and demands. Therefore, when designing such a system, the following challenges emerge.

2.2.1 Challenge 1: Common knowledge and shared beliefs

Software agents often operate on locally stored information (the agent's knowledge and beliefs). When multiple agents are expected to cooperate, the sharing of information and the updating of each other's believes is a non-trivial exercise. Since the agents share mission goals on vessel level, synchronising the representation and interpretation of the world is necessary to enable contributions to those goals.

2.2.2 Challenge 2: Goal driven reasoning

Intelligent agents need an objective function or goal function to evaluate and decide on their behaviour. In dynamic environments and with operator defined goals, this function can be subject to change during run-time. When multiple agents must cooperate towards shared higher goals (the mission goals), it is important that the agents share these goals. However, interpreting a high level mission goal, such as *"escort and protect another vessel"* to actions for specific software agents poses a number of challenges such as: how to evaluate the agent's actions in the greater context, how to resolve conflicting actions on agent-level and how to address inter-dependencies between agent domains (for example, when one agent depends on another for a resource critical to its performance). In accordance with the literature (De Groot, 2015), we propose to use a utility function (De Gier et al., 2019b; Kester and Ditzel, 2014) to evaluate which tasks to plan and resources to allocate.

2.2.3 Challenge 3: Resource management

The management of resources, which takes place while planning tasks, is traditionally considered under the assumption that there is a shortage, i.e. that not every demand can be fulfilled at all times. In the scenario at hand, this is potentially the case whenever operational resources such as power, chilled water, sensor bandwidth or the vessel's course are required by multiple agents at the same time (De Gier et al., 2019b). Furthermore, the output of the system is a plan of tasks *over time* meaning that the allocation of resources has to coincide with specific moments in time as well as states of the world.

2.3 Representative example scenario

The example scenario used to demonstrate the MAS involves a vessel escorting a High-Value Unit (HVU), e.g. an oil tanker, through a narrow strait. In this area multiple small surface threats emerge. In this asymmetric warfare situation, the vessel is tasked with protecting the HVU (e.g., see Figures 1, 5 and 8).

2.3.1 Example systems

In our use case, the vessel is discretized to a number of subsystems relevant to the scenario (De Gier et al., 2019b). Analogous to the real world, subsystems are controlled by individual software components. Specifically, the vessel's sensor suite is used for situational awareness, and for providing high fidelity tracks of contacts with an hostile intent. Weapon systems are used for engagements, and tactical navigation may impact the performance of both these agents (see Figure 4).

 https://doi.org/10.24868/issn.2515-818X.2020.016

2.3.2 *Challenges revisited*

The challenges identified in §2.2 are represented in the use case as follows:

Common knowledge and shared beliefs The software agents governing different subsystems have access to, and control over, different information and resources. The radar system, e.g., has information on targets needed by the agent in charge of the weaponry, but is at the same time dependent on the agent controlling the course and speed of the vessel to perform its own operations successfully. Furthermore, the operational readiness of the agent that reasons about the radar is dependent on the agents controlling resources such as energy or chilled water.

Goal driven reasoning Intelligent agents need an objective function to evaluate and decide on their behaviour. A mission goal such as escorting and protecting an HVU is not easy to interpret directly in terms of actions or resource allocations for most systems. For instance, the position of the vessel can be considered a separate resource to which different agents may lay claim, as both the sensing and the engagement agent rely on the vessel's position to track or engage opponents, respectively. As these agents' services are both required to achieve the overall objective, the agents must include not only their own goals into their reasoning process.

Resource management Ultimately, the complexity of naval operations arises from uncertainties regarding the state of the real world as well as the fact that resources are constrained. The latter refers to ammunition, fuel, electrical power or chilled water as well as to the services of specific agents, controlling e.g. the course and speed of the vessel, the operation of weaponry or the delivery of targeting information from the radar system. Orchestrating the usage of these resources across agents and over time is a fundamental challenge (§2.2.3).

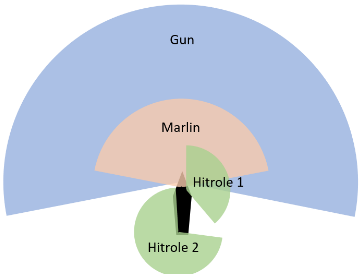

Figure 2: The weapon coverage (indicative values). **Gun**: bearing $-100°$ to $100°$, optimal range 8 km; **Marlin**: bearing $-75°$ to $75°$, range 300 m to 2400 m; **Hitrole 1**: bearing $0°$ to $150°$; **Hitrole 2**: bearing $105°$ to $-10°$, range 50 m to 1200 m.

