Flexible Resource Assignment in Sensor Networks: A Hybrid Reasoning Approach

Geeth de Mel[1], Murat Sensoy[1], Wamberto Vasconcelos[1], and Alun Preece[2]

[1] Department of Computing Science, University of Aberdeen,
Aberdeen AB24 3UE, Scotland, United Kingdom
{g.demel,m.sensoy,w.w.vasconcelos}@abdn.ac.uk
[2] Cardiff School of Computer Science, Cardiff University,
Queen's Buildings, 5 The Parade, Roath,
Cardiff CF24 3AA, United Kingdom
A.D.Preece@cs.cardiff.ac.uk

Abstract. Today, sensing resources[3] are the most valuable assets of critical tasks (e.g., border monitoring). Although, there are various types of assets available, each with different capabilities, only a subset of these assets is useful for a specific task. This is due to the varying information needs of tasks. This gives rise to assigning useful assets to tasks such that the assets fully cover the information requirements of the individual tasks. The importance of this is amplified in the intelligence, surveillance, and reconnaissance (ISR) domain, especially in a coalition context. This is due to a variety of reasons such as the dynamic nature of the environment, scarcity of assets, high demand placed on available assets, sharing of assets among coalition parties, and so on. A significant amount of research been done by different communities to efficiently assign assets to tasks and deliver information to the end user. However, there is little work done to infer sound alternative means to satisfy the information requirements of tasks so that the satisfiable tasks are increased. In this paper, we propose a hybrid reasoning approach (viz., a combination of rule-based and ontology-based reasoning) based on current Semantic Web[4] technologies to infer assets types that are necessary and sufficient to satisfy the requirements of tasks in a flexible manner.

Key words: Sensors, Platforms, Resource Assignment, Semantic Web, Rules, Hybrid Reasoning

1 Introduction

A sensor network [1] is a collection of heterogeneous sensing resources[3], composed of sensors and platforms. Sensors capture phenomena whereas platforms provide the durability, mobility, communication capabilities, and so, on to the mounted sensor(s). Advances in technology have made the deployment of sensor networks

[3] A sensing resource (henceforth referred to as an "asset") is a platform which contains one or more sensors.
[4] http://www.w3.org/2001/sw/

a robust and viable solution to reliably monitor and obtain timely, continuous, and comprehensive observations about dynamic situations [17, 19]. Therefore, for many critical tasks like border monitoring or surveillance, selection of sensing assets for tasks play a key role in their success or failure. This leads to the problem of assigning proper assets to tasks such that the assigned assets cover the information needs of the individual tasks.

Effective and efficient assignment of assets to such tasks is an important but computationally hard problem in sensor networks domain. The difficulty of this problem is amplified in the intelligence, surveillance, and reconnaissance (ISR) domain, and especially in a coalition context, where the assets belonging to different parties are shared to archive tasks. This is due to a variety of reasons. First, the environments in which these resources are deployed could rapidly change (i.e., new high-priority tasks emerge, assets become unreliable, weather conditions change, and so on) yielding new information requirements or assets requirements. Second, the demand placed on available assets typically exceeds the inventory [14] resulting in complex assignment choices. Last but not the least, the inability to obtain a bird's-eye view of the available assets to tasks makes it impossible to perform assignments in an informed manner. All these reasons imply the necessity to infer sound alternative means to satisfy the information requirements of tasks so that the different capabilities provided by assets can be used to cover the information requirements of tasks properly, thus increasing the number of satisfiable tasks.

Many communities have investigated the assignment problem and proposed different mechanisms that could be applied to solve it. Some of these approaches rely on having a *human in the loop* to decide which assets are appropriate to satisfy the requirements of tasks [4] whereas other approaches have tried to automate the assignment process [5, 13, 22]. However, these automated approaches are highly constrained in terms of their assumptions. For example, the work discussed in [5] assumes an unlimited inventory of assets, whereas [13] assumes assets to be of the same type (i.e., any assets could provide some utility to a task). This is not the case in general and especially in the environments highlighted above. Assets are heterogeneous (different capabilities, operational conditions etc.) by nature and only suitable for particular tasks.

