Managing Intelligence Resources Using Semantic Matchmaking and Argumentation

Alun Preece, Timothy J. Norman, Mario Gomez and Nir Oren

Abstract. Effective deployment and utilisation of limited and constrained intelligence, surveillance and reconnaissance (ISR) resources is seen as a key issue in modern network-centric joint-forces operations. In this chapter, we examine the application of semantic matchmaking and argumentation technologies to the management of ISR resources in the context of coalition operations. We show how ontologies and reasoning can be used to assign sensors and sources to meet the needs of missions, and we show how argumentation can support the process of gathering and reasoning about uncertain evidence obtained from various sources.

1. Introduction

Effective deployment and utilisation of limited and constrained intelligence, surveillance and reconnaissance (ISR) resources is seen as a key issue in modern network-centric joint-forces operations. For example, the 2004 report *JP 2-01 Joint and National Intelligence Support to Military Operations* states the problem in the following terms: "ISR resources are typically in high demand and requirements usually exceed platform capabilities and inventory. ... The foremost challenge of collection management is to maximise the effectiveness of limited collection resources within the time constraints imposed by operational requirements."[1]

Our work focuses upon the application of Virtual Organisation technologies to manage coalition resources. In the past we have shown an agent-based VOs can manage the deployment and utilisation of network resources in a variety of domains, including e-business, e-science, and e-response [1, 2]. Two distinguishing features of our work are (1)

This research was sponsored by the US Army Research Laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the US Government, the UK Ministry of Defence or the UK Government. The US and UK Governments are authorised to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

[1] http://www.dtic.mil/doctrine/jel/new_pubs/jp2_01print.pdf, pages III–10–11, accessed April 27, 2007.

the use of semantically-rich representations of user requirements and resource capabilities, to support matchmaking using ontologies and reasoning, and (2) the use of argumentation to support negotiation over scarce resources, decisions about which resources to use, and the combining of evidence from information-providing resources (e.g. sensors).

In this chapter, we examine the application of (1) and (2) to the management of ISR resources in the context of coalition operations. The first part of the chapter describes an ontology-based approach to the problem of assigning sensors and sources to meet the needs of missions. The second part then looks at how argumentation and subjective logic can facilitate the process of gathering uncertain evidence through actions collectively referred to as sensor probes, and combining that evidence into a set of arguments in support of, and in opposition to, a particular decision.

Our applications involve agents that must cooperate, but still try to maximise their individual utilities, possibly to the detriment of other agents in the system. This type of scenario often appears in military settings, including within coalition operations. Each member of the coalition requires certain assets — including physical assets such as materiel (personnel, vehicles, equipment, etc), and information assets including various forms of intelligence — to achieve their mission, but these assets are oversubscribed. By advancing arguments as to why they should have the assets, the coalition members may make their own missions more easy to achieve. However, they might have to gather additional information so as to be able to justify their arguments, thus introducing some form of utility cost.

2. Semantic Matchmaking of Sensors and Missions

The assignment of ISR assets to multiple competing missions can be seen as a process comprising two main steps: (1) assessing the fitness for purpose of alternative ISR means to accomplish a mission, and (2) allocating available assets to the missions. Our work draws upon current military doctrine, specifically the Missions and Means Framework (MMF) [3] which provides a model for explicitly specifying a *mission* and quantitatively evaluating the utility of alternative warfighting solutions: the *means*.

Figure 1 shows how missions map to ISR means. Starting from the top left the diagram sketches the analysis of a mission as a top-down process that breaks a mission into a collection of operations (e.g. search-and-rescue), each of which is broken down further into a collection of distinct tasks having specific capability requirements (e.g. wide-area surveillance). On the right hand side, the diagram depicts the analysis of capabilities as a bottom-up process that builds up from elementary components (e.g. electro-optical/infrared (EO/IR) camera) into systems (e.g. camera turret), and from systems up into platforms equipped with or carrying those systems (e.g. an unmanned aerial vehicle (UAV)).

The way MMF describes the linking between missions and means naturally fits the notion of matchmaking. Matchmaking is basically the process of discovering, based on a given request (e.g. ISR requirements), promising partners/resources (e.g. sensors) for some kind of purpose (e.g. accomplishing a mission). Important issues arise when the

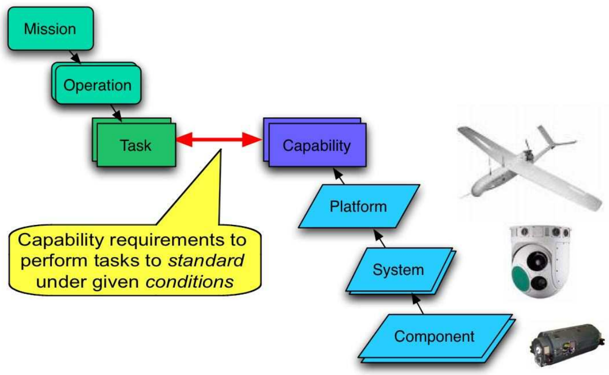

FIGURE 1. Overview of the Mission and Means Framework (MMF)

search is not limited to identity matches but, as in real life, when the objective is finding partners/resources suitable at least to some extent, or (when a single partner cannot fulfil the request) to find a pool of cooperating partners (a sensor network, or a platform equipped with several sensors) able to accomplish it. As this process may lead to various possible matches, the notion of ranking becomes central: to provide a list of potential partners ordered according to some criteria. Due to the diversity of frameworks of application, several communities have studied matchmaking through perspectives and techniques. Recently, semantic matchmaking, which is based on the use of ontologies [4] to specify components, has become a central topic of research in many communities, including multi-agent Systems, Web services and Grid computing.

