Knowledge-Driven Agile Sensor-Mission Assignment

A. Preece*, D. Pizzocaro*, K. Borowiecki*, G. de Mel†, W. Vasconcelos†,
A. Bar-Noy‡, M. P. Johnson‡, T. La Porta§, H. Rowaihy§
*Cardiff University, UK; contact email: a.d.preece@cs.cardiff.ac.uk; †University of Aberdeen, UK
‡City University of New York, USA; §Pennsylvania State University, USA

***Abstract*—In this paper, we show how knowledge representation and reasoning techniques can support sensor-mission assignment, proceeding from a high-level specification of information requirements, to the allocation of assets such as sensors and platforms. In our previous work, we showed how assets can be matched to mission tasks by formalising the military missions and means framework in terms of an ontology, and using this ontology to drive a matchmaking process derived from the area of semantic Web services. The work reported here extends the earlier approach in two important ways: (1) by providing a richer and more realistic way for a user to specify their information requirements, and (2) by using the results of the semantic matchmaking process to define the search space for efficient asset allocation algorithms. We accomplish (1) by means of a rule-based representation of the NIIRS approach to relating sensed data to the tasks that data may support. We illustrate (2) by showing how the output of our matching process can drive a well-known efficient combinatorial auction algorithm (CASS). Finally, we summarise the status of our illustration-of-concept application, SAM (Sensor Assignment to Missions), and discuss various roles such an application can play in supporting sensor-mission assignment.**

I. Introduction

The *sensor-mission assignment* problem consists of allocating a collection of intelligence, surveillance and reconnaisance (ISR) assets (including sensors and sensor platforms) to a set of missions, in an attempt to satisfy the ISR requirements of those missions. This problem is challenging for several reasons. Firstly, the relationship between assets and missions is complex. Missions typically include a number of tasks, and those tasks require various capabilities, some of which are concerned with ISR. Assets provide capabilities, so one aspect of the problem includes modelling the relationship between required and provided capabilities, in order to determine which assets are suitable for which tasks [1]. Secondly, it is often the case that a collection of assets are needed to meet the requirements of a task, not just one, so there is a need to model these asset bundles [2]. Thirdly, there is competition between missions for assets, and often demand exceeds supply, necessitating hard resource allocation decisions [3]. Finally, all of this needs to be done in a highly dynamic manner, as the situation — mission needs and asset availability — is in a constant state of flux [4].

In our previous work we have argued for an *agile* approach to sensor-mission assignment, which supports just-in-time decision-making and resource allocation [5]. In terms of maximising agility, we aim to provide as much automation as possible in the assignment of sensing assets to missions. This involves attempting to capture the information requirements of mission tasks in a manner that is as independent as possible of the capabilities of specific types of sensor or platform, to allow multiple degrees of freedom in allocation and reallocation of assets. For this reason, we adopted a knowledge-based approach to modelling tasks and assets that was able to build on previous work done in defining sensor, platform, and task ontologies. The chief advantage we claimed for this approach was that it was able to exploit principled techniques from the area of Semantic Web Services to solve the sub-problem of matching asset capabilities to task requirements [6]. As we will show, however, this solution was limited in terms of supporting maximal agility in sensor-task assignment, because the representation of information requirements was at a relatively low level. One of the goals of the work presented in this paper was to "open up" the space of asset choices as widely as possible, by moving to a higher-level representation of mission requirements.

As noted above, solving the sensor-mission assignment problem in an agile manner requires a dynamic approach for allocating asset instances among a set of competing missions. Moreover, in some cases, a bundle of assets is necessary to meet the requirements of a single mission task, and we have shown that this is an NP-hard problem [7]. In our earlier work, we have argued that the knowledge-based approach allows us to delineate the search space for such allocation algorithms [6], [8]. However, in demonstrating this we made the restrictive assumption to limit bundles to a single type of sensor, or single sensor platform. In this paper we show the general case where our knowledge-based approach can drive the generation and allocation of arbitrary bundles of sensors and platforms among multiple competing missions, using a well-known efficient combinatorial auction algorithm.

The paper is structured as follows: Section II provides a recap of our formulation of the sensor-mission assignment problem as a graph comprising mission tasks, bundles, and individual assets. In Section III, we show how the tasks-bundles-assets graph for a specific problem instance can be constructed using our original ontology-based approach. Then, in Section IV we argue for a higher-level representation of mission requirements, and show how this can be achieved by formalising the well-known NIIRS approach. Section V

demonstrates how the core knowledge-based approach can be used to drive asset allocation, using the CASS combinatorial auction algorithm as an illustration. In Section VI we review the status of our illustration-of-concept application, SAM (Sensor Assignment to Missions), and discuss a variety of roles this kind of tool can play in supporting the process of sensor-mission assignment.

While other papers have presented earlier or incomplete parts of our knowledge-driven approach to sensor-mission assignment (notably [6], [8], [9]) and resource allocation (notably [10], [7]) this is the first paper to show how these elements can provide an integrated solution.