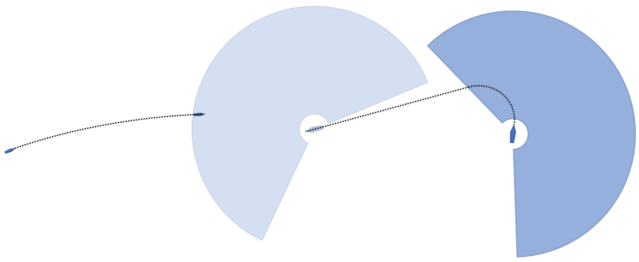

Figure 3: Illustrating agent interdependencies using position (mobility) and weapons orientation (engagement). Weapons have an effective range and operate only within a specific angle relative to the vessel's orientation. This implies required cooperation between the engagement and the mobility agent (as well as (not shown) the sensor agent for updates on target locations.

3 Design Choices and Lessons Learned

3.1 *Design choice 1: the developed system is fundamentally distributed*

3.1.1 *Motivation*

The distributed approach facilitates a scalable, extensible, modular setup. This also makes it easier for system providers to interact with other (sub)systems without having to share all internal knowledge.

3.1.2 *Critical reflection / lessons learned*

A distributed approach results in smaller coherent sets of variables, which reduces the computational cost to determine good solutions. For overlapping (i.e., shared) variables (due to, e.g. simultaneous usage of a resource) inter-agent planning is required to ensure the feasibility of resulting solutions. This can be computationally expensive and not all approaches are guaranteed to converge on acceptable solutions. With regard to the work referenced in this paper, the number of overlapping variables were small providing a judicious distribution of variables, which improved the probability for convergence. Extending the approach to modularise more individual subsystems will certainly lead to increased communication overhead. The literature suggests hierarchical compartmentalizing as a solution to this (Ephrati and Rosenschein, 1994).

 https://doi.org/10.24868/issn.2515-818X.2020.016

3.2 Design choice 2: adopting the Multi-Agent Systems paradigm

3.2.1 Motivation

Since the 1980s multi-agent systems have been studied as a field on their own and have since then gained popularity, partly due to the possibility of adding agents to distributed systems (Wooldridge, 2001). In addition to being inherently distributed and extensible, using a MAS allows autonomy of the heterogenous agents by separating beliefs, intentions and capabilities.

3.2.2 Approach / Realisation

The subsystems, represented individual agents, have jurisdiction over, and are dependent on, different capabilities and resources. As all agents share the vessel's mission goal, they need to cooperate and provide resources and capabilities to one another. They determine to do so individually and based on their local information, assuming only changes to that local knowledge during the planning process (cf. Figure 6).

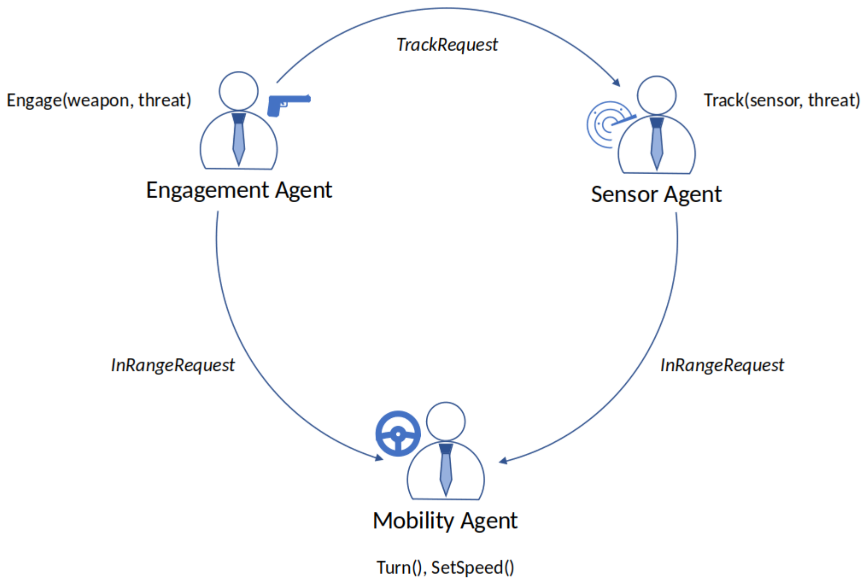

Figure 4: Dependencies between the agents involved in the planning process shown in Figure 6.

3.2.3 Critical reflection / lessons learned

Enabling agents to share local / partial plans to enable their counterparts to account for their needs increases performance and decreases planning time. It does, however, require the timely exchange of this information (which is potentially problematic under real world conditions) and hinges on providing the right level of detail (which cannot always be determined by the individual). Providing not enough information can lead to non-converging behaviour as the agents are unable to identify inter-dependencies and resource bottlenecks.