In particular, we propose the use of ontologies to support the following activities:

- specifying the requirements of a mission;
- specifying the capabilities provided by ISR assets (sensors, platforms and other sources of intelligence, such as human beings);
- comparing — be a process of automated reasoning —the specification of a mission against the specification of available assets to either decide whether there is a solution (a single asset or combination of assets) that satisfies the requirements of a mission, or alternatively providing a ranking of solutions according to their relative degree of utility to the mission.

2.1. Ontologies for matchmaking

People, organisations and software systems need to communicate and share information, but due to different needs and background contexts, there can be widely varying viewpoints and assumptions regarding what essentially the subject matter is. The lack of shared understanding leads to poor communication between people and their organisations, severely limits systems interoperability and reduces the potential for reuse and sharing.

Ontologies[2] aim at solving these problems. On the one hand, ontologies facilitate communication and knowledge sharing by providing a unifying framework for parties with different viewpoints. On the other hand, ontologies can improve interoperation and cooperation by providing unambiguous semantics in a formal, machine-interpretable way. Matchmaking can benefit from these general properties as far as the elements of the process are distributed or there are several viewpoints; additionally, the use of semantically rich specifications enable the use of specific forms of reasoning that are not available when using a syntactic approach, such as for example subsumption and disjunction. Below we provide a simple motivating example to illustrate on such forms of reasoning for matchmaking.

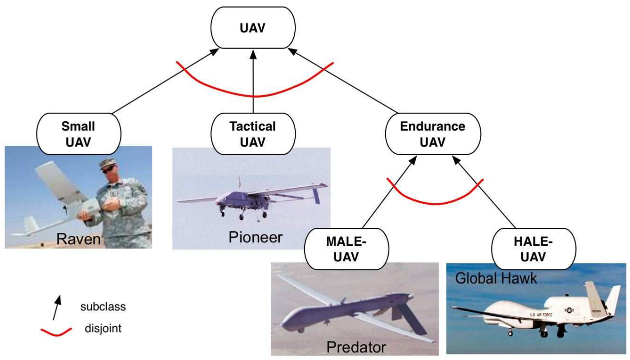

FIGURE 2. Partial classification of unmanned aerial vehicles (UAVs)

Figure 2 depicts a partial classification of unmanned aerial vehicles (UAVs). The figure shows six classes of UAV, and the various specialisation (subclass) relationships among them. At the top of the classification, the UAV class encompasses all kinds of UAV, which may range in cost from a few thousand dollars to tens of millions of dollars, and range in capability from Micro Air Vehicles (MAV) weighing less than one pound to aircrafts weighing over 40,000 pounds. In this example we include just three categories that are specialisations of the UAV class; these are, from left to right: the Small UAV (SUAV), designed to perform "over-the-hill" and "around-the-corner" reconnaissance; the Tactical UAV (TUAV), which focuses on the close battle in direct response to a brigade commander; and the Endurance UAV (EUAV), which supports a division in deep battle. Further, we have included two categories that specialise the Endurance UAV class: the Medium Altitude Long Endurance (MALE) UAV, designed to operate at altitudes between 5000 and 25000 feet, and the High Altitude Long Endurance (HALE) UAV, designed to function as Low Earth Orbit satellites. The arcs between subclass relationships indicate a disjoint relationship among subclasses; a disjoint relation among a set of classes entails

[2]For a modern definition of the term, we refer the reader to [5]: "an ontology is a set of logical axioms designed to account for the intended meaning of a vocabulary".

that an individual cannot belong to more than one of those classes; for example, a UAV that is classified as a Small UAV, can not be classified as being a Tactical UAV. Next, we introduce some basic examples illustrating specific forms of reasoning enabled by the use of ontologies. Let us suppose that we have the following UAVs available for a mission:

- A Pioneer, which is a TUAV
- A Predator, which is a MALE-UAV
- A Global Hawk, which is a HALE-UAV

Now suppose that as part of a given mission a persistent-surveillance task over a wide area is required to detect any suspicious movement. This kind of task is best served by an Endurance UAV, since it is able to fly for long periods of time. From just the concept definitions we know that: (1) the Pioneer is not an endurance UAV (because of the disjoint relationship among Endurance-UAV and TUAV), and (2) both the Predator and the Global Hawk are Endurance-UAVs (because of the subclass relationships)[3]. Therefore, the matchmaking process will select both the Predator and the Global Hawk as the assets satisfying the specified mission requirements.

Now, suppose that according to the weather forecast, storms are very likely to occur in the area of operations during the surveillance period. Then, the best option would be to use a HALE-UAV, which has the capability of flying "above the weather". Consequently, the matchmaking process would select the Global Hawk as the only asset satisfying the mission requirements.

The UAV examples introduced above refer to a simple form of matching relationships known as *subsumption*, but it is possible to devise more complex information containment relationships and even an ordinal ranking scale comprising several degrees of matching just by using the subclass relationship. Figure 3 represents graphically the main kinds of matching relations that are found in the literature in terms of information containment, using concepts from the ISR domain. Q denotes a query which specifies some requirements to be met, which in our context are ISR requirements, and $S1 - S5$ denote the specification of components to be matched against Q, which in our domain are associated with ISR assets such as UAVs.

Commencing at the left, our query Q specifies two basic requirements to be met: (1) provide infrared (IR) vision and (2) be able to carry out night reconnaissance. Going from left to right and top to bottom, the figure shows the specification for several assets that verify different types of relation in terms of information containment. Below follows a description of these matching relations listed in decreasing strength order:

1. *ExactMatch(S1, Q)* holds when the specification of a component provides exactly the same type of information described by the query. In the example, $S1$ describes an asset that provides IR vision and is designed to perform night reconnaissance tasks, just as stated in Q. This is represented as $S1 = Q$.
2. *Plugin(S2, Q)* holds when the class of information described by the query subsumes (i.e. is more general than) the class of information specified by the component. In

[3] Note that we only state minimum explicit information about the UAVs (e.g. Pioneer is-a Tactical-UAV); everything else is inferred from the concept definitions (e.g. the Pioneer is not a HALE-UAV).

