Policy Support for Autonomous Swarms of Drones

Alan Cullen[1], Erisa Karafili[2], Alan Pilgrim[1], Chris Williams[3], and Emil Lupu[2]

[1] BAE Systems Applied Intelligence Laboratories
alan.pilgrim@baesystems.com
[2] Imperial College London
{e.karafili, e.c.lupu}@imperial.ac.uk
[3] Defence Scientific and Technology Laboratory
cwilliams@dstl.gov.uk

Abstract. In recent years drones have become more widely used in military and non-military applications. Automation of these drones will become more important as their use increases. Individual drones acting autonomously will be able to achieve some tasks, but swarms of autonomous drones working together will be able to achieve much more complex tasks and be able to better adapt to changing environments. In this paper we describe an example scenario involving a swarm of drones from a military coalition and civil/humanitarian organisations that are working collaboratively to monitor areas at risk of flooding. We provide a definition of a swarm and how they can operate by exchanging messages. We define a flexible set of policies that are applicable to our scenario that can be easily extended to other scenarios or policy paradigms. These policies ensure that the swarms of drones behave as expected (e.g., for safety and security). Finally we discuss the challenges and limitations around policies for autonomous swarms and how new research, such as generative policies, can aid in solving these limitations.

Keywords: Swarm; Drone Systems; Policies; Coalitions.

1 Introduction

Drones are being adopted in dull, dirty and dangerous military operations [15] as well as non-military applications such as pipeline inspection, highway monitoring and filming [11]. As drones become accepted and more capable there is emerging interest in the potential of swarms of multiple cooperating drones.

In this work we provide a definition for a swarm of drones. Through the use of a scenario we describe how the drones of a swarm communicate and coordinate with each other to achieve their common and individual goals. The constraints and rules of the swarm are represented using a generic policy language, as a set of policies that regulate the actions and behaviours of the drones of a swarm.

There is increasing interest over the swarm of drones as they rise in popularity and use. DARPA carried out a *capture the flag* trial between opposing teams, each flying dozens of low cost drones, demonstrating both robustness in numbers and the benefits of diversity between fixed wing and quadcopter platforms [1]. Another swarming example is provided by a simulation of data ferrying (i.e. physically transporting data between communicating parties) using drones where policies are used to synchronize the flight of drones that cannot communicate directly with each other [13]. Swarms are a familiar concept in nature, described by collective nouns such as a swarm of insects, a flock of birds, or a school of fish. These swarms have benefits such as protection from predators and a larger effective search pattern in the quest for food [25].

The scenario we introduce in our work commences with a requirement for four drones to observe seven surveillance targets. It is reasonable to assert that in this situation an optimal allocation of drones to specific targets will outperform flocking solutions - where the drones move together in a crowd from one surveillance target to another. The exception is possibly when the probability of attack is so high that the defensive benefits of flocking outweigh a solution that, for example, maximizes the observation time at each target. This is not surprising because surveillance and search have different requirements, and even if drones are deployed on a search mission they may benefit by exploiting capabilities lacking in nature such as long range communications.

A swarm can comprise a number of different components and resources. The swarm needs to make decisions about the tasks to perform, where to fly, as well as to coordinate with the other components during the flight and to communicate with other components or the leader of the swarm. This decision process is made depending on the constraints and rules already defined to the swarm, as well as the environmental/mission conditions where the swarm is operating. In our work we introduce a set of policies that represents these constraints and rules and are used by the swarm decision process.

1.1 Related work

Lately, research focused on unmanned aircrafts (e.g., drone systems) is increasing, especially with the expansion of their usage (e.g., drone package delivery like *Amazon's Prime Air* [2] or *Google's Project Wing* [29]). Different studies have been made by Amazon [3,4], Google [14] and NASA [23] concerning the safety and efficiency of design, management, operations of unmanned aircraft systems (UAS), their safe airspace access, and their communications and collaborations. The authors in [12] present a model of architecture for coordinating the access of UAS to controlled airspace and for providing navigation services between interested locations. The increasing autonomy on the airborne drones in joint collaborative operations between different parties and their impact is analysed in [10].

Deciding the tasks to perform and the areas where to fly can be seen as a planning problem. In this work we are not dealing with the planning process performed by the swarm of drones. Therefore, we will not solve any optimality problems related to the decision process made by the swarms. For further details regarding the planning and optimality problem we refer the reader to [26,28] where these problems are solved using techniques taken from artificial intelligence and automatic control [8]. Our aim is instead, to represent the various options and behavioural rules of the swarms with the use of a generic policy language.