Practically speaking, the engineer should be weary of overusing the MAS approach as e.g. creating separate agents for safe navigation and tactical navigation (two aspects closely related and operating almost entirely on shared resources) might result in excessive negotiations with a detrimental impact on overall performance.

3.3 Design choice 3: utility based reasoning

3.3.1 Motivation

Simon (1955)'s model of rational choice, subjective expected utility theory is based on the assumption that reasoning entities are perfect rational decision makers (Ai and Deng, 2017). While this may not be the case for humans (Dörner, 1996), machines can be expected to be consistent in their reasoning approaches (Hildmann, 2018). Given the dynamic and unpredictable domain, an objective function guiding rational, mission-driven resource allocation is needed (De Groot et al., 2014). Utility based reasoning has been proposed as an approach to relate operator defined mission objectives to a changing and hostile environment (Kester and Ditzel, 2014).

 https://doi.org/10.24868/issn.2515-818X.2020.016

3.3.2 Approach / Realisation

The information provided by the Commanding Officer on the mission goals and mission constraints maps onto a goal function or utility function. This is a mathematical function that weighs the outcomes of (mathematical) sub-utility functions, each of which assign a sub-utility value to a more conceptual feature of the world. An example of such a concept is the expected future damage to assets, which is derived from, e.g., the positions and capabilities of the assets in the current and predicted future situations.

3.3.3 Critical reflection / lessons learned

A conceptual world description, based on the aggregation of lower level information, facilitates behaviour evaluation in terms of the mission goals. However, this hinges on the assumption of unbiased and mutually orthogonal aggregated information. Furthermore, using a utility value as a linear combination of sub-utility values per aggregated concept greatly limits expressibility. While well-defined components – treated in isolation – enable experts to contribute their knowledge and assists operators to interpret decisions made by the system, this does come at a cost: if aggregated information is not orthogonal or when their aggregation is too subjective, the assigned utility value may not constitute a adequate quantification of the goals.

Practically speaking, the authors recommend preparing multiple aggregation functions and sub-utility functions in advance, in order to facilitate operators adapting the utility function in real-time to account for changes in the environment, the mission or the available resources. Realistically, only the weighting between different functions can be adapted in real time, implying the need for preparatory work in advance.