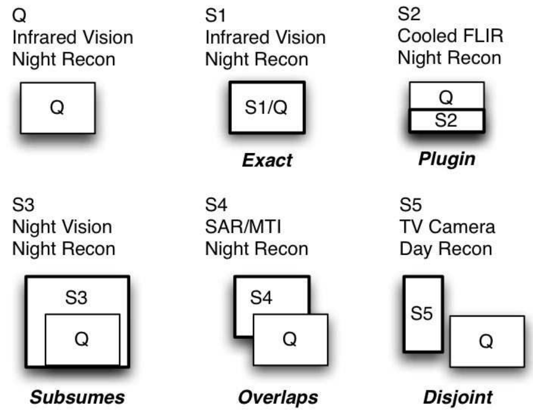

FIGURE 3. Basic matching relationships

the example, the asset described by $S2$ refers to a Cooled FLIR (forward looking IR), which is a specific type of IR camera. This is represented as $Q \subseteq S2$.

3. *Subsumes (S3, Q)* holds when the class of information described by the query is subsumed by the specification of the component, i.e. when the specification of the component is more general than the query. In the example, $S3$ refers to an asset providing night vision capability, which is a more general concept than infrared vision, and also provides night reconnaissance. This is represented as $S3 \supseteq Q$.
4. *Overlaps(S4, Q)*: holds when the query and the specification share some information, but neither one subsumes the other entirely. In our example, $S4$ describes an asset that provides night reconnaissance as required by Q, but the first requirement is not satisfied, since it carries a radar (SAR, Synthetic Aperture Radar) instead of an IR camera, and these two concepts are disjoint. This is represented as $S4 \cap Q$.
5. *Disjoint(S4, Q)*: holds when there is no degree of information containment between the specification of the component and the query. In the example, $S5$ describes an asset that provides TV video and is suited to perform day reconnaissance tasks; radar imagery is disjoint with IR vision, day reconnaissance is disjoint with night reconnaissance, so there is no intersection or information containment between the concepts. This is represented as $S4 \perp Q$

The kind of matching relationships introduced above are typically used to discover software components or services satisfying some specific requirements. Herein we are proposing to use these kinds of matching relations to discover ISR assets that satisfy intelligence requirements. Although different matchmaking problems could seem very similar in term of basic matching relationships used, they could differ when considering the matching relationship at the component level, rather than at the attribute level.

2.2. Matchmaking abstract architecture

A matchmaking application is not entirely characterised by the semantic relationships that might be established among concepts. An important issue of a matchmaking application is the distinction between the attribute-level and the component-level: a component may be described by different attributes, and so different matching schemas could be applied to each attribute depending on the particular meaning or role it plays within the component.

In our application, we have identified two main kinds of components to be matched against the ISR requirements of a mission, each one characterised by different attributes that deserve a separate treatment. Note that the kind of capability requirements that are relevant to select a specific kind of sensor are quite different from the requirements that are relevant to select a platform. For example, in order to assess the utility of different sensors it is very important to consider the kind of intelligence to be produced (e.g. Imagery Intelligence (IMINT), Measurement and Signature Intelligence (MASINT), Signals Intelligence (SIGINT), since each type of sensor provides information that supports a different kind of intelligence (e.g. infrared cameras support IMINT, while acoustic sensors support MASINT). Besides, to select a specific UAV for a reconnaissance mission there are other factors to consider, such as the range to the targets of interest, the presence or absence of enemy anti-air assets, and so on. In addition, UAVs are limited in the weight and type of sensors they can carry, and the performance of some sensors may be influenced by conditions that depend on the platform they are attached to, such as the altitude. Therefore, one cannot select UAVs and sensors independently; instead, the interaction between these components must also be taken into account.

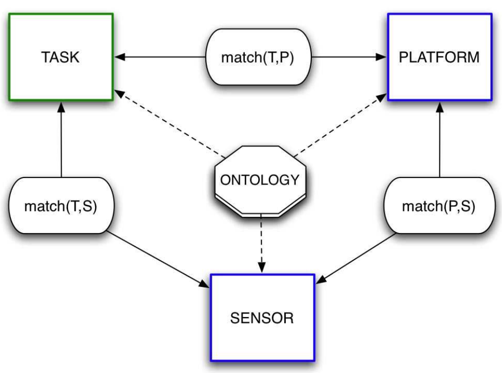

FIGURE 4. Abstract matching architecture

To address the issues above, we define an abstract architecture based on three types of components and three kinds of matching relations, as showed in Figure 4. In each case we build on existing work in defining ontologies for the specific components:

- *Tasks* are the actions to be performed in order to accomplish a mission. A task may have attached environmental conditions (weather, terrain, enemy, etc) that are expected to impact the performance of a task. We seek to use standardised catalogues of Tasks and Conditions such as those found in the Universal Joint Task List[4]
- *Sensors* are the assets that collect the information required to satisfy the intelligence requirements of a mission. However, sensors do not operate as independent entities, they have to be attached to (or carried by) devices that provide them with energy, protection, mobility, etc. Several ontologies of sensors already exist, e.g. [6, 7].
- *Platforms* are the systems to which sensors are attached so as to get energy, protection, mobility, communication, etc. Platforms include both static and mobile systems operating on land, in sea and air. Again, some work has already been done to create ontologies of these, e.g. [8].