An important challenge that arrises during the coordination and planning phases of drone systems, especially in collaborative scenarios, is the decision process of applicable actions for particular cases. A policy analysis for drone systems is developed in [19] that is able to capture and solve conflicting rules, and improve the efficiency of the used set of rules, based on argumentation and abductive reasoning [16], in particular it uses preference-based argumentation [6,24,5]. The work in [19] does not deal with the notion of swarm of drones and their operation. This analysis is an extension of the one proposed in [18,21], that is used for enabling data sharing in different contexts by enforcing the correct data sharing agreements, and during forensics investigations for attributing cyber attacks to attackers [17]. Another interesting technique that uses an argumentation based analysis is introduced in [22], where the authors present a method for goal conflict resolution by analysing competing hypotheses and beliefs of stakeholders.

Section 2 presents a use case taken from a military scenario for a swarm of drones. Section 3 provides a definition and model of a swarm and describes as well how this swarm can operate. Section 4 provides example policies relevant to our scenario and how this set of policies controls the behaviour of the drones. Section 5 discusses the challenges and limitations in generating ad-hoc policies for the scenario and how new research can aid in solving these limitations. Finally we present the conclusions in Section 6.

2 Introduction to the Scenario: Collaborative Emergency Area Monitoring between Military and Civil Organization

A military coalition has been formed to provide support to the government of a country that is suffering from severe flooding and whose resources (e.g., drones and aircraft) are occasionally attacked by rebels. There are two coalition partners (nations N_1 and N_2) and they each operate drones from their *forward operating bases* (FOBs). In addition *civil/humanitarian organisations* have allowed the coalition to command their drones that are operated from sites HN_1 and HN_2 as shown in Figure 1.

The map shows the operating bases/sites, *emergency landing sites* (E_1, E_2, $\cdots$, E_5), and *no-fly-zones* that have been assigned to protect the health and safety of citizens living in clusters of towns in the south. The main flooding is in the west where there are a number of industrial plants that will cause pollution if the flood defences fail. Hence a set of surveillance *targets* (T_1, T_2, $\cdots$, T_7) has been established in order to monitor the water level of the rivers and lakes that are at risk of flooding, and the corresponding flood defences. Monitoring these areas will allow appropriate action to be taken at the industrial plants, such as shutting them down, depending on the likelihood of flooding.

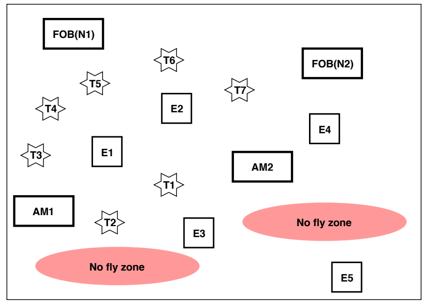

Fig. 1. Initial scenario representation

The commander handling the flooding emergency tasks a swarm of drones, formed from the coalition nations and civil/humanitarian organisations, to commence surveillance of the seven targets. The *lead drone* (D_1) is allocated three[4] further drones (D_2, D_3 and D_4), and after planning assigns reconnaissance targets to the drones as shown in Figure 2.

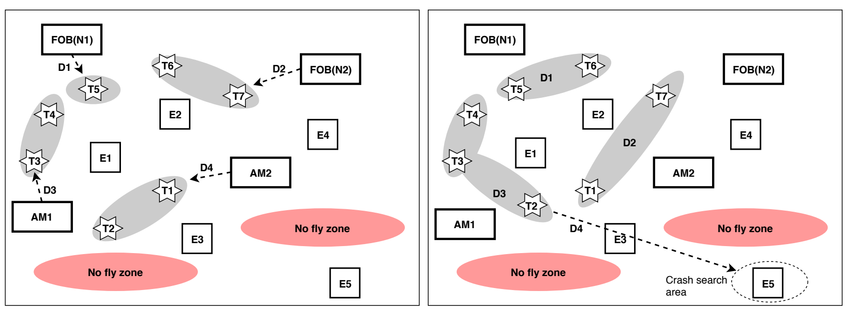

Fig. 2. Left: Reconnaissance targets assignation between the drones of a swarm. Right: Targets re-assignation after the emergency request made to D_4.

Some time later drone D_4 receives a weak *emergency* radio signal from a coalition manned aircraft saying that it has crashed and that the crew have survived but need assistance. The navigation equipment has failed and the location is described as "near E_5". D_4 authenticates the signal, reports it to the lead drone and decides autonomously to commence a search mission for the lost aircraft. The lead drone reassigns the surveillance roles and reports to the commander. The new state of the swarm is shown in Figure 2.

Nation 1, (N_1), has a spare drone (D_5) and allocates it to the swarm, and the lead drone updates the plan. D_4 continues flying towards the search area,

[4] For the sake of simplicity, during the description of the scenario, we use a small number of drones.