II. Sensor-Mission Assignment Problem Formulation

We formulate the sensor-mission assignment problem as a graph; an example is shown in Figure 1. We distinguish three kinds of node:

- *Tasks* denote the entities that are competing for available assets.[1] In our approach, the only important feature of a task is its information requirements — these are what drive the asset matching and allocation processes — so the task nodes represent information requirements.
- *Bundles* are collections of individual assets (sensors and platforms). An arc between a bundle and a task indicates that the bundle is able to meet (at least to some extent) that information requirement (task). Each bundle has a unique *bundle type*, described in more detail below. Each bundle is composed of several assets and, depending on its type, may be suitable for several tasks.
- *Assets* represent the individual sensor and platform resources. An arc between an asset and a bundle means that that asset can be assigned to that bundle. Assets may be suitable for assignment to several bundles, again depending on the bundle type. Although not shown in the figure, an asset may be assigned to bundles of different types.

In these terms, a solution to the sensor-mission assignment problem is an assignment of bundles to tasks, subject to the constraints: each task can have at most one bundle assigned to it, a bundle may be assigned to at most one task, each asset may be assigned to at most one bundle. Thus, an assignment is a matching in the sub bipartite graph (B, T) and is a semi matching in the sub bipartite graph (A, B). Note that the thin arcs in Figure 1 show possible assignments, while the bold arcs show one actual assignment. The figure is drawn with tasks on the left and assets on the right to emphasise that, although the assignment arcs go right to left, solutions to the assignment problem are constructed left to right, that is, starting with tasks.

To illustrate this with a concrete example (originally from [5]): in a peace support operation [4], UK and US bases have been established to detect/deter insurgent activity on a border. The bases rely on a main supply route

[1] In much of our work, we consider these to be synonymous with "missions" (e.g. [8], [7]).

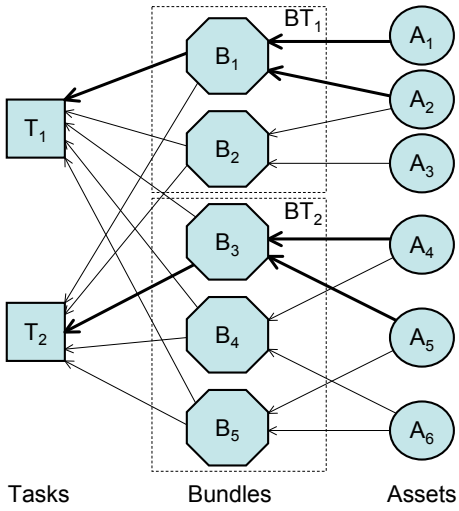

Fig. 1. Example sensor-mission assignment problem as a graph

(MSR) which must be surveilled and protected. Surveilling the border will likely involve, among other things, detection of suspicious vehicle activity near it: vehicle detection can be formalized as an information requirement task T_1. This may be accomplished by a variety of means, depending on the kinds of assets available. We assume these include flying a UAV over the border to gather IMINT, or using acoustic sensing to identify types of vehicles likely to be used by insurgents. Further, we assume that a single UAV can cover the area of interest (AoI), but that at least two acoustic arrays will be needed. Each of these options is represented as a type of bundle: $BT_1 = \langle\{\mathsf{UAV}, \mathsf{IMINT\text{-}sensor}\}\rangle, BT_2 = \langle\{\mathsf{AcousticArray}\}, \{\mathsf{AcousticArray}\}\rangle$. Assume there are multiple UAV assets available (A_1 and A_3) but only one functioning IMINT payload (A_2) that can be mounted on either A_1 or A_3. Assuming there are three acoustic arrays deployed in the area (A_4, A_5, A_6), then we can meet the requirements of T_1 by assigning UAV-IMINT bundles $\{A_1, A_2\}$ or $\{A_3, A_2\}$ or one of the pairs of acoustic arrays (e.g. $\{A_4, A_5\}$, $\{A_4, A_6\}$, $\{A_5, A_6\}$).

However, it is likely that there will be other tasks, potentially in competition for these assets; for example, a requirement to detect vehicles posing a potential threat to the MSR (T_2) may be accomplished by the same types of bundle as T_1 but, because the areas of interest (MSR and border) do not intersect, it will not be possible to share assets between these tasks. So, for example, if we assign the UAV to T_1 then we will have to satisfy T_2 by means of acoustic intelligence (ACINT).

We argue that, for a specific problem instance, construction of the task-bundle-asset graph requires a significant amount of domain knowledge. Specifically, need to know which types of bundle are suitable for which tasks, and which types of asset can be collected into bundles. In our original work [6] — summarised in the next section — this knowledge is expressed in the form of a set of domain ontologies, allowing us to apply a reasoning procedure to derive the graph for a given

set of tasks and assets. The bundle types are generated as a by-product of the reasoning procedure.

III. Ontology-Based Sensor-Task Matching

The goal of our earlier work was to solve the problem of matching types of assets (sensors and platforms) to types of tasks for which they are suitable. We sought to do this in a way that was conceptually well-founded in our target domain, and this led us to adopt the Missions and Means Framework (MMF). MMF was developed by the US Army Research Laboratory to provide a model for explicitly specifying a military mission and quantitatively evaluating the utility of alternative means to accomplish it [1]. While there existed previous work in formalising MMF using set theory [11], ours was the first formulation of MMF as an ontological framework. Our approach to sensor-mission matching is founded on the use of ontologies to represent the capabilities required by tasks and provided by assets, and reasoning to determine logically-sound matches [8]. Ontologies define formally the semantics of a set of terms, allowing automatic reasoning to be performed using the terms, in a manner consistent with their real-world interpretation [12].