The three components involved and the dependencies between them result in three different matching relations, as follows:

- *Task-Sensor matching*: a sensor S matches a task T, $match(T, S)$, if S provides the information collecting capabilities required to satisfy the intelligence requirements of T.
- *Task-Platform matching*: a platform P matches a task T, $match(T, P)$, if P provides the kind of ISR-supporting capabilities (mobility, survivability, communication) required to perform T.
- *Platform-Sensor matching*: a sensor S matches a platform P, $match(P, S)$, if S can be carried by and is compatible with the characteristics of P.

In order to satisfy the ISR requirements of a mission one needs to select both a platform and a combination of sensors such that the three matching relations of the architecture are satisfied simultaneously.

2.3. Towards a multidimensional solution

Although one can envisage a single ontology supporting the entire sensor-mission matchmaking process, actually we adhere to the Semantic Web vision of multiple interlinking ontologies covering different aspects of the domain. First, we provide an ontology based on the Missions and Means Framework (MMF), which is basically a collection of concepts and properties that are essential to reason about the process of analysing a mission and attaching the means required to accomplish it (mission, task, capability, or asset). Then we provide a second ontology that refines some of the generic concepts in the MMF ontology so as to represent the ISR-specific concepts that constitute our particular application domain. This second ontology comprises several areas frequently organised as taxonomies, such as a classification of sensors (acoustic, optical, chemical, radar) and information sources, a classification of platforms (air, sea, ground and underwater platforms), a classification of mission types, or a classification of capabilities. As noted in the previous section, there are existing ontologies covering at least part of each of these domains.

[4] See http://www.dtic.mil/doctrine/jel/cjcsd/cjcsm/m350004c.pdf and http://www.daml.org/2002/08/untl/

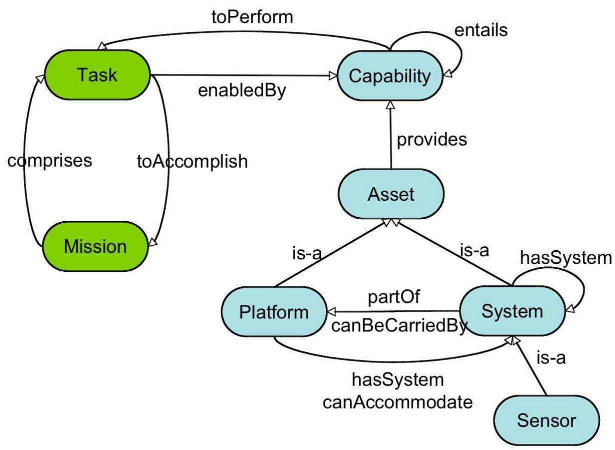

FIGURE 5. Main ontological concepts and their relationships

Figure 5 shows a high level view of the main concepts and relationships that support our semantic matchmaking approach. On the left hand side, we find the concepts related to the mission: a mission comprises several tasks that need to be accomplished. On the right hand side we find the concepts related to the means: a sensor is a system that can be carried by or constitutes part of a platform; inversely, a platform can accommodate or have one or more systems, and both platforms and systems are assets; an asset provides one or more capabilities; a capability can entail a number of more elementary capabilities, and is required to perform certain type of tasks and inversely, a task is enabled by a number of capabilities.

In the next section, we focus on the use of argumentation to manage the gathering of evidence from a set of sensors and sources assigned to a task.

3. Arguing About Evidence in Partially Observable Domains

In this section, we examine how argument may be used to reason about sensor assignment based on evidential and diagnostic reasoning. Informally, we are trying to address situations where different agents, each with their own goals and viewpoints, are attempting to reach a shared agreement about the state of a subset of their environment. By reaching agreement, they may take decisions about how their actions should be coordinated. We further assume that the environment is partially observable, and that any information about it is obtained through the use of (possibly incorrect) evidence. Finally, we assume that the agents are self interested. The argumentation approach has a number of advantages over competing methods, including understandability, improved running time and ease of knowledge representation.

Without a trusted third party, a centralised solution to this problem is difficult. Our proposed approach involves the agents engaging in dialogue with each other, exchanging arguments, and obtaining evidence (possibly via existing sensors) for additional information about the environment. By basing arguments on evidence, a shared world view can be constructed. To tackle the problem, a representation mechanism for the environment, agents' knowledge and arguments is required, as well as a method for determining which conclusions are justified when opposing arguments interact. A specification is also needed, detailing how dialogue may take place. Finally, agents must be able to decide which arguments to advance, and what sensors to probe for evidence.

Prakken [10] identified these as the logical, dialectic, procedural and heuristic layers of an argument framework. Our logical layer is built around Subjective Logic [11], allowing us to represent concepts such as likelihood and uncertainty in a concise and elegant manner. The way in which arguments are constructed in our framework and used at the dialectic level is intended to support a rich representation of arguments; we are able to represent concepts such as accrual of arguments, argument schemes and argument reinforcement in a natural manner. While the logical and dialectic layers are domain independent, acting as a general argument framework, the explicit introduction of evidence at the procedural level allows us to attack our problem.

Evidence is gathered via sensors, where a sensor refers to anything that can determine the state of a portion of the environment. Multiple sensors may exist for certain parts of the environment, and some of these sensors may be more accurate than others. Finally, sensors may not perform their services for free. Thus, sensors capture an abstract notion of a source of evidence within our framework.

At the procedural level, agents engaging in dialogue, taking turns to advance arguments and probe sensors in an attempt to achieve their goals. In this context, an agent's goal involves showing that a certain environment state holds. We assume that an agent associates a utility with various goal states. Our heuristic layer guides an agent and tells it what arguments to advance, and which sensors to probe during its turn in the dialogue game.

The logic of our framework is built on Subjective Logic [11], which, in turn, is based on Dempster-Schafer theory. We may assign an *opinion* to predicates representing portions of the environment. These opinions are $\langle belief, disbelief, uncertainty \rangle$ triples[5].