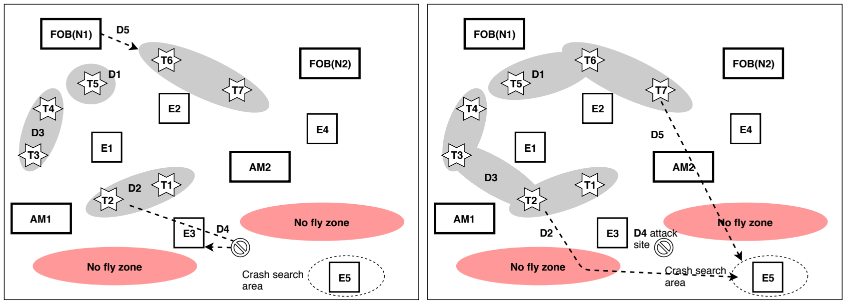

Fig. 3. Left: Target assignation update after the addition of D_5 into the swarm and D_4 emergency landing in E_3. Right: Replanning tasks assignation to respond the D_4's emergency.

but comes under attack from suspected rebels and has to make an emergency landing, selecting site E_3, as shown in Figure 3.

The lead drone replans the search activity in response to the emerging situation of D_4. Given the importance of the search and a risk assessment based on the health of the drones, the lead drone assigns the two nearest drones to the search task (i.e. D_2 and D_5), requesting them to transit the no-fly-zones in order to both minimise delays and avoid the probable location of the rebels that are believed to have attacked D_4 as shown in Figure 3.

Drones D_2 and D_5 belong to different coalition nations (N_2 and N_1 respectively) and have different policies for transiting no-fly-zones. The policy assigned to D_2 allows transits over no-fly-zones if the primary motivation for the zone is to *protect the local population* and if the task is of sufficient importance. In contrast D_5 requires a senior officer from its own nation to authorise the violation.

In the coming section, we introduce a formal definition of swarm of drones, how they operate and a set of policies that regulate the behaviour of the swarm of drones of the above described scenario.

3 Definition and Operation of a Swarm

Despite growing literature about examples of swarming by drones, to the best of our knowledge there is not any fundamental work on the concept of swarming. In this section, we propose a definition and then discuss a model for swarming taking examples from the scenario in Section 2.

3.1 Definition of a swarm of drones

Let us first give a definition of a swarm.
Definition: *A swarm is a collection of assets that operate in a collaborative way to achieve a set of goals.*

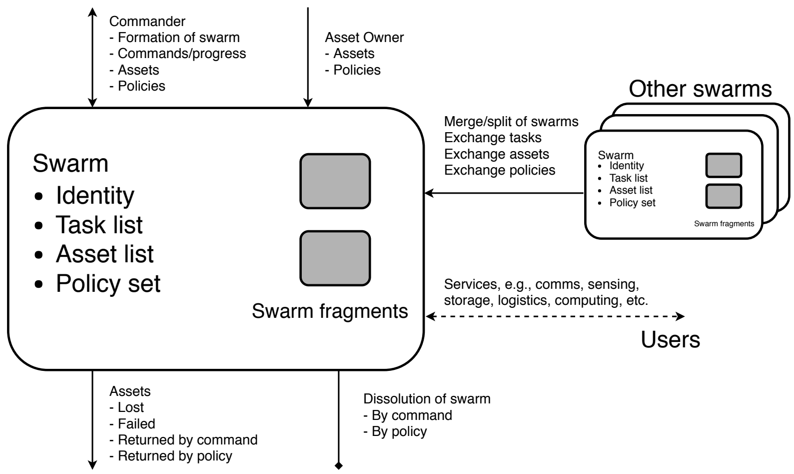

Fig. 4. The representation of the model of a swarm.

This definition does not constrain swarm membership to a homogenous set of drones. The swarm in our example scenario consists of military and civil/ humanitarian drones so it is likely to be heterogeneous. Indeed, a swarm may be a mix of drones and land or maritime vehicles. We view a swarm primarily as a collaborative concept, and it may be composed of any assets that benefit from collaboration. Hence a swarm should be of the right size; too large and it becomes difficult to learn the appropriate individual behaviours to avoid unwanted and emergent behaviours. The assets are also a limited resource.

In our scenario the swarm had a *lead* drone that carried out the planning, although distributed control is an alternative approach. For example, the swarm members could work independently, reacting to the tasks the other members are performing to achieve the overall set of goals. The ASIMUT project [7] provides a further example of distributed control in a search mission where drones work collectively to cover the search space and the need to communicate is minimized by using predictable pseudo random paths. If a swarm has a leader there are varying levels of autonomy for the other members. At one extreme the leader could provide detailed requests to the member which would limit its automation (e.g., fly to a particular location, by using a certain route and at a specified speed and altitude). Alternatively, the requests can be at a much higher command level, allowing greater autonomy for the member to achieve its task (e.g., travel to and monitor this location).