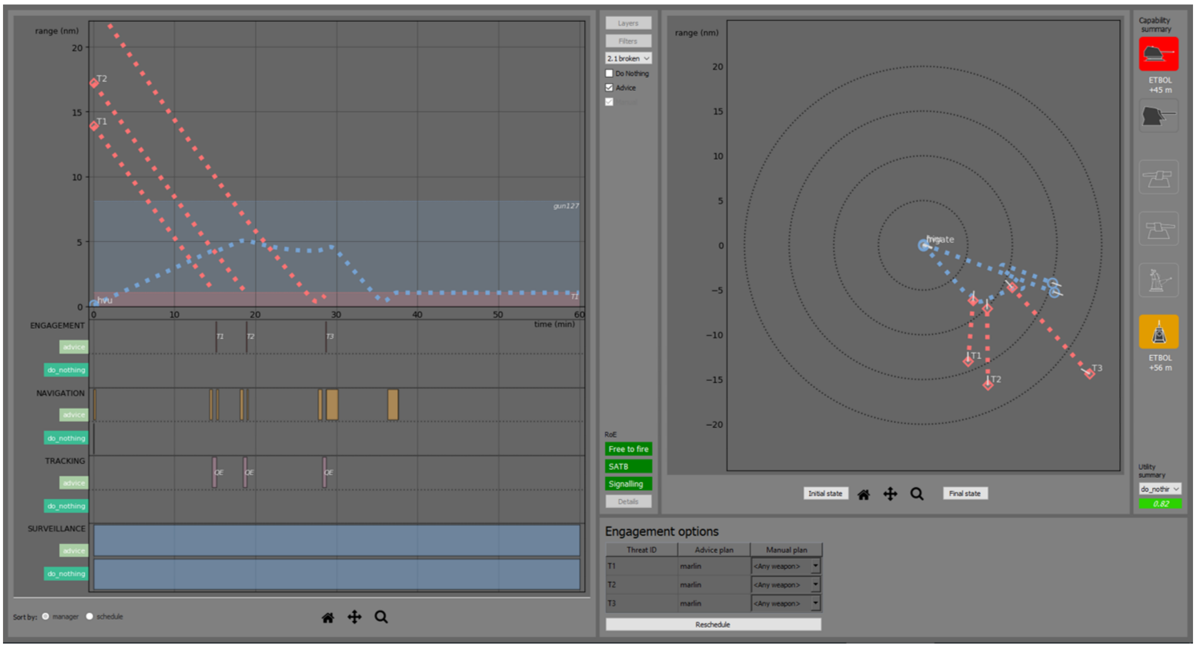

Figure 5: Visual output of the demonstrator. When compared to Figure 1 the warning signs (right margin) draw attention: both the main gun (top, red) as well as the radar (middle, orange) are unavailable (red) or limited available (orange). As before, the right window shows the courses of the two blue vessels and three hostile contacts (red). Since the main gun is unavailable, the Marlin with its shorter range (similar to the modelled enemies' weapon range) is used. This allows the hostile contacts to almost come withing their weapon range of the HVU.

3.4 Design choice 4: planning the execution of tasks and allocation of resources

3.4.1 Motivation

With dynamically changing goals, automated and goal-driven resource management is needed (De Groot et al., 2018). De Gier et al. (2019b) discuss multi-agent planning in the maritime domain, while De Gier et al. (2019a) report specifically on utility-based resource management between software agents aboard a navy vessel. Due to the complexity of interdependencies of the functionalities of a navy vessel, supporting the operators in their decision making is crucial. The system provides insight in tasks and resulting resource demands, by presenting feasible plans that help meet the mission goals.

3.4.2 Approach / Realisation

Agents use their local view on the world and shared mission goals (expressed as a utility function) to construct a plan, consisting of tasks to execute and resources to allocate. The agent level plans are local, with actions and resources that are under the agent's direct control. By negotiating (cf. §3.5) with other agents, all local plans will be compatible and together they form a globally feasible plan. Each agent provides an interface that indicates the type of requests that are interpretable by this agent during negotiation.

3.4.3 Critical reflection / lessons learned

Since each agent has the ability to propose multiple plans to meet the mission goals, one could expect an explosion of plans. In the current setup, most agents use local rules – such as preferences on resource usage – to avoid sending requests for multiple plans. Only if a plan turns out to be infeasible or undesirable, another plan option is considered. This somewhat interferes with the idea of globally maximising the utility function. However, the agents should strive to avoid an explosion of plans because this translates to unacceptable calculation times. Currently, if an agent is not granting a request for a task or resource from another agent, it will provide a moment in the future at which time that request will be granted. A refusal can be due to infeasibility (e.g., a resource is not available due to damage, a vessel has navigational limits) or a more utile alternative (for the overall objective). To help an agent choose what alternative requests to send, it is useful to extend the feedback of the denial with a reason why the request is denied.