A key insight was to adopt MMF essentially as a linking framework, to connect multiple interlinking ontologies covering different aspects of the domain: sensors, platforms, tasks, etc. We were thus able to draw upon substantial pre-existing work, as well as adhering to the Semantic Web notion of a "Web of ontologies" [13], which offers more extensibility, maintainability, and reusability than attempting to create a single all-emcompassing ontology for a given problem domain.

The way MMF describes the linking between missions and means naturally fits the notion of matchmaking: on the one hand, we have missions breaking down into operations, and operations into tasks, where each task may require different capabilities to be accomplished; on the other hand, we have the capabilities provided by assets as a result of aggregating the capabilities of its constituent systems and subsystems. In relation to our task-bundle-asset model of the previous section, the missions of MMF break down into tasks, and the means of MMF correspond to assets.

Figure 2 shows the main concepts and relations comprising our ontology based on MMF, which is implemented using the Web Ontology Language, OWL [13]. On the left hand side, we have concepts related to the mission: a Mission comprises one or more Operations to be carried out, and each operation breaks down into a number of Tasks that must be accomplished. The important feature of a Task is that it is defined as requiring some capabilities; in our domain, we focus on ISR capabilities, but the approach generalises to other forms of capability too [11]. We define the requires relation to associate an individual Task to the individual Capability instances that it requires. Because the definitions of mission, operation and task are somewhat subjective in practice, we adopt a simple model of hierarchical trees of tasks, where a Task can be broken down into sub-tasks (which are also

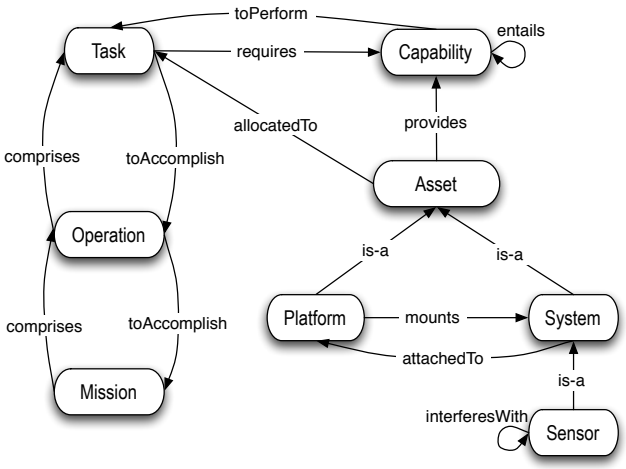

Fig. 2. Main concepts and relations in the MMF ontology

Tasks). These definitions give us flexibility in specifying the required capabilities of a mission, and aim to be broadly compatible with a range of mission-planning approaches. The set of capabilities required by a task T (at any level) is the aggregation of all capabilities required by T itself or by any subclass of T:

$$\mathsf{requires}(T', C) \wedge \mathsf{subTaskOf}(T', T) \rightarrow \mathsf{requires}(T, C)$$

where subTaskOf is a transitive object property [13].

On the right hand side of Figure 2 we have concepts related to capability-provision ("means" in MMF terms), with a focus on the sensor-mission matching problem: Platform and Sensor are two kinds of Asset; a Sensor is a kind of System that can be attached to a Platform; inversely, a Platform can mount one or more Systems.[2] Some sensors can interfere with other sensors, so they cannot be used simultaneously. Assets provide capabilities, which provides the link to Tasks as discussed above. Moreover, a Capability can entail a number of more elementary capabilities[3]. At some point, assets will be allocated to specific tasks that require the capabilities provided by them: the object property allocatedTo allows us to construct solutions to the sensor-mission assignment problem.

Note that all concepts shown in Figure 2 are general MMF concepts with the exception of Sensor, which we have introduced (as a refinement of the MMF core concept System) in order to link the MMF ontology with the ISR domain-specific ontologies. This is because, while the MMF ontology describes the main concepts used in our matchmaking framework — and is generic to military and military-style missions and means in the widest sense — in order to describe specific instances of those concepts we need domain-specific vocabulary. There is already a sizeable amount of work done in providing descriptive schemas and ontologies for sensors, sensor platforms, and

[2]Note that the "is-a" relation in Figure 2 denotes the OWL sub-class property, which is transitive.

[3]Primitive capabilities are called *functions* in MMF, but we have simplified this.

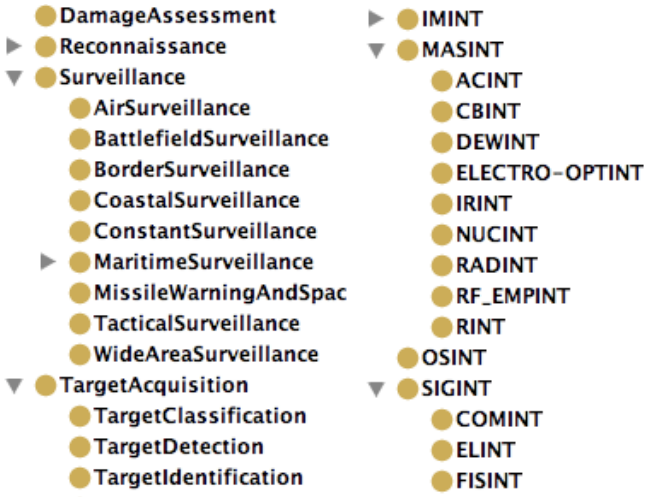

Fig. 3. Sample concept taxonomies relevant to the ISR domain: ISR tasks and INT types

their properties (e.g. [14], [15], [16]). There are also several well-known structured descriptions of tasks in the military missions context, most notably the Universal Joint Task List (UJTL)[17]. Based largely on these pre-existing sensor and task ontologies, we have identified a collection of concept hierarchies relevant to the ISR domain. Portions of two of these (defining ISR tasks and various types of intelligence information — "INT" types) are shown in Figure 3.