As our scenario shows, a swarm is dynamic. It is *created* when required and achieves a set of goals that can be dynamic as well. A swarm *terminates* when the goals have been achieved, cancelled or become impractical. The *membership* of the swarm is dynamic; assets may be added, removed or fail, however, there will be finite number of assets that could be added to the swarm. Each asset has an owner, and as our scenario shows a swarm may have several asset owners.

The diagram in Figure 4 shows a model of a swarm. It has an identity, tasks assigned by the commander, assets, and policies. The swarm will eventually be dissolved, and it may become fragmented as a result of communications limitations, a possibility considered in our scenario.

Although our scenario has a single swarm, there may be several swarms. Communication between swarms is possible if they have interoperable equipment, and may be on a planned or opportunistic basis to exchange tasks, assets and policies. Moreover, to ensure swarms have a near optimum size as their objectives change, they may split or merge.

The swarm in our scenario carried out surveillance and search tasks. Other tasks involving drones include environment sensing; crop spraying; searching for an emitter; provision of computing, storage or communications services; logistics delivery; decoy; protective convoy and acrobatic displays. It may be beneficial if a swarm undertakes several tasks concurrently, as our scenario illustrates.

3.2 Operating swarms

In this section we introduce an example of how a policy enabled swarm could operate by exchanging messages between the entities. The messages have been designed for research purposes to show a step by step swarming sequence, and are not intended as an engineered solution. Moreover there are a number of simplifications made for brevity:

- Each swarm is controlled by a lead asset that is not lost, e.g. no provision is made for the appointment of a deputy leader with failover.
- Fully decentralised operation, without a lead asset, is not supported. However the assets can autonomously react to certain situations.
- All messages include an acknowledgement which is received successfully in the scenario under consideration.
- There is interoperability of command, policy and messaging between the civil/humanitarian and military drones. In practice, the communication between the different nations and civil/humanitarian drones is likely to be performed through the nation bases rather than directly from drone to drone.

In Figure 5 we represent the message sequence diagram that corresponds to the scenario in Section 2. In detail, operation commences at time T_1 (times are shown to the left of the diagram) when the commander sends a $FormSwarm$ message to appoint D_1 as the lead drone. Next, at time T_2, the drone owners allocate drones D_2, D_3 and D_4 to join the swarm using $JoinSwarm$ messages. Note that the communications between the commander and the drone owners is not shown, it may for example have been verbal via military liaison officers.

The commander uses a $MissionCmd$ message at time T_3 to task the swarm with surveillance. The lead drone plans the operation and in turn assigns roles to each drone using $MissionCmd$ messages. The drones take off and fly to their surveillance areas (T_4), and commence surveillance when they arrive. Progress

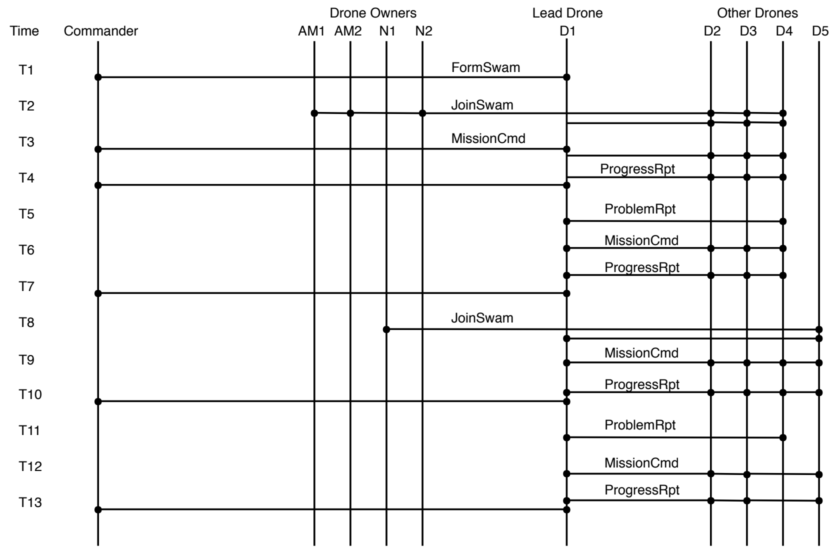

Fig. 5. The message sequence diagram of the scenario in Section 2.

is reported using a series of *ProgressRpt* messages (only the first message is illustrated). Surveillance reports will also be sent[5].

A manned coalition aircraft crashes near E_3 and the emergency signal is received by drone D_4 at time T_5. D_4 autonomously commences a search task and informs the lead drone using a *ProblemRpt* message. The lead drone replans the mission at time T_6, with the surveillance now carried out by drones D_1, D_2 and D_3, and the search carried out by D_4. Mission progress continues to be reported (T_7) and drone owner N_1 allocates drone D_5 to join the swarm at T_8. Once again the lead drone replans the mission, and sends the revised plans to all drones (T_9). On this occasion there is no change to the D_4 search plan. Mission progress continues to be reported (T_{10}).