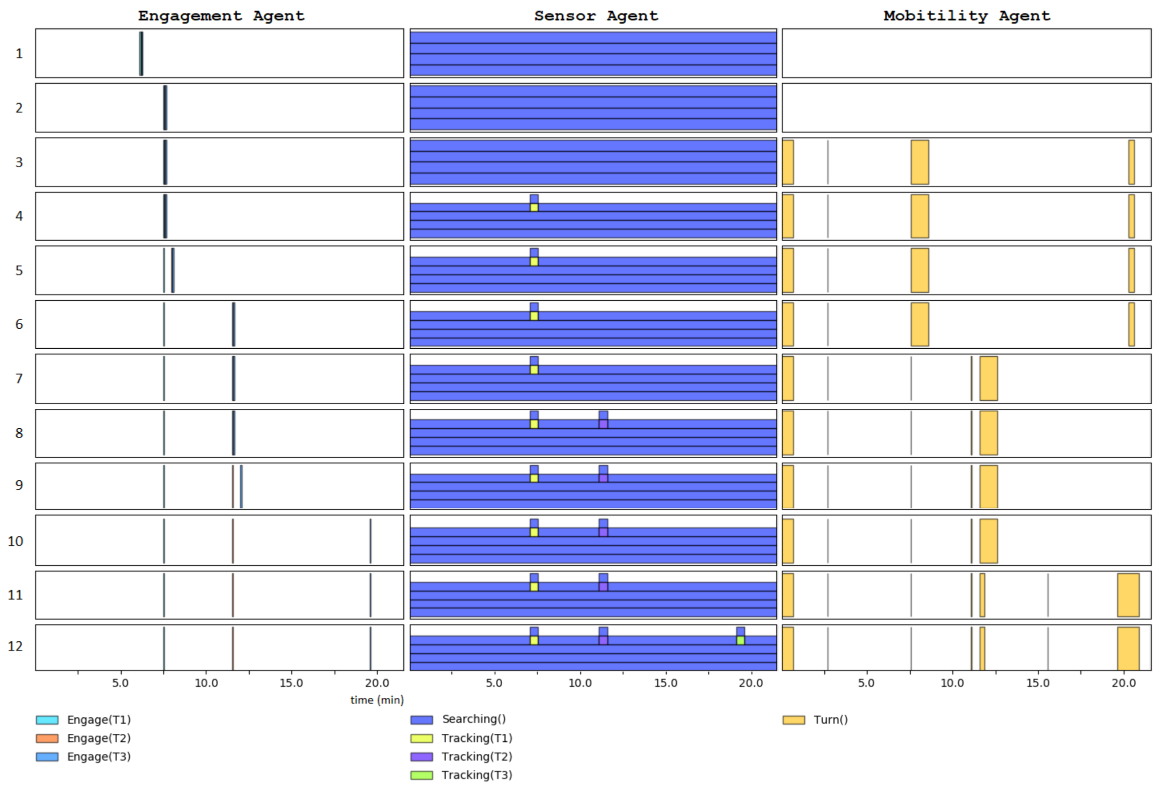

Figure 6: Visualisation of the evolution of a possible plan with local (agent specific) plans being adapted in several negotiation steps (from top to bottom) resulting in a complete and feasible plan for the scenario shown in Figure 1.

3.5 *Design choice 5: conflict resolution using inter-agent negotiation*

3.5.1 Motivation

The planning decisions of one agent can interfere with decisions of other agents due to interdependencies between tasks and resources. Resolving these conflicts is a core aspect of the application domain.

3.5.2 Approach / Realisation

To allow the alignment of the local plans of the agents, a negotiation protocol was devised. This protocol is loosely based on the generalised partial global planning (GPGP) approaches, as described in e.g. (Decker and Lesser, 1995), and consists of requests for actions and responses to these requests (cf. example in Figure 7). Agents are synchronised using a separate coordinating agent, the coordinator, to ensure that requests and responses arrive at the proper agent at the right moment during the process of planning.

 https://doi.org/10.24868/issn.2515-818X.2020.016

3.5.3 Critical reflection / lessons learned

The coordinator does not interpret the messages between agents and is therefore not able to have substantive interaction with the agents. This limits its capabilities to ensure convergence towards a feasible plan. It can, however, detect loops in the communication and hence determine if the planning process is diverging, in which case it can signal to the agents that a different approach should be taken.

Many conflicts between agents arise from the lack of knowledge on the partial plans that emerge during the planning process of other agents. Extending the capabilities of the coordinator so that it can indicate (utility based) preferences of not (yet) globally feasible agent plans to other agents can speed up the process and reduce the number of conflicts. This approach is investigated in a follow-up research program (Janssen, 2020).

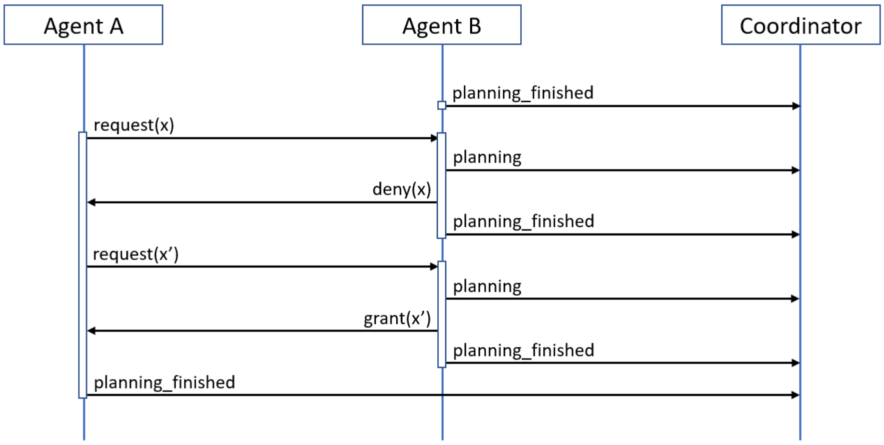

Figure 7: An example of the negotiation protocol for two agents (De Gier et al., 2019b) for inter-agent planning.

4 The Demonstrator

To showcase the approach a demonstrator was built, in which a number of separate agents were implemented, each with its own local knowledge (e.g. about the agent's capabilities) as well as information shared between the agents. The shared knowledge of agents involves the utility function and models on the evolution of the world, such as the behaviour model of the hostile contacts, which e.g., assumes a worst case interception course with the HVU. The implementation of the demonstrator was done in Python and for the interface PyQt was used.

4.1 Agent based implementation

Three agents were implemented, each with a different functionality: the sensor agent, the mobility agent and the engagement agent.