Jøsang defined a large number of operators that are used to combine opinions, some of which are familiar such as conjunction and disjunction, and some less so such as abduction. We look at three operators, namely negation, discounting, and consensus.

The propositional negation operator calculates the opinion that a proposition does not hold. A negated opinion's belief is equal to the original opinion's disbelief, while the original disbelief becomes the opinion's belief. Uncertainty remains constant.

[5]This is in fact a simplification, Subjective Logic ordinarily uses 4-tuples, with the forth element representing atomicity.

Discounting is used to model hearsay. That is, given that an agent has an opinion a about agent β's reliability, and that β has an opinion x about something, without any additional information, α will have an opinion $a \otimes x$, where $\otimes$ is the discounting operator.

The independent consensus operator gives the opinion an imaginary agent would have about x if it had to assign equal weighting to different opinions x_1, x_2 about a state of the world x. It is represented as $x_1 \oplus x_2$.

3.1. The Framework

Following Prakken's model[10], we build our framework in layers, starting at the logical layer, where we describe how an argument is constructed. In the dialectic layer, we look at how arguments interact, and then show how agents may engage in dialogue in the procedural layer. Finally, in the heuristic layer, we show how agents can decide which lines of argument should be advanced in a dialogue.

Facts in our model are represented as grounded predicates, and have an associated opinion. An argument is an instantiated argument scheme [12] linking facts to other facts. Argument schemes are common, stereotypical patterns of reasoning, often taking on a non-deductive or non-monotonic form. A simple argument scheme (Modus Ponens) could be represented as follows:

$$(ModusPonens, \{holds(A), implies(A, B)\}, \{holds(B))\}, F, true)$$

Here, F is:

$$\omega(holds(B)) = \begin{cases} \langle 0, 0, 1 \rangle & b(holds(A)) < 0.5 \text{ or} \\ & b(implies(A, B)) < 0.5 \\ \omega(holds(A)) & b(holds(A) < b(implies(A, B)) \\ \omega(implies(A, B) & \text{otherwise} \end{cases}$$

where $holds(A)$ and $implies(A, B)$ are the premises of the argument scheme (i.e. these facts must hold for the argument scheme to be instantiated into an argument). $holds(B)$ is the conclusion of the argument scheme (i.e. this fact may be instantiated if the argument scheme is applicable), F is a function allowing us to compute the opinion for the conclusion based on the opinions associated with the premises, and finally $true$ is an *applicability* function, stating any restrictions on the application of the argument scheme. We make use of first order unification to transform an argument scheme into a concrete argument. any symbols in capital letters are unified with facts, as done in prolog, so as to instantiate the scheme.

Until now, we have described what individual arguments look like. However, arguments do not exist in isolation. Instead, they interact with each other, reinforcing or weakening opinions about predicates in the process. Unlike most other argumentation frameworks, we do not explicitly model rebutting and undercutting attacks to show how arguments interact. Instead, we use the concept of accrual of arguments to allow for both argument strengthening and weakening. To represent interactions between arguments, we must be able to answer the following question: what happens when two different arguments have opinions about a (partially shared) set of predicates in their conclusions?

The independent consensus operator gives us a default technique for applying accrual. Thus, given a set of arguments for and against a certain conclusion, and given no extra information, we apply the consensus operator based on the opinions garnered from the arguments to arrive at a final opinion for the conclusion.

While some researchers have suggested that accrual of arguments is an argument scheme and can be treated as such (arguably, for example [13]), Prakken's view, in our understanding, is that the best way to handle accrual of arguments is by following a two stage process. First, determine what arguments may enter into an accrual, and second compute the effects of the accrual. We agree that accrual of arguments cannot be treated as "just another" argument scheme due to its role and nature. We believe, however, that in certain situations (usually obeying principle 1), accrual of evidence can be treated as an argument scheme. The way in which our framework aligns these two views is one of its most unique aspects.

Informally, given multiple arguments for a conclusion, we apply the standard consensus rule. However, if an argument is advanced which subsumes (some of the) arguments which take part in the consensus, the subsumed argument's conclusions are ignored, and the subsuming rule is used instead. If any of those arguments are attacked and defeated, then our accrual rule is itself defeated, allowing all its undefeated (and previously subsumed) members to act again. If some of the newly activated sub-members were, in turn, part of accruals, those accruals would enter into force again.

Given these underpinnings, it is possible to provide an algorithm for evaluating how sets of instantiated arguments interact. Such an algorithm operates in a way similar to the way reasoning occurs in probabilistic networks, and is best explained by thinking of our sets of arguments and predicates as a graph. Both predicates and arguments can be thought of as nodes, with a directed edge between the two if the predicate appears in the premises or conclusions of an argument. The edge enters the argument in the case of the predicate being a premise, and exits the argument otherwise.

To operate, our algorithm requires an argument graph, as well as a starting set of opinions. We assume that these opinions are not under dispute, and the associated nodes must, therefore, have no edges leading into them. Our algorithm then propagates these opinions forward through the graph, until all applicable arguments in the graph have been taken into account. The specific details of the algorithm appear in [14].

At this point, we have a way of determining which conclusions hold given a set of arguments. It is now possible to define a procedure for how the set of arguments is generated. This can be done in two phases. In the first, a dialogue between agents may be defined. This states when an agent may make an utterance, and what form these utterances should take. We assume that agents take turns to speak, and that the game ends when both agents pass (i.e. say nothing) during their turn.

Since we are interested in arguing about evidence in partially observable domains, we assume that the environment holds a number of sensors. These sensors may be probed to obtain opinions about the value of various relations. In practise, sensors may be agents, static parts of the environment, or some other entity capable of providing an opinion about the environment. We assume that multiple sensors can give opinions about the same relations, and that some sensors are more reliable than others.