In these terms, we define a *bundle type* to be an intensional definition of a set of bundles of assets that can satisfy a task. The essential part of a bundle type is the specification of required sensor and platform types needed to provide the capabilities to satisfy the task. For this, we define a *platform configuration* $\pi = \langle P, \mathcal{S} \rangle$, where P is a type of platform, and $\mathcal{S} = \{S_1, \ldots, S_m\}$ is a set of sensor types that can be mounted on P simultaneously. Given a task T with a set of required ISR capabilities $\mathcal{C}^T = \{C_1, \ldots, C_n\}$, a single platform configuration is a *valid match* for T if the combined capabilities of P and $\mathcal{S}$ satisfy $\mathcal{C}^T$. Formally, $\mathcal{V}(T) = \{\pi_1, \ldots, \pi_n\}$ is the set of valid matches for task T iff $\langle P, \mathcal{S} \rangle \in \mathcal{V}(T) \Leftrightarrow (\forall C_i \in \mathcal{C}^T)((P \sqsubseteq \exists \mathsf{provides}.C_i) \sqcup (S_i \in \mathcal{S} \sqsubseteq \exists \mathsf{provides}.C_i))$.

More commonly, the requirements of a task will not be satisfied by a single platform configuration. We define a *package configuration* $\Pi = \{\pi_1, \ldots, \pi_n\}$, where π_i is a platform configuration. Π is a *valid match* for T if collectively the platforms and sensors in $\{\pi_1, \ldots, \pi_n\}$ satisfy the capabilities in $\mathcal{C}^T$, and Π is minimal with respect to $\mathcal{C}^T$ (there does not exist any subset of Π that is a *valid match* for T).

Bundle types are created by post-processing the package configurations, to add cardinality constraints. This is currently done using pre-defined configuration knowledge, for example:

- at least 1 UAV with at least 1 Camera
- at least 2 AcousticArrays with exactly 2 ACINTSensors

While this approach was conceptually simple, extensible (in terms of the modularity of the domain-specific ontologies), and well-founded (on MMF), it was rather limited in that it required task capabilities to be specified at a low level in order to drive the matching procedure. Specifically, the approach was over-reliant on the ontology of INT types shown in Figure 3 to determine the type of sensor required; for example, a choice of ACINT or IMINT in $\mathcal{C}^T$ would determine — rather trivially — the need for acoustic or imagery sensing). We sought a higher-level representation of ISR tasks, as described in the next section.

IV. Specifiying Tasks

Our main requirements for a higher-level representation of tasks were:

- That the task should as much as possible specify only *what* is the information requirement, and avoid saying anything about *how* it should be obtained. The reasons for this requirement were to avoid the main flaw in our earlier approach, and allow a greater degree of flexibility in allocating assets to tasks.
- That the representation of tasks should be familiar to potential users of the approach, not only to make any tool based on the approach easier to use, but also to maximise the acceptability of our approach (the same reason that we adopted MMF as the underlying framework for the task-asset matching).

Following guidance from subject-matter experts, we based our representation on the National Imagery Interpretability Rating Scale (NIIRS) for various kinds of imagery intelligence[18]. NIIRS is a well-established way to determine the kinds of data (for example, visible imaging, or radar) that are interpretable to answer particular information requirements (detecting, identifying and distinguishing various kinds of thing). The NIIRS scale defines ratings on a ten point scale (0–9) for various kinds of imagery data; for example, detection of vehicles of particular types is achievable by Visible NIIRS 4 and Radar NIIRS 6. This is useful because sensor/platform combinations can then be rated in terms of their NIIRS values, allowing analysts to seek data that are suitable for their tasks. From our point of view, NIIRS supports the kind of task-asset matching we established as described in the previous section.

We define a task as a quadruple, $\langle NC, DS, A, T \rangle$, where:

- NC is one of three basic "capabilities" defined in NIIRS: detect (find or discover the presence or existence of an entity), distinguish-between (determine that two detected entities are of different types), and identify (name an object by type);
- DS is a set of "detectable" types of entity (for example, people, vehicles, or installations at various levels of specificity);
- A defines an area-of-interest;
- T defines a period of time.

It is worth noting that, in accordance with our earlier work, we continue to refer to this representation as a "task" representation. In NIIRS, and in ISR more generally, these kinds of description are referred to as "capabilities". In a sense,

as our approach makes clear, they can be regarded as high-level capabilities, from which lower-level capabilities can be inferred. However, we prefer to locate these representations in our ontology as specialisations of the Task class rather than the Capability class to avoid confusion. Note also that, in the current implementation, we omit T as we assume that all tasks are required to happen in the same mission period.