The **sensor agent** provides situational awareness through horizon search and the tracking of contacts. It delivers high fidelity tracks to the engagement agent. To perform optimally, this agent requires assistance from the mobility agent, for example, when a certain radar face is not available (cf. Figure 5) a repositioning of the vessel is required to ensure that the area in question is in view (of the radar system). The **mobility agent** in turn is responsible for the navigation of the vessel. It can perform tactical navigation, e.g. escorting a HVU, and - on a smaller timescale - navigate such that requests from other agents can be granted at the required times (cf. Figure 3). Finally, the **engagement agent** handles the engagement of hostile contacts. To achieve this, its internal knowledge consists mainly of the effects of its weapons (cf. Figure 2) but to effectively use these cooperation of the two other agents is required. Its decisions cover which weapon to use and in which order to engage the targets.

4.2 Performance evaluation

The demonstrator was developed to showcase the ability of the system design to handle detailed specific mission goals, and to evaluate its performance for such goals under different circumstances, such as (1) resource abundance (cf. Figure 1), (2) resource scarcity (cf. Figure 5) and (3) the changing to a new mission goal (cf. Figure 8). All of these scenarios feature extensive interaction between the different agents and required decisions such as whether (and when) to engage the enemy or break contact. These decisions commonly required collaboration with other agents, for example, high fidelity track information as well as access to the ship's movement are required for successful engagement, while the radar system delivering these tracks also depends on vessel orientation. The system was tested in the presence of (former) RNLN sailors / officers and performed well for the tested scenarios.

 https://doi.org/10.24868/issn.2515-818X.2020.016

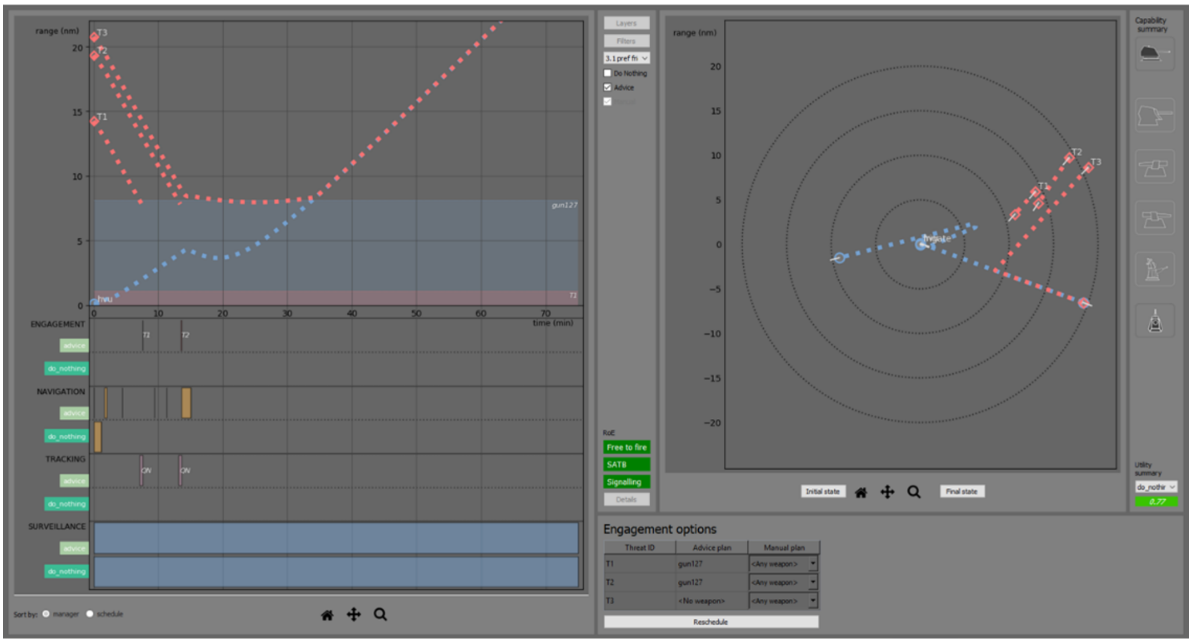

Figure 8: Visual output of the implemented demonstrator. When compared to Figures 1, 5 the most striking difference in the scenario is the difference in mission objective: the vessel is **not** tasked with protecting the HVU, it must preserve itself. Consequently, the courses sailed differ (right window), the vessel only engages the first two hostile vessels (red) before breaking off contact. The last red contact is left chasing the HVU. This can also be seen in the left window where both the red contact as well as the HVU gradually disappear in the distance.