During their turn, an agent may advance a connected set of arguments, and probe a number of sensors. These sensor probings are one way to associate an opinion with a fact. The other way is to have the fact be the conclusion of an argument.

At each step in the dialogue, an opinion is calculated for every fact. When participating in the game, an agent must decide which utterance to make. We associate a cost to probing actions, and a utility gain to the showing that certain facts hold in the world. Then the agent selects the utterance that maximises their utility. In effect, the agents perform one step lookahead during their turn. Increasing the level of lookahead requires some form of opponent modelling.

3.2. An example scenario

In this section, we describe a dialogue in a hypothetical sensor assignment scenario. A commander, fronted by an agent α, has a mission (labelled $mission(m)$) to accomplish. To successfully execute the mission, he requires the use of a sensor package that can be deployed on either a Predator UAV, or a Sentry UGV (with deployment on the UAV preferred by the commander). Another agent β, is also present in the system. Both agents share some knowledge, but both also have private beliefs. β could represent another commander, a member of a coalition, or, though not explicitly examined in this scenario, someone with their own goals, some of which may not be compatible with α's mission. We assume that certain sensors have already been deployed in the field, and that the agents have access to these and other sources of information such as GIS systems. α must argue with β in an attempt to allocate resources for its mission. In the interests of clarity, the description of the dialogue that follows is semi-formal.

Assume the agents have the following argument schemes available to them:

Name	**Premises**	**Conclusions**
$ModPon$	$A, B, implies(A, B, C)$	C
$HumInt$	$atLocation(E, L), claims(E, A), inArea(A, L)$	A
$MisAss$	$capable(T, R), available(R), hasTask(M, T)$	$assigned(M, R)$
M_1	$higherPriority(M, N), uses(N, R)$	$reassignReq(N, M, R)$
M_2	$reassignReq(N, M, R), reassign(M, R)$	$assigned(M, R)$
D_1	$ugv(U), taskLocated(T, L), hasRoad(L)$	$capable(U)$
D_2	$ugv(U), taskLocated(T, L), mud(L)$	$capable(U)$
D_3	$ugv(U), taskLocated(T, L), mud(L), hasRoad(L)$	$capable(U)$

We do not show the admissibility and mapping functions in this table, but assume that they are unique to their associated argument scheme.

Some arguments here are very general, for example, $ModPon$ represents standard two premise Modus Ponens. Others, such as $HumInt$ and $MisAss$, are specific to the military domain. The former, similar to Walton's argument from expert opinion [12], represents an argument based on information from "expert" human intelligence. The latter argument scheme allows agents to reason about when a resource may be assigned to a task. M_1 and M_2 are very specific to the military domain, and represent how agents may reason about task assignments, while the remaining argument schemes are used to reason about the applicability of a UGV to different types of domains. Note that D_3 is able to handle more specific cases than D_1 and D_2.

α would like to assign either a UGV or a UAV to his mission (preferring a UAV), and thus has the goals

$$assigned(mission(m), uav(predator)), assigned(mission(m), ugv(sentry))$$

With a higher utility being given to the former goal.

Both agents are aware of the following facts:

$hasTask(mission(m), task(t))$	$higherPriority(mission(m), mission(n))$
$capable(t, uav(predator))$	$implies(recentRain(l), sand(l), mud(l))$
$ugv(sentry)$	$taskLocated(t, l)$
$atLocation(h, l)$	

Agent α also believes that $available(uav(predator)), hasRoad(l)$ and, believes there is a good chance that, if necessary $reassign(mission(m), uav(predator))$ would work. It also believes that no rain has fallen at l, and that the human intelligence assets would agree with it, i.e. $claim(h, \neg recentRain(l))$ and $inArea(l, \neg recentRain(l))$.

Agents can probe a GIS system to determine the status of $hasRoad(l)$ at very little utility cost, while $recentRain(l)$ and $sand(l)$ would cost α more utility. Probing whether the UAV is available can be done at very little cost by looking at different inventory databases. We also define two expensive sensors for the reassignment request and the reassignment itself. These represent the cost of going up the chain of command to ask for the UAV/UGV to be reassigned. Finally, it is possible to probe the opinion of the human intelligence for details such as the $claim()$ predicate, but this is very expensive as the location of the assets might be compromised.

Agent α begins the conversation by making the utterance

$$((MisAss, \{hasTask(mission(m), task(t)), capable(t, uav(predator)), \\ available(uav(predator))\}, \{assigned(mission(m), uav(predator))\}), \\ \{available(uav(predator))\})$$

In other words, it attempts to check that the predator UAV is available for the mission, and assign it (if possible). We assume that the probe succeeds.

β responds with its own sensor probe $(, \{available(uav(predator))\})$, as it believes the UAV is not available.

When this returns an opinion of $\langle 0.1, 0.9, 0 \rangle$, α's argument is nullified. α now has two options. It may either ask to get the UAV reassigned to it (which would involve a large cost in utility), or may attempt to use the UGV. Since low cost sensor probes are available to it, it will get a greater utility gain by attempting to use the UGV than by following the former route. It thus makes the utterance:

$$(\{(D_1, \{hasRoad(l), taskLocated(t,l), ugv(sentry)\}, \{capable(t, ugv(sentry)\}), (MisAss, \{hasTask(mission(m), task(t)), capable(t, ugv(sentry)), available(ugv(sentry)))\}, \{assigned(mission(m), ugv(sentry))\})\}, \{available(ugv(sentry)), hasRoad(l)\})$$

In other words, it claims that since there are roads at the location, and since the UGV is available, it can use it for its mission.

β believes that (due to rain and sand), mud exists at the location. This leads to the utterance:

$$((\{ModPon, \{recentRain(l), sand(l), implies(recentRain(l), sand(l), mud(l))\}, \{mud(l)\}), (D_3, \{ugv(sentry), taskLocated(t,l), mud(l), hasRoad(l)\}, \{capable(t, ugv(sentry))\}), \{recentRain(l), sand(l)\})$$

Argument D_3 subsumes D_1, meaning that $capable(t, ugv(sentry))$ is no longer believed.