Fig. 4. A portion of our ontology of "detectable" types of entity from NIIRS

Full details of the representation of the NIIRS scale, and our rules for performing reasoning with it, are given in [9]. The various "detectable" types of entity are defined in an ontology, drawn from the entities appearing in the military version of the NIIRS documentation. A portion of this ontology is shown in Figure 4. Using this ontology, we formalised the various NIIRS interpretation tasks as a set of clauses of the following form: $\langle NC, DS, FS, C, NT, NR\rangle$, where:

- NC is a NIIRS "capability" as above (one of: detect, distinguish-between, identify);
- DS is a set of detectables, as above;
- FS is a set of more specific features of the detectable entities (for example, the roads or guard posts of a base, the runways of an airport, or the piers and warehouses of a port) — these features are also defined in the ontology of "detectables";
- C is a context, defining the preconditions that must hold for the clause to apply (for example, detection of a ship in the context of open water, or identification of vehicles in a known motor pool); and
- NT is the type of imaging from the NIIRS scale (for example, Visible or Infrared); and
- NR is a NIIRS rating (for example, Visible-6 or Infrared-4).

Some examples:

- $\langle$detect, {Port}, {}, {}, Visible, Visible-1$\rangle$: a port can be detected using visible imagery of rating 1;
- $\langle$detect, {Port}, {Pier, Warehouse}, {}, Radar, Radar-1$\rangle$: a port can be detected based on the presence of piers and warehouses using radar imagery of rating 1;
- $\langle$distinguish-between, {Taxiway, Runway}, {}, {Airfield}, Visible, Visible-1$\rangle$: taxiways and runways can be distinguished at an airfield using visible imagery of rating 1;
- $\langle$detect, {Vehicle}, {}, {Motor-Pool}, Radar, Radar-4$\rangle$: vehicles can be detected in a known motor pool using radar imagery of rating 4;

The basic reasoning procedure for using these clauses is as follows (there are additional features described in [9]:

1) Given a task $T = \langle NC, DS, A, T\rangle$, find all clauses $\langle NC', DS', FS, C, NT, NR\rangle$ where $NC = NC'$ and $DS \subseteq DS'$ or $DS \subseteq FS$;
2) For each clause where C is not empty, check that all the conditions in C hold, normally by referencing mapping information for area A, and if they do not hold, discard the clause;
3) Gather all of the NT and NR elements from the remaining clauses to form the set of required capabilities, $\mathcal{C}^T$;
4) Because higher ratings of a given NIIRS type entail lower ones, we reduce $\mathcal{C}^T$ so that it contains only the highest rating of each NIIRS type.

We then use our previous approach to matching, described in Section III, with two important modifications:

- We have added the relevant NIIRS capabilities to the sensor and platform types. Because of the way NIIRS is defined, we associate the NIIRS imaging types NT with sensors (for example, Visible or Radar) and the NIIRS ratings NR with platforms (for example, Visible-1 or Radar-4). Thus, an inferred platform configuration must satisfy both the required imaging type and the rating.
- Because NIIRS gives us *alternative* ways to satisfy a task, each different NR and its corresponding NT is matched disjunctively; so, for example, if $\mathcal{C}^T$ contains Visible-1 and Radar-4, we generate separate sets of valid matches for each, and then aggregate them into a complete set of valid matches.

An important limitation of this approach is the published NIIRS scale is restricted to imagery intelligence. However, in principle, there is no reason why it cannot be extended to cover other types of intelligence also. For our purposes, we have extended our knowledge base where we know of non-imagery approaches that have been shown - at least in controlled trials - to have particular capabilities. For example, following [19] we introduced clauses that indicate acoustic data can be used to detect vehicles (note that we leave the rating essentially unspecified by using zero): $\langle$detect, {Vehicle}, {}, {}, Acoustic, Acoustic-0$\rangle$

In this section, we have shown how we extended our original matching approach to support higher-level and more realistic task representations. In the next section, we address the issue of asset allocation based on the output from the matching process.

V. Asset Allocation

To recap, the outcome of the matching process is a set of bundle types, each of which is an intensional definition of a set of bundles of assets that can satisfy a task (to some extent). A bundle type consists of a specification of required sensor and platform types needed to provide the capabilities to satisfy the task, and cardinalities on these. In order to allocate bundles of sensor and platform instances to a task, we now need to generate candidate bundles, and to find an allocation that satisfies some overall "goodness" criteria. In our earlier work, we have shown that the generic problem of assigning assets bundles to tasks is NP-hard, even to approximate [7].

One way of understanding our problem is as a combinatorial auction, that is a silent auction in which bidders (missions) can express preferences on bundles or *combinations* of items (assets) [20]. Given a fixed supply of goods, the goal of the winner determination problem [21] is to maximize revenue earned from the sale of disjoint item combinations. Given that there may be exponentially many bids, this is a difficult problem to solve and therefore much of the research focus has been on AI or algorithm-engineering approaches (for a survey, see [22]).

It has been argued that combinatorial auctions can, in practice, provide good approximate solutions within reasonable time for problem instances of reasonable size (in terms of the number of bids) [22]. To test if this was the case for our sensor-mission assignment problem, we modeled it using the combinatorial auction formalism and we used a well-known existing algorithm for general combinatorial auctions called CASS (Combinatorial Auctions Structured Search) [23] to solve it.