5 Summary, Conclusion and the Road Ahead

This paper presents and discusses a multi-agent system for distributed task and resource allocation on RNLN vessels. The system, which was implemented as a demonstrator, assists a reduced crew in their task to allocate limited resources in a timely fashion. The goal of this paper is to inform the community of the lessons learned during this process and to offer recommendations.

The developed knowledge, concepts, techniques, and partial solutions enable follow-on initiatives to develop an Integrated Mission Management System to optimally support the crews on-board future RNLN vessels.

5.1 Summary

One fundamental decision was to follow the distributed MAS paradigm. For the presented work, this was motivated by the desire to create a scalable approach. Its inherently modular character allows for the adding of additional agents, resources and inter-dependencies without having to re-design the underlying architecture. It furthermore enabled the solving of the overall problem as a combination of smaller tasks.

Another important choice was to use a utility function to assess the profitability of a solution. This decision was taken based on the fact that such a function can be seen as a unified mathematical formulation of a goal. The difficulty herein lies in how to obtain unbiased aggregated information to which a utility value can be assigned.

Planning was solved by first planning locally for each agent individually and then merging plans between agents using negotiation on emerging dependencies. This can, at least in theory, lead to an explosion of possible options. It is hard to determine how the proposed options would ensure convergence towards the global plan. To investigate ways to reduce the impact of this, work in later projects used agents designed to guide the search by prioritizing approaches based in their utility value (see the section on ongoing and future work).

5.2 Recommendations and concluding remarks

The main recommendation for designing a distributed system like ours is to impose hierarchies on the MAS (Muñoz et al. (2018)) to avoid, or solve, conflicts. MAS are extremely powerful and versatile, but they can (as it did for us) come at the cost of a potentially large communication overhead as well as uncertainties regarding the timeliness or relevance of exchanged information. As a recommendation for future work in this domain, we suggest investigating ways to limit agent autonomy without affecting performance. One possible way to achieve this would be through a separate coordination agent which could embody the high-level planning aspect and guide (but not entirely control) the other agents with regard to their preferences for or against certain shared actions (thereby reducing the overhead as well as the complexity of the inter-agent decision making process).

 https://doi.org/10.24868/issn.2515-818X.2020.016

We furthermore recommend supporting non-orthogonal aggregation of information for the futility functions. To explain what is meant by this, consider that there is some correlation between the risk posed by an enemy and their capabilities. In the presented approach, we assume the (simplified) case where these two are orthogonal (unrelated). Such interrelations between aggregation components could be described by, for instance, so-called Choquet integrals (Grabisch and Labreuche, 2010).

Finally, the negotiation process bears the significant risk of losing sight of the overall objective. Ensuring that a sequence of choices ultimately (and within a feasible amount of time) results in a global plan is a challenge. Ensuring convergence of the negotiation could be achieved by expanding the functionality of the coordinator agent to include additional input from the individual agents to gain more insight on agent decisions. In addition, we suggest to search for a local optima as this will speed up the negotiation process.

5.3 Ongoing and future work

As for our own work, the experiences gained are already used to guide the design of systems in ongoing research projects, such as Smartship (Janssen, 2020), and to inform future projects of the Manning and Automation program.

For example, for a project following the one discussed in this paper, we designed the agents in such a way that the most utile approach was always evaluated first. Time permitting / in future projects, we plan to analyse and prioritise alternatives this way. Using the coordinator agent's insights and preferences as a guiding mechanism, an informed choice can be made with regard to which agents are asked to provide alternatives, potentially reducing the complexity of the search significantly. As recommended above, we plan to limit / reduce agent autonomy. One way of doing this is the mentioned coordinator agent which decides what plan / course of action to take based on the mission and situational information. Managers can then indicate that requests are undesirable for them because of the current actions that are relevant according to the coordinator, however, requests are only allowed to be denied when they are impossible.

The authors look to the community for fruitful discussions and collaborations for future projects. Therefore, and to drive this, we would like to conclude the paper by asking questions, meant to guide our own future work as well as to illicit feedback from the community. In the interest of brevity, our questions then are:

- How can we express the utility function such that it captures the mission goals (defined through elicitation from the user) while at the same time being concise and intuitively understandable to the user?
- If there is a distinction between local and global knowledge, how - and when - should this information be synchronized? Additional information can be used to understand the reasoning behind a proposed plan but it is not immediately clear which information has the most potential to that end. The question therefore is: how to approach the trade-off between the cost of sharing the information and the benefit that the sharing of some specific information can bring.
- What insights can be used to improve the functionality and performance of a MAS in dynamic environments? If replanning is required, when should this be triggered and what is the time limit the system has for coming up with an answer? For example, improvements could come from a deeper understanding of the specific environment and domain or from mathematical analysis of the solution space and the negotiation algorithms.