α can now either probe human intelligence to check for the presence of mud, or attempt to get the mission's resources reassigned (we assume that the UAV was assigned to $mission(n)$). The latter option yields it more utility, and it makes an utterance using argument schemes M_1 and M_2, while probing $reassign$ and $reassignReq$.

β has no more responses, and thus passes, as does α, meaning that the UAV will be assigned to the mission.

Obviously, the dialogue described here is simplified. In a realistic scenario, the agents would have access to more information and many more argument schemes. Figure 6 illustrates the argument graph that resulted from this dialogue, though for clarity, part of the graph is omitted.

While α has managed to get the UAV assigned, it paid a steep utility cost. α would have preferred to get the UGV assigned to it without having to have asked for the reassignment of resources, but would then not have been able to complete its mission (due to β's criticism).

Once the dialogue terminates, predicates are associated with opinions. Depending on the form of the admissibility function, they, or their negation may be judged to be admissible. Thus, for example, if $assigned(mission(m), uav(predator))$ exceeds a certain threshold, it is assumed to be assigned to mission m.

3.3. Discussion

Our framework was designed to allow for complex argument to take place, particularly in the domain of evidential reasoning. Uncertainty is a key feature of such domains, hence our decision to base our framework on Subjective Logic. Catering for uncertainty in argumentation frameworks is by no means new. Pollock [13] made probability a central feature of his OSCAR architecture. We disagree with his extensive use of the "weakest

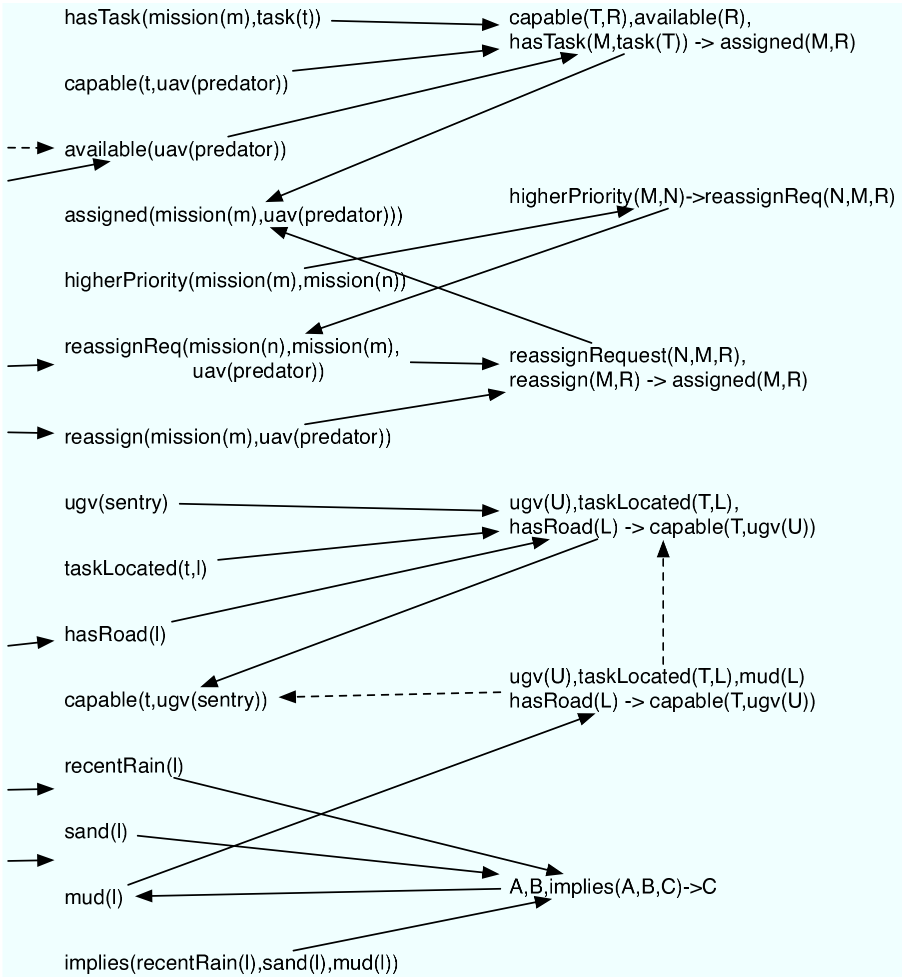

FIGURE 6. The argument graph for the dialogue. The second use of the *MisAss* argument scheme is omitted. Solid arrows indicate support for an argument or predicate, while dashed lines represent an attack or weakening. Arrows with no source indicate sensor probes.

link” principle, however, believing that, while it may hold in general, it is not always applicable (as mentioned in [15]. His use of probability, rather than uncertainty is another point at which our approaches diverge.

Our use of Subjective Logic as the basis of the framework provides us with a large amount of representational richness. Not only are we able to represent probability (via belief), but we are also able to speak about ignorance (via uncertainty). Differentiating between these two concepts lets us represent defaults in a natural, and elegant way. A default can be represented by specifying, within the A function, that a conclusion may hold as long as the disbelief for a premise remains below a certain threshold. By requiring that belief remain above some threshold, normal premises can also be represented. A simple example of this was provided in the previous section, where everyone, by default, is assumed to be an expert. Burden of proof [16] is very closely related to defaults, and we model it in the same way.