Drone D_4 is attacked at time T_{11} and needs to make an emergency landing, selecting site E_3. This is reported to the lead drone using a *ProblemRpt* message. The lead drone replans the surveillance and search missions (T_{12}), with drones D_1 and D_3 carrying out modified surveillance roles and drones D_2 and D_5 commencing the search activity. Mission progress continues to be reported (T_{13}).

4 Policies to Support Autonomous Swarms

In this section, we introduce a policy representation of the rules that control the behaviour of the drones that are part of a swarm. In particular, we repre-

[5] For the sake of readability, we decided to do not illustrate the surveillance reports in the diagram.

sent, through the use of a policy language, the rules of the scenario presented in Section 2. The policy language used is a semi-natural language that extends an existing policy language [9], where on the right hand side we state the conditions that should hold for the left hand side action to be true. This language is based on an event-condition-action paradigm. Our introduced policy language is a generic one which can be easily extended and adapted to more specific paradigms such as generative policies [27]. Generative polices are policies that can be dynamically refined and adapted by the individual drone, for example, to suit the characteristic of the drone or the task it is currently performing.

The main actors of the above described scenario are:

- the drones involved in the scenario: $D_1, D_2, \cdots, D_5$;
- the swarm where the drones are part of: SW;
- the countries that own the drones: N_1 and N_2;
- the operating bases of the two countries: $FOB(N_1)$ and $FOB(N_2)$;
- the emergency landing areas: $E_1, \cdots, E_5$;
- other generic entities denoted by C.

We denote the relation of being part of a swarm with: $D \in SW$, where this predicate describes that drone D is part of swarm SW. The property of being the *lead drone* in a swarm is denoted by $Leader(SW)$, where $D = Leader(SW)$ represents that drone D is the leader of swarm SW. As described in the previously, we expect to have just one lead drone for a swarm. Being the leader implies being part of the swarm as well, $D = Leader(SW) \rightarrow D \in SW$. For the sake of simplicity, we denote the lead drone by L instead of D. In case a given entity C is not part of the swarm, then this entity cannot be its leader: $C \notin SW \rightarrow C \neq Leader(SW)$. We denote by $req(Sub, Tar, Act, T)$ the predicate that represents sending a request at the instant of time T from the subject Sub to the target Tar for performing a certain action Act. The predicate $perform(Sub, Act, T, permit)$ describes that subject Sub is permitted to start carrying out the action Act, and this permission is given at the instant of time T. Once the perform predicate is permitted then Sub will start performing the granted action:

$$do(Sub, Act, T) \leftarrow perform(Sub, Act, T, permit). \qquad (1)$$

For the sake of simplicity, we omit this predicate in the future and every time a perform predicate is permitted we understand that the action will start to be executed on the same instant of time. In the scenario a drone/swarm is requested to perform a certain $Task$ that can be accepted or denied. Therefore, the perform predicate will be of the form: $perform(Sub, Task, T, permit/deny)$.

Given the drones that are part of a swarm, they are permitted to start carrying out actions that are given/requested just from their lead drone. The drones do not perform the tasks requested from other drones that are not the nominated swarm leader[6]. These rules are represented as below:

$$\begin{aligned} perform(D, Task, T, permit) \leftarrow\ & D \in SW,\ L \in SW,\ L = Leader(SW), \\ & req(L, D, Task, T). \end{aligned} \tag{2}$$

$$\begin{aligned} perform(D, Task, T, deny) \leftarrow\ & D \in SW, L \neq Leader(SW), \\ & req(L, D, Task, T). \end{aligned} \tag{3}$$

In the case there is an *emergency* where humans of a coalition are involved and the drones receive a request to perform a certain task from them, then the drones accept the requested action, as described in the following rule.

$$\begin{aligned} perform(D, Task, T, permit) \leftarrow\ & Emerg(C),\ req(C, D, Task, T),\ D \in SW, \\ & C \neq Leader(SW),\ C \in Coalition(D), \\ & Human(C). \end{aligned} \tag{4}$$

In the above rule, given an entity denoted by C, that makes a request to a certain drone D to perform a task, where this entity is in an emergency situation $Emerg(C)$, it is a human $Human(C)$, in a coalition relation with D, denoted by $Coalition(D)$, and is not part of the swarm where D is part of, therefore not the swarm leader, then the drone is permitted to start performing the requested task. This permission is given as the request came from a human that is a coalition member and is in an emergency. Note that we define the set of all entities that are in coalition with a given subject S by $Coalition(S)$. Thus, we use the relation $\in$ for denoting when a certain object O is in coalition with S, e.g., $O \in Coalition(S)$, means that object O is part of the same coalition as S.