Most of the current approaches have ignored important qualitative attributes such as the capability provided by assets, prevailing weather conditions, etc. These attributes play a major role in deciding which assets could be deployed to achieve the information needs of tasks. Moreover, important many-to-many relationships between assets and tasks (i.e., a task could be accomplished in several different ways; an asset could be used to achieve several different kinds of tasks) are not considered. We argue that considering these relationships allows agile management of information providing assets by enabling reasoning about different capabilities of assets and requirements of tasks.

In this paper, we propose knowledge-rich models and mechanisms based on Semantic Web[4] technologies to address the issues highlighted above. We propose a rule-based system to infer multiple capabilities that could be used to satisfy the information requirements of tasks. We then discuss an ontology-based reasoning framework to identify suitable asset types that meet those identified capabilities, thus increasing the flexibility of the assignment. We present tools that are built around these models to assist the decision makers in the assignment process in order to identify suitable asset types for tasks. The proposed system not only recommends asset types in an agile manner but also guarantees the soundness of the solution inference process.

The rest of this document is structured as follows. In section 2, we survey related research done in sensor networks and other domains that have inspired our work. Section 3 introduces a rule-based system that enables inference of different capabilities to satisfy the goals of tasks and gives some example outputs from the rule system. In section 4, we highlight an ontology-based matchmaking framework to infer sound solutions to the assignment problem based on the capabilities provided by the assets and the requirements advertised by the tasks. A case study, applying out approach is illustrated in section 5 and we conclude in section 6, also providing future directions for this work.

2 Related Work

As stated previously, different communities have proposed a variety of approaches to solve the problem of assigning assets to tasks. These approaches can be grouped and summarized as follows:
Algorithmic Approaches. Many approaches have proposed a utility-based solution with heuristics-based enhancements. For example, in [5] Byers and Nasser propose a framework to solve the assignment problem based on energy conservation to maximize the utility of a sensor network while keeping the cost of the assignment per task under a pre-defined budget. Johnson *et al.* propose an energy-aware approach to select assets for tasks in both static and dynamic environments [13] for competing tasks. One major drawback in these approaches is the fact that all assets are assumed to be of the same type. We argue that this is not the general case. Assets are heterogeneous (different capabilities, operational conditions etc.) by nature and only suitable for particular tasks.

In [22], Tatton proposes an approach to optimize the assignment of assets to task based on probability of target detection. In [8], Doll has further extended the *sensor allocation model* by introducing notion of probability of line of sight and field-of-view to the model in order to better estimate the asset performance. The drawback of these approaches is the assumption that there exists a classification that pre-identifies assets being suitable for some particular tasks in order to perform the assignment.
Semantic-based Approaches. In [23], Whitehouse *et al.* propose a framework based on semantics to allow users to perform declarative queries over a sensor network (i.e., rather than querying raw data, users query whether a vehicle is a car or a truck). A major drawback in this approach is the fact that all the desired inference units must be declared for a sensor network before users can start using the system. This is difficult, if not impossible, for a heterogeneous sensor network deployed in a dynamic situation. Also the declarative language

described in the work is not standardised (i.e., the language is described using Prolog [6] predicates) which hinders the extensibility of the system.

Recent research has considered standardised descriptive schema representations (e.g., XML [21], RDFS[5], OWL [7]) to assist in assets-to-tasks assignment [4]. The keystone of this approach is to have standardised schemas to describe assets, asset properties, and requirements. There is already a significant amount of work done in this area, for example XML-based approaches such as the OpenGeospatial Consortium (OGC)[6] suite of Sensor Web Enablement (SWE) [4] specifications to ontologies such as OntoSensor [10], the Marine Platforms Ontology [3], etc.