The sensor-mission assignment problem can be formulated as a combinatorial auction in which the bidders are missions $M_1, \ldots, M_m$, and the items are assets $A_1, \ldots, A_n$. Each mission is associated with a utility demand d_j, indicating the amount of sensing resources needed, and a profit p_j (defined below), indicating the importance of the mission. Each mission places a bid, equal to its own profit, for any set of assets which satisfy the mission's demand and respect the mission's budget constraint. Formally, we compute the *value* that bidder M_j obtains if it receives the bundle B of assets using the following *valuation function*:

$$v_j(B) = \begin{cases} p_j, & \text{if } u_j \geq d_j \text{ , } w_j \leq b_j \\ p_j \cdot u_j/d_j, & \text{if } T \leq u_j/d_j \text{ , } w_j \leq b_j \\ 0, & \text{otherwise} \end{cases}$$

Where:

- u_j represents the joint utility of the asset bundle B to the mission M_j. In cases where the joint utility of the bundle is simply additive (as in [10]), $u_j = \sum_{i \in B} e_{ij}$. Where bundles exhibit non-additive joint utility (as in [7]), we need to select an appropriate joint utility function based on the type of task and the bundle type.
- w_j is the total cost of B to M_j, given by $w_j = \sum_{i \in B} c_{ij}$ where c_{ij} is the potential asset assignment cost for asset A_i to mission M_j (for platform assets, this could reflect the costs associated with the platform and its deployment).
- T is a fractional satisfaction threshold designed to filter out bundles whose utility is marginal compared to mission demand.

Note that defining all of the above, and also the mission profit and demand, requires additional information beyond the knowledge-based representations of tasks, bundles, and assets described in the previous sections.

We list all the bids for each mission M_j as pairs $(B, v_j(B))$ and we use the OR* bidding language [24] to explicitly introduce mutual exclusion between bids placed by the same mission. The CASS algorithm then conducts a structured depth-first search on the list of all bids trying to find the subset of bids which maximize the total revenue under the constraint that each item can be allocated to at most one bidder.

The number of bids is potentially exponential if we adopt the naive approach of enumerating all nonzero-value bids containing bundles respecting the missions' demand and budget constraints (of order $2^n \times m$ bids in the worst case, but lower in practice). In our experiments with CASS, we adopted this naive approach to test if the number of bids generated within our scenario is computationally tractable. As a comparison, we also considered a constrained version of our problem in which we assume that missions behave as *single-minded bidders*, i.e. each mission bids only for the bundle of assets which evaluates to be the best bundle it can obtain (using the valuation function $v_j(B)$), therefore in total there will be at most m bids. In this version of the problem, the number of bids is linear. Each "optimal" bid is generated using the standard knapsack FPTAS [25]: for each mission we solve the knapsack problem of finding the subset of assets which maximizes the total mission profit using the profit function

$$p_j(u_j) = \begin{cases} p_j, & \text{if } u_j \geq d_j \\ p_j \cdot u_j/d_j, & \text{if } T \leq u_j/d_j \\ 0, & \text{otherwise} \end{cases}$$

while respecting the mission's budget constraint. Any assets assigned to a mission that has greater than 100% satisfaction, and which can be released without reducing the percentage below 100%, are then removed from the bundles, obtaining what can be considered the best asset bundle for each particular mission.

Our initial experiments with simulations comparable to those we used previously in [10] showed that the general CASS algorithm is not able to solve the problem in our scenario due to the exponential growth in the number of bids. However, if we have single-minded bidders the problem instance becomes computationally tractable using CASS.

VI. Implementation and Discussion

Since its previous presentation in [5], we have extended our pilot application, SAM (Sensor Assignment to Missions), to incorporate the higher-level task representation described in

Section IV. A screenshot of the new user interface is shown in Figure 5. As before, a user logs-in as a member of a coalition (in our example, we have a US/UK coalition). The user is able to select one or more areas-of-interest on a map (left panel) and, for each, to select multiple ISR tasks (top-right panel) using our NIIRS-based representation. (As we noted in Section IV our approach is currently simplified to assume that all tasks are required at the same time; this could easily be extended.)

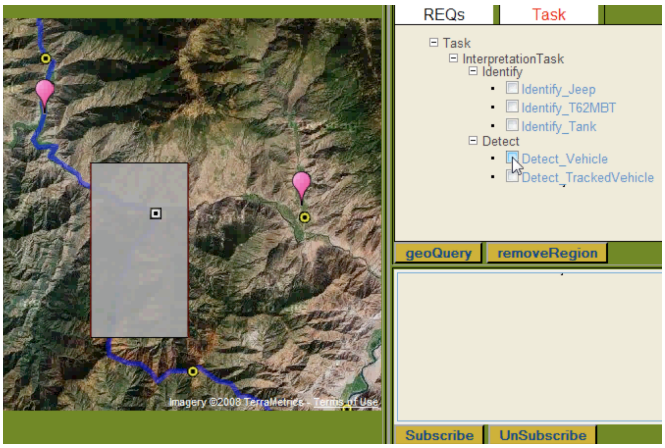

Fig. 5. Setting an area-of-interest and selecting tasks using the SAM application