When the drone accepts an action from an entity that is not its leader, then it sends a notification ($Emerg\_N$) to its leader as well[7].

$$\begin{aligned} send(D, L, Emerg\_N, T, permit) \leftarrow\ & L = Leader(SW), D \in SW, \\ & C \neq Leader(SW), req(C, D, Task, T), \\ & perform(D, Task, T, permit). \end{aligned} \tag{5}$$

The task given to the drone is composed of the reconnaissance targets that are translated into the areas where the drone should fly to reach the targets. The predicate $PermArea(Task, Coalition(D))$ holds when all the areas to be passed by drone D, to perform the task $Task$, are permitted flying areas for the coalition where drone D is part of, $Coalition(D)$, and there is not any no-fly-zone. In the case in the given $Task$ there are not any no-fly-zone, then the drone is permitted to fly and perform the task.

$$\begin{aligned} fly(D, Task, T, permit) \leftarrow\ & perform(D, Task, T, permit), \\ & PermArea(Task, Coalition(D)). \end{aligned} \tag{6}$$

[6] We assume that we deal with one request for performing an action at a time, and in the case the leader requests an action to a drone while it is still performing a previous one, then it drops the previous action and starts the new one.

[7] To be precise, rules (5)-(10) would not have $perform(D, Task, T, permit)$, but $do(D, Task, T)$. As explained previously, we decided for the sake of understandability to omit the *do* predicate, and use this short-hand annotation.

In the case there is at least one no-fly-zone area included in the requested task, then the drone is not permitted to fly for performing that task.

$$\begin{aligned} fly(D, Task, T, deny) \leftarrow\ & perform(D, Task, T, permit), \\ & \textbf{not}\ PermArea(Task, Coalition(D)). \end{aligned} \tag{7}$$

In the case the drones need to fly into no-fly-zones, then there are different rules for permitting it, and they vary from the drones' owner. The drones of nation N_2 can fly into no-fly-zones in case there is an emergency involving humans.

$$\begin{aligned} fly(D, Task, T, permit) \leftarrow\ & D \in SW, D \in N_2, Emerg(C), C \in Coalition(D), \\ & Human(C),\ perform(D, Task, T, permit), \\ & \textbf{not}\ PermArea(Task, SW). \end{aligned} \tag{8}$$

The rule is different for drones of country N_1. For permitting drones of N_1 to fly into no-fly-zones, an authorization request, $auth\_req$ to the FOB of N_1 for $Task$ needs to be generated. In the case N_1 accepts this request, then an authorization is issued for the drone to fly into the non permitted area.

$$\begin{aligned} auth\_req(FOB(N_1), D, Task, T) \leftarrow\ & D \in SW, D \in N_1, \\ & perform(D, Task, T, permit), \\ & \textbf{not}\ PermArea(Task, SW). \end{aligned} \tag{9}$$

$$\begin{aligned} fly(D, Task, T, permit) \leftarrow\ & perform(D, Task, T, permit), D \in SW, D \in N_1, \\ & \textbf{not}\ Perm\_Area(Task, SW), \\ & Auth(FOB(N_1), D, Task, permit). \end{aligned} \tag{10}$$

Rule (7) is in conflict with rule (8) and (10). In our scenario rules (8) and (10) prevails over rule (7), as they are more specific. For solving the conflicts between policies we can use an abductive and argumentation reasoning for conflict resolution as in [18,21].

If the drone finds itself in an emergency situation, then it is permitted to land in a designated emergency area allocated to its swarm. In this case, the drone notifies its leader with an emergency landing notification.

$$land(D, E, T, permit) \leftarrow Emerg(D), D \in SW, E \in Emerg\_Area(SW). \tag{11}$$

$$send(D, L, EL\_N, T, permit) \leftarrow Emerg(D), D \in SW, L = Leader(SW). \tag{12}$$

In the above case, E is one of the emergency areas of the given swarm ($Emerg\_Area(SW)$), and the notification sent to the leader of the swarm is denoted by EL_N.

5 Scenario and Policy Discussion

In this section we consider some variants of the scenario described in Section 2 to highlight the challenges in generating and using policies for autonomous swarms.

5.1 Communications between drones

The first discussion is around communications between drones. Drone D_4 was the first asset to receive the emergency signal from the crashed aircraft and on the basis of this authenticated signal D_4 decided autonomously to commence a search operation, and also reported the situation to the lead drone. This report was critical because D_4 subsequently came under attack; what if D_4 had been unable communicate with the lead drone due to the contested communications, and the subsequent attack had damaged the drone and it had failed to reach the emergency landing site?

If D_4 was the only asset to receive the emergency signal, and D_4 was damaged before it had communicated the news, then the mission would be delayed until the emergency was detected in other ways, e.g., a timeout on arrival or after failing to submit or respond to agreed communications. This may increase the risk to the survivors of the crash. If on the other hand, D_4 knew/suspected that it was the only asset to receive the emergency signal then policies could be used to guide the drone to position itself to increase the probability of communicating with the coalition. This decision is a trade off between delaying the search and the risk that the knowledge of the crash is lost.