Argument schemes have been extensively discussed in the literature (see for example [17, 12]). A small, but growing number of argumentation frameworks provide explicit support for argument schemes (e.g. [18]). We believe that supporting argument schemes in our framework not only enhances argument understanding, but that such support also provides clear practical advantages, including the separation of domain and argument knowledge, re-usability, and a possible reduction in computational complexity when deciding what arguments to advance. The separation between arguments and agent knowledge created by argument schemes raises the intriguing possibility of the modification and dynamic creation of argument schemes during a dialogue.

The interplay between sensors and arguments is an area in which little formal work has been done [19]. While our model is very simple, it elegantly captures the fact that sensor data is inherently unreliable in many situations. Enriching our model of sensors is one area in which we plan to do future work.

4. Conclusions

In this chapter, we have described how two aspects of our work on managing resources in Virtual Organisations can be applied to the problem of deploying and utilising intelligence assets in coalition operations. We have shown how modern military doctrine, in the form of the Missions and Means Framework, can be captured in a semantically formal representation, allowing sensors and other ISR resources to be assigned to a mission through matchmaking reasoning. This approach has the advantages that the MMF concepts are familiar and transparent to users (e.g. commanders) and the assignments are logically sound.

We have also shown how argumentation can be used to manage the process of gathering and reasoning about evidence from sensors and sources. Because such sources are fallible, and the military domain typically involves environments that are only partially observable, we needed to devise a novel framework for argumentation in domains containing uncertainty. The concept of argument schemes is built into the framework, allowing for a rich set of primitives to be utilised in the argumentation process. We have also attempted to cater for other important concepts in argument such as accrual of arguments, defaults, and burden of proof. While the lowest levels of the framework are general enough to be applied to almost any area in which argument is used, the higher levels are aimed at evidential reasoning, incorporating abstract models of sensors and the notion of obtaining information from the environment.

References

[1] Norman, T.J., Preece, A.D., Chalmers, S., Jennings, N.R., Luck, M.M., Dang, V., Nguyen, T., Deora, V., Shao, J., Gray, W.A., Fiddian, N.J.: CONOISE: Agent-based formation of virtual organisations. Knowledge-Based Systems **17**(2–4) (2004) 103–111

[2] Preece, A., Chalmers, S., McKenzie, C.: A reusable commitment management service using semantic web technology. Knowledge-Based Systems **20**(2) (2007) 143–151

[3] Sheehan, J.H., Deitz, P.H., Bray, B.E., Harris, B.A., Wong, A.B.H.: The military missions and means framework. In: Proceedings of the Interservice/Industry Training and Simulation and Education Conference. (2003) 655–663

[4] Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. Journal of Human Computer Studies **43(5/6)** (1994) 907–928

[5] Guarino, N.: Formal ontologies and information systems. In: Proceedings of the 1st International Conference on Formal Ontologies in Information Systems (FOIS-98), IOS Press (1998) 3–15

[6] McMullen, D., Reichherzer, T.: The common instrument middleware architecture (CIMA): Instrument ontology & applications. In: Proceedings of the 2nd Workshop on Formal Ontologies Meets Industry, Trento, Italy (2006) 655–663

[7] Russomanno, D., Kothari, C., Thomas, O.: Building a sensor ontology: A practical approach leveraging ISO and OGC models. In: Proceedings of the 2005 International Conference on Artificial Intelligence, CSREA Press (2005) 637–643

[8] Bermudez, L., Graybeal, J., Arko, R.: A marine platforms ontology: Experiences and lessons. In: Proceedings of the 2006 Workshop on Semantic Sensor Networks, Athens GA, USA (2006)

[9] Reed, C.A., Norman, T.J., eds.: Argumentation Machines: New frontiers in argumentation and computation. Kluwer (2003)

[10] Prakken, H., Sartor, G. In: Computational Logic: Logic Programming and Beyond. Essays In Honour of Robert A. Kowalski, Part II. Volume 2048 of LNCS. Springer-Verlag (2002) 342–380

[11] Jøsang, A.: A logic for uncertain probabilities. Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **9** (2001) 279–311

[12] Walton, D.N.: Argumentation Schemes for Presumptive Reasoning. Erlbaum (1996)

[13] Pollock, J.L.: Cognitive Carpentry. Bradford/MIT Press (1995)

[14] Oren, N., Norman, T.J., Preece, A.: Subjective logic and arguing with evidence. Artificial Intelligence Journal (2007) to appear

[15] Prakken, H.: A study of accrual of arguments, with applications to evidential reasoning. In: Proc. of the 10th Int. Conf. on Artificial Intelligence and Law. (2005) 85–94

[16] Walton, D.N.: Burden of proof. Argumentation **2** (1988) 233–254

[17] Bex, F., Prakken, H., Reed, C., Walton, D.: Towards a formal account of reasoning about evidence: Argumentation schemes and generalisations. Artificial Intelligence and Law **11**(2-3) (2003) 125–165

[18] Verheij, B.: Dialectical argumentation with argumentation schemes: An approach to legal logic. Artificial intelligence and Law **11** (2003) 167–195

[19] Oren, N., Norman, T.J., Preece, A.: Argumentation based contract monitoring in uncertain domains. In: Proc. of the 20th Int. Joint Conf. on Artificial Intelligence, Hyderabad, India (2007) 1434–1439

Alun Preece
Department of Computing Science
University of Aberdeen
Aberdeen, AB24 3UE
UK
e-mail: `apreece@csd.abdn.ac.uk`

Timothy J. Norman
Department of Computing Science
University of Aberdeen
Aberdeen, AB24 3UE
UK
e-mail: `tnorman@csd.abdn.ac.uk`

Mario Gomez
Department of Computing Science
University of Aberdeen
Aberdeen, AB24 3UE
UK
e-mail: `mgomez@csd.abdn.ac.uk`

Nir Oren
Department of Computing Science
University of Aberdeen
Aberdeen, AB24 3UE
UK
e-mail: `noren@csd.abdn.ac.uk`