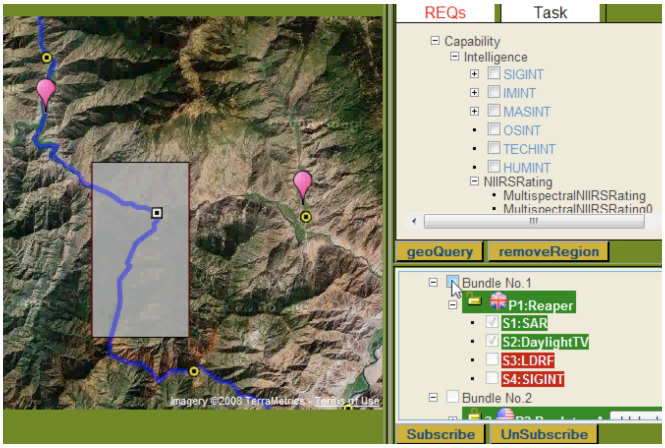

Fig. 6. Selecting an asset bundle manually using the SAM application

SAM is implemented as a Web application in Java. The NIIRS-based reasoning to infer capabilities from the selected tasks is implemented as a rule-based system in Java (for details, see [9]. The matching procedure is implemented using the Pellet reasoner [26]. Before performing reasoning, the SAM application queries inventory catalogues to determine what types of platforms and sensors are actually available, so it can recommend package configurations including the most specific types that are potentially deployable. Once the package configurations have been found, SAM retrieves all asset instances of the appropriate types from the inventory catalogues, and generates all asset bundles specified by using the package configurations as bundle types, together with additional cardinality information as noted at the end of Section III. Because the number of assets is small in our demonstration, the set of bundles is currently presented to the user, to make the choice of preferred bundle. This step is shown in Figure 6.

In doing this, SAM takes into account access policies on these resources, including ownership (note that the figure shows UK/US ownership of assets in the bottom-right panel) and whether the user has sufficient privileges to task those assets (shown by the "lock" icon next to the asset types). While simple, this mechanism is intended as an expansion point to allow the incorporation of more sophisticated access policies in future, such as those described in [4]. Once an asset has been selected, SAM allows the user to "subscribe" to available data feeds from that asset, using the ITA Sensor Fabric [27]. Additionally, if an asset fails, SAM allows the user to "backtrack" and make a different choice of bundle, and then obtain data that satisfies their task in some other way. For example, the user might initially choose to satisfy the vehicle detection task (Figure 5) by imagery data to be obtained by a UAV (Figure 6); then, if the UAV fails for some reason, they could opt instead to solve the task using acoustic data from, for example, an acoustic array.

While the current version of SAM demonstrates the concept of knowledge-driven sensor-mission assignment, it does not yet incorporate automated asset assignment, as it assumes an inventory with a relatively small number of assets. The combinatorial auction mechanism is currently implemented separately. Integrating these two components requires a number of choices to be made, most importantly:

- Users need a way to specify mission/task "profit" in a way that is meaningful to them. Potentially this could be done using a simple indication of priority. In principle, it ought to be part of a more general plan representation formalism. An open question here is whether priority can in some cases be inferred from the context of the task (for example, the type of mission: hostage rescue would normally take priority over environmental monitoring).
- Further research is necessary to determine where are the most appropriate "choice points" for user intervention. It is clearly only feasible for users to select bundle instances when there are very few choices available. In more typical cases, it would be more sensible to allow the user to select (or rank) alternative bundle types (or package configurations), this giving a (perhaps "soft") preference as to how they would prefer to obtain their data.
- SAM is intended to be used in a distributed fashion, as explained in [5]. Multiple users in a coalition submit their tasks using instances of SAM, and we need to examine alternative ways to apply the allocation algorithms in this context. We have investigated distributed versus centralised algorithms in the context of homogeneous sensor

networks [10]; further work is required to determine how best to address this problem for heterogeneous networks.

VII. Conclusion

In this paper, we presented our knowledge-driven solution to the sensor-mission assignment problem, proceeding from a high-level specification of information requirements, to the allocation of assets such as sensors and platforms. We showed how our original ontology-based approach — founded on a formalisation of the military missions and means framework — has been extended to provide: (1) a richer and more realistic way for a user to specify their information requirements, by means of a rule-based representation of the NIIRS approach; and (2) efficient asset allocation using a well-known efficient combinatorial auction algorithm (CASS). Finally, we summarised the status of our pilot application, SAM, and discussed the various roles such an application can play in supporting sensor-mission assignment.

Acknowledgments

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this paper are those of the author(s) 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 authorised to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] J. H. Sheehan, P. H. Deitz, B. E. Bray, B. A. Harris, and A. B. H. Wong, "The military missions and means framework," in *Proceedings of the Interservice/Industry Training and Simulation and Education Conference*, 2003, pp. 655–663.

[2] S. J. Tutton, "Optimizing the allocation of sensor assets for the unit of action," Naval Postgraduate School, California, Tech. Rep., 2006, master Thesis.

[3] Joint Publication, "2-01: Joint and national intelligence support to military operations," 2004. [Online]. Available: http://www.dtic.mil/doctrine/jpintelligenceseriespubs.htm

[4] T. Pham, G. Cirincione, D. Verma, and G. Pearson, "Intelligence, surveillance, and reconnaisance fusion for coalition operations," in *Proc 11th International Conference on Information Fusion*, 2008.