This example highlights the possible conflicts between policies such as rules (8) and (10) with (7), in Section 4, and clearly in any policy system a strategy for dealing with conflicts is required. A proposed solution would be to use the argumentation and abductive reasoning procedure introduced in [18,21], and more specifically for drones [19], where the rules are left as they are, and priorities between them are introduced. The priority relation between rules is denoted by $\prec$, where $r_1 \prec r_2$ means that rule r_2 takes priority over rule r_1. In our scenario, we have that rule (8) prevails over rule (7), and rule (10) prevails over rule (7), denoted correspondingly by $(7) \prec (8)$ and $(7) \prec (10)$.

5.2 Policy sharing

The second discussion point is around sharing of policies. In the scenario we have drones from civil/humanitarian organisations and from military organizations of two nations that have different policies for transiting the no-fly-zones. The lead drone (D_1) is part of nation N_1, so therefore should have access to the policies for other drones that are part of the same nation, such as D_5, who it instructs to transit a no-fly-zone in order to respond to the crashed aircraft. However, the lead drone (D_1) also instructs D_2 from N_2 to respond and transit the no-fly-zone. If the polices are shared between the two nations, then the lead drone can successfully plan the tasks. However sharing policies with another nation could present a security risk.

In our scenario we are working with two types of drones, military and civil/humanitarian organizations. The rules for sharing information and data can be different for both of them, e.g., more restrictive from the military drones side. For future work in managing the different behavioural rules between the two types of drones we suggest to use the policy analysis introduced in [19]. In the case there is a need of performing data alteration, or restriction of data quality when information is shared between drones that have different owners and different relations between each other, we propose to use the data alteration mechanism introduced in [20]. The proposed alteration mechanism is based on the same policy language we extend in this work, and is attached to the data in a similar mechanism as the sticky policies.

One of the limitations with existing policy based architectures is a lack of flexibility and customisation of the polices at the individual drone. This limits the automation of the drone to adapt the polices to its own characteristics or to a changing environment. A generative policy architecture [27] aims to solve this by allowing an initial policy specification called a generative policy to be distributed to all drones, but allowing it to be adapted at the individual drone, a simple example for drone systems is discussed in [10].

6 Conclusions

In this paper we have defined a realistic scenario that can be used to investigate and trial autonomous swarms of drones. In summary a military coalition of drones from two nations and humanitarian/civil organisations work together to monitor areas prone to natural disasters, such as flooding. In this area vehicles are suspect to occasional attacks from rebels, and while monitoring an area of interest a drone receives a distress signal from a coalition aircraft that it needs to investigate and then itself comes under attack. Throughout this the drones need to work together and react to the changing situation in order to complete their existing monitoring tasks and the additional high priority search and rescue incidents.

We have defined a swarm as a *collection of assets that operate in a collaborative way to achieve a set of goals* and provided a model for swarm. The model identifies that a swarm has an identity, task list, asset list and a policy set. It also allows assets to join or leave the swarm, and interacts with assess owners, commanders and other swarms. An example of how a policy enabled swarm can operate has been provided, assuming that it is controlled by a leader.

A set of polices have been defined to control the behaviour of the drones. These policies cover areas such as whether the drones can accept the given task and if they can transit no-fly-zones. We have introduced a policy language using a semi natural language approach based on the event-condition paradigm. This language can easily be extended and adapted to more specific paradigms.

We have then discussed the challenges the scenario introduces and the limitations of existing policy based solutions. We then pull together some interesting future work that can be done by using generative polices and approaches for communications between coalitions that can improve and perhaps solve these limitations.