Acknowledgement

The work described in this paper was funded by the Ministry of Defense of the Netherlands. The authors wish to acknowledge in particular Commander Ton van Heusden and Nine Badon Ghijben from the Ministry of Defense and Hanno Hildmann of TNO, as well as their many colleagues that contributed to various Manning and Automation projects invaluably, be it through content or critical discussions.

References

Ai, G., Deng, Y., 2017. An overview of prospect theory in travel choice behavior under risk, in: 2017 4th International Conference on Transportation Information and Safety (ICTIS), pp. 1128–1133. doi:10.1109/ICTIS.2017.8047912.

Arazy, O., Woo, C.C., 2002. Analysis and design of agent-oriented information systems. Knowl. Eng. Rev. 17, 215–260. URL: https://doi.org/10.1017/S0269888902000450, doi:10.1017/S0269888902000450.

Benaskeur, A., Bossé, E., Blodgett, D.E., 2007. Combat Resource Allocation Planning in Naval Engagements. Defence R&D Canada – Valcartier Technical Report TR 2005-486. DRDC Valcartier.

De Gier, J.M., Bakker, B.J.H., Hanckmann, P., Lomme, M., Rieter-Barrell, Y., Janssen, J.A.A.J., 2019a. Utility-Based Resource Management. Technical Report TNO 2018 R11551. TNO. The Hague.

De Gier, J.M., Nijsten, L.M.J., Bakker, B.J.H., Knegjens, R.J., Janssen, J.A.A.J., 2019b. Multi-manager Planning. Technical Report TNO 2018 R11643. TNO. The Hague.

De Groot, T.H., 2015. Mission-driven Resource Management for Reconfigurable Sensing Systems. Ph.D. thesis. TU Delft, Electrical Engineering, Mathematics and Computer Science.

De Groot, T.H., Krasnov, O.A., Yarovoy, A.G., 2014. Mission-driven resource allocation based on subjective input with extra level of uncertainty, in: 17th International Conference on Information Fusion (FUSION), pp. 1–7.

De Groot, T.H., Krasnov, O.A., Yarovoy, A.G., 2018. Mission-driven resource management for reconfigurable sensing systems. IEEE Systems Journal 12, 1531–1542. doi:10.1109/JSYST.2016.2599072.

Decker, K., Lesser, V.R., 1995. Designing a family of coordination algorithms., in: ICMAS, pp. 73–80.

Dörner, D., 1996. The Logic of Failure: Recognizing and Avoiding Error in Complex Situations. Merloyd Lawrence book, Basic Books.

Ephrati, E., Rosenschein, J.S., 1994. Divide and conquer in multi-agent planning, in: Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 1), American Association for Artificial Intelligence, USA. p. 375–380.

Grabisch, M., Labreuche, C., 2010. A decade of application of the choquet and sugeno integrals in multi-criteria decision aid. Annals of Operations Research 175, 247–286.

Hildmann, H., 2018. Designing behavioural artificial intelligence to record, assess and evaluate human behaviour. Multimodal Technologies and Interaction 2. URL: https://www.mdpi.com/2414-4088/2/4/63, doi:10.3390/mti2040063.

Janssen, J.A.A.J., 2019. Eindrapportage van Programma V1514 Functionele Integratie van Toekomstige Schepen. Technical Report TNO 2018 R11644. TNO. The Hague.

Janssen, J.A.A.J., 2020. Intelligent cooperation between crew and naval systems, information leaflet research programme v2006.

Kester, L.J.H.M., 2010. Method for Designing Networking Adaptive Interactive Hybrid Systems. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 401–421. URL: https://doi.org/10.1007/978-3-642-11688-9_15, doi:10.1007/978-3-642-11688-9_15.

Kester, L.J.H.M., Ditzel, M., 2014. Maximising effectiveness of distributed mobile observation systems in dynamic situations, in: 17th International Conference on Information Fusion (FUSION), pp. 1–8.

Muñoz, P., R-Moreno, M., Barrero, D., Ropero, F., 2018. Mobar: a hierarchical action-oriented autonomous control architecture. Journal of Intelligent & Robotic Systems doi:10.1007/s10846-018-0810-z.

Simon, H.A., 1955. A behavioral model of rational choice. The Quarterly Journal of Economics 69, 99–118. doi:10.2307/1884852, arXiv:http://qje.oxfordjournals.org/content/69/1/99.full.pdf+html.

Wooldridge, M.J., 2001. Introduction to Multiagent Systems. John Wiley & Sons, Inc., USA.