[5] A. Preece, D. Pizzocaro, K. Borowiecki, G. de Mel, M. Gomez, W. Vasconcelos, A. Bar-Noy, M. P. Johnson, T. L. Porta, and H. Rowaihy, "Reasoning and resource allocation for sensor-mission assignment in a coalition context," in *Proc MILCOM 2008*, 2008.

[6] M. Gomez, A. Preece, M. Johnson, G. de Mel, W. Vasconcelos, C. Gibson, A. Bar-Noy, K. Borowiecki, T. L. Porta, D. Pizzocaro, H. Rowaihy, G. Pearson, and T. Pham, "An ontology-centric approach to sensor-mission assignment," in *Proc 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2008)*, 2008.

[7] H. Rowaihy, M. Johnson, D. Pizzocaro, A. Bar-Noy, L. Kaplan, T. L. Porta, and A. Preece, "Detection and localization sensor assignment with exact and fuzzy locations," in *Proc 5th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS'09)*, 2009.

[8] A. Preece, M. Gomez, G. de Mel, W. Vasconcelos, D. Sleeman, S. Colley, G. Pearson, T. Pham, and T. L. Porta, "Matching sensors to missions using a knowledge-based approach," in *SPIE Defense Transformation and Net-Centric Systems 2008 (SPIE Defense and Security Symposium)*, 2008.

[9] G. de Mel, M. Sensoy, W. Vasconcelos, and A. Preece, "Flexible resource assignment in sensor networks: A hybrid reasoning approach," in *1st International Workshop on the Semantic Sensor Web (SemSensWeb 2009)*, 2009.

[10] M. Johnson, H. Rowaihy, D. Pizzocaro, A. Bar-Noy, S. Chalmers, T. L. Porta, and A. Preece, "Frugal sensor assignment," in *Proc 4th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS'08)*, 2008, pp. 219–236.

[11] P. J. Tanenbaum and W. P. Yeakel, "A framework linking military missions and means," in *Mathematics for Industry: Challenges and Frontiers*. SIAM, 2005.

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

[13] G. Antoniou and F. van Harmelen, *A Semantic Web Primer, 2nd edition*. MIT Press, 2008.

[14] M. Botts, A. Robin, J. Davidson, and I. Simonis, "OpenGIS sensor web enablement architecture document," Open Geospatial Consortium Inc, Tech. Rep., 2006, OpenGIS Discussion Paper.

[15] D. Russomanno, C. Kothari, and O. Thomas, "Building a sensor ontology: A practical approach leveraging ISO and OGC models," in *Proceedings of the International Conference on Artificial Intelligence*, 2005, pp. 637–643.

[16] L. Bermudez, J. Graybeal, and R. Arko, "A marine platforms ontology: Experiences and lessons," in *Proceedings of the ISWC 2006 Workshop on Semantic Sensor Networks*, Athens GA, USA, 2006.

[17] UJTL Ontology, http://orlando.drc.com/semanticweb/daml/ontology/condition/ujtl/condition-ont.

[18] National Imagery Interpretability Rating Scale (NIIRS), http://www.fas.org/irp/imint/niirs.htm.

[19] B. Guo, Y. Wang, P. Smart, N. Shadbolt, M. Nixon, and T. Damarla, "Approaching semantically-mediated acoustic data fusion," in *Proc IEEE MILCOM)*, 2007.

[20] J. Abrache, T. G. Crainic, M. Gendreau, and M. Rekik, "Combinatorial auctions," *Annals of Operations Research*, vol. 153, no. 1, pp. 131–164, 2007.

[21] M. H. Rothkopf, A. Peke?, and R. M. Harstad, "Computationally manageable combinational auctions," *Management Science*, vol. 44, no. 8, pp. 1131–1147, Aug. 1998, ArticleType: primary_article / Full publication date: Aug., 1998 / Copyright 1998 INFORMS. [Online]. Available: http://www.jstor.org/stable/2634691

[22] S. de Vries and R. Vohra, "Combinatorial auctions: a survey," *INFORMS J. on Computing*, vol. 15-3, pp. 284–309, 2003.

[23] Y. Fujishima, K. Leyton-Brown, and Y. Shoham, "Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches," in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 548–553. [Online]. Available: http://portal.acm.org/citation.cfm?id=646307.687748

[24] N. Nisan, "Bidding and allocation in combinatorial auctions," in *Proceedings of the 2nd ACM conference on Electronic commerce*. Minneapolis, Minnesota, United States: ACM, 2000, pp. 1–12. [Online]. Available: http://portal.acm.org/citation.cfm?id=352871.352872

[25] V. V. Vazirani, *Approximation Algorithms*. Springer, 2001.

[26] Pellet OWL DL Reasoner, http://pellet.owldl.com/.

[27] F. Bergamaschi, D. Conway-Jones, C. Gibson, and A. Stanford-Clark, "A distributed test framework for the validation of experimental algorithms using real and simulated sensors," in *First Annual Conference of the International Technology Alliance (ACITA 2007)*, 2007.

Appendix - Glossary of Acronyms

ACINT	Acoustic Intelligence
ISR	Intelligence, Surveillance and Reconnaissance
IMINT	Imagery Intelligence
MMF	(Military) Missions and Means Framework
MSR	Main Supply Route
NIIRS	National Imagery Interpretability Rating Scale
OWL	Web Ontology Language
UAV	Unmanned Aerial Vehicle
UJTL	Universal Joint Task List