Acknowledgments

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

1. Darpa: Service academies swarm challenge live-fly competition. `https://www.youtube.com/watch?v=RZ-CKA4fUhg` (2017)
2. Amazon.com Inc: Amazon Prime Air. `https://www.amazon.com/b?node=8037720011`
3. Amazon.com Inc: Determining safe access with a best-equipped, best-served model for small unmanned aircraft systems. `https://images-na.ssl-images-amazon.com/images/G/01/112715/download/Amazon_Determining_Safe_Access_with_a_Best-Equipped_Best-Served_Model_for_sUAS.pdf` (2015)
4. Amazon.com Inc: Revising the airspace model for the safe integration of small unmanned aircraft systems'. `https://images-na.ssl-images-amazon.com/images/G/01/112715/download/Amazon_Revising_the_Airspace_Model_for_the_Safe_Integration_of_sUAS.pdf` (2015)
5. Amgoud, L., Dimopoulos, Y., Moraitis, P.: Making decisions through preference-based argumentation. In: Brewka, G., Lang, J. (eds.) KR. pp. 113–123. AAAI Press (2008)
6. Bondarenko, A., Dung, P.M., Kowalski, R.A., Toni, F.: An abstract, argumentation-theoretic approach to default reasoning. Artificial intelligence **93**(1-2), 63–101 (1997)
7. Bouvry, P., Chaumette, S., Danoy, G., Guerrini, G., Jurquet, G., Kuwertz, A., Müller, W., Rosalie, M., Sander, J., Segor, F.: ASIMUT project: Aid to SItuation Management based on MUltimodal, MUltiUAVs, MUltilevel acquisition Techniques. In: Proceedings of the 3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications. pp. 17–20. ACM (2017)
8. Cortes, J., Martinez, S., Karatas, T., Bullo, F.: Coverage control for mobile sensing networks. In: IEEE International Conference on Robotics and Automation (ICRA'02). vol. 2, pp. 1327–1332. IEEE (2002)
9. Craven, R., Lobo, J., Ma, J., Russo, A., Lupu, E.C., Bandara, A.K.: Expressive policy analysis with enhanced system dynamicity. In: Proceedings of the 2009 ACM Symposium on Information, Computer and Communications Security, ASIACCS. pp. 239–250 (2009)
10. Cullen, A., Williams, B., Bertino, E., Arunkumar, S., Karafili, E., Lupu, E.: Mission support for drones: a policy based approach. In: Proceedings of the 3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications, DroNet@MobiSys 2017. pp. 7–12 (2017)

11. (EASA), E.A.S.A.: Concept of operation for drones: a risk based approach to regulation of unmanned aircraft. https://www.easa.europa.eu/sites/default/files/dfu/204696_EASA_concept_drone_brochure_web.pdf (2015)
12. Gharibi, M., Boutaba, R., Waslander, S.L.: Internet of drones. IEEE Access **4**, 1148–1162 (2016)
13. Hunjet, R., Stevens, T., Elliot, M., Fraser, B., George, P.: Survivable communications and autonomous delivery service a generic swarming framework enabling communications in contested environments. In: IEEE Military Communications Conference (MILCOM). pp. 788–793 (2017)
14. Inc, G.: Google UAS Airspace System Overview. https://utm.arc.nasa.gov/docs/GoogleUASAirspaceSystemOverview5pager[1].pdf (2015)
15. Joint Doctrine Note 2/11: The UK approach to unmanned aircraft systems (March 2011)
16. Kakas, A.C., Kowalski, R.A., Toni, F.: Abductive logic programming. Journal of logic and computation **2**(6), 719–770 (1992)
17. Karafili, E., Kakas, A.C., Spanoudakis, N.I., Lupu, E.C.: Argumentation-based security for social good. AAAI Fall Symposium Series pp. 164–170 (2017)
18. Karafili, E., Lupu, E.C.: Enabling data sharing in contextual environments: Policy representation and analysis. In: Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies, SACMAT. pp. 231–238 (2017)
19. Karafili, E., Lupu, E.C., Arunkumar, S., Bertino, E.: Argumentation-based policy analysis for drone systems. In: 2017 IEEE SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI. pp. 1–6 (2017)
20. Karafili, E., Lupu, E.C., Cullen, A., Williams, B., Arunkumar, S., Calo, S.B.: Improving data sharing in data rich environments. In: IEEE International Conference on Big Data, BigData 2017. pp. 2998–3005 (2017)
21. Karafili, E., Spanaki, K., Lupu, E.C.: An argumentation reasoning approach for data processing. Computers in Industry **94**, 52–61 (2018)
22. Murukannaiah, P.K., Kalia, A.K., Telangy, P.R., Singh, M.P.: Resolving goal conflicts via argumentation-based analysis of competing hypotheses. In: IEEE 23rd International Requirements Engineering Conference (RE). pp. 156–165 (2015)
23. NASA: NASA UTM 2015: The next era of aviation. https://utm.arc.nasa.gov/utm2015.shtml (2015)
24. Prakken, H., Sartor, G.: Argument-based extended logic programming with defeasible priorities. J. of Appl. Non-Class. Log. **7**(1), 25–75 (1997)
25. Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. In: ACM SIGGRAPH computer graphics. vol. 21, pp. 25–34. ACM (1987)
26. Steve, O.E., Hanks, D.W., Draper, D.: An approach to planning with incomplete information. In: Third International Conference Principles of Knowledge Representation and Reasoning (KR'92). p. 115 (1992)
27. Verma, D.C., Calo, S.B., Chakraborty, S., Bertino, E., Williams, C., Tucker, J., Rivera, B.: Generative policy model for autonomic management. In: 2017 IEEE SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI. pp. 1–6 (2017)
28. Weld, D.S.: Recent advances in ai planning. AI magazine **20**(2), 93 (1999)
29. X, G.: Project Wing. https://x.company/wing/