Lack of semantics in SensorML [18] (i.e., descriptions of assets and their capabilities are in plain text) makes it difficult to be used in automated capability inference mechanisms. *OntoSensor* [10] was created to assist in semantic data fusion. Therefore, a great deal of emphasis has been put on modelling the data from assets, but not their functional aspects. Hence, it cannot also be used as it is in capability inferences.

The proposed approach builds upon the existing standards and mechanisms for knowledge representation and reasoning in order to enable semantic-aware assignment of assets to tasks. In the next section we propose a knowledge-based rule system to address the issue of inferring different capabilities that can satisfy the same information requirements of tasks.

3 Agile Inference of Capabilities: A Rule-based Approach

In an environment where there are many-to-many relationships between tasks and assets, it is prudent to allow tasks' requirements to be specified in manner that is independent of specific capabilities of assets. For example, in a surveillance task rather than asking for *infrared capability* one could specify the information requirement for *detecting vehicles*. Let us assume that, according to the available inventory, detecting a vehicle could be done with infrared, radar, or acoustic capabilities, thus, yielding multiple degrees of freedom in (re)assignment of assets to tasks.

We propose a rule-based system to address this issue. The proposed system allows users to describe what they want to achieve (e.g., detect vehicles, identify a particular building, etc.) and use the rule-based system to infer the different capabilities that could be used to achieve tasks. In order to infer the required capabilities to achieve tasks, tasks must be formalised with respect to the capabilities that are required to achieve them. There are many knowledge corpora that provide adequate information about the different capabilities required to achieve the same task. In the next subsections, we discuss one of these knowledge corpora and show how we have formalised it so that different capabilities can be inferred to satisfy the same task.

[5] http://www.w3.org/TR/rdf-schema/
[6] http://www.opengeospatial.org

3.1 National Image Interpretability Rating Scale (NIIRS)

NIIRS[7] is an approach embraced by intelligence and civilian communities to express the information potential of different image types [12]. NIIRS is defined for visible, infrared, radar, and multispectral imagery, providing a 10-level scale with each level containing several interpretation tasks or criteria. Within each spectrum higher NIIRS levels inherit the criteria of their subordinates. For example, with a NIIRS-3-rated image one can satisfy criteria set out by NIIRS 1 and 2.

The criteria indicate the expressivity of an image in terms of the amount of information that could be extracted from it at the given scale. For example, with visible NIIRS 4, identification of individual tracks is possible whereas with an image of visible NIIRS 6, identification of a vehicle is made possible (i.e., the make/model of the vehicle can be identified). Additionally a task can be achieved using different spectra with different NIIRS values. For example, detecting a large aircraft could be done with infrared and visible imagery using ratings 2 and 3 respectively. The image classification criteria could be broadly categorized as detect[8], distinguish[9], and identify[10]

In section 3.3 we show how we formalised the NIIRS knowledge corpus. In order to formalise NIIRS, first we need to come up with a classification of the elements in the environment. In the next section, we discuss a possible representation for this classification using an OWL-DL ontology.

3.2 Detectable Ontology

Let us first introduce the notion of *detectable*: detectable are the objects (e.g., vehicles, building, people and so on) of interest. For example let us take the task *detect large buildings (e.g., hospitals, factories)*. In this case detectables can be classified as buildings. Let us take another example task *detect individual vehicles in a row at a known motor pool*. Vehicles are the detectables in this example.

We have created a detectables ontology to represent these concepts. Figure 1 shows a fragment of the taxonomies we have developed. We have used these concepts in the formalization of the criteria described in NIIRS, as we explain in Section 3.3. As Figure 1 shows, detectable concepts are broadly categorized into *Area*, *Component*, *Equipment*, *LinesOfTranspotation*, *Platform*, *Sensor*, and *Structure*. We have classified other detectable concepts as subclasses of these main concepts. A *Car* which is a subconcept of *WheeledVehicle* is a *GroundPlatform* (i.e., Figure 1(b)). *SiteConfiguration* represents a collection of buildings whereas *SiteComponent* refers to an individual building such as *Pier*, *Hanger*

[7] http://www.fas.org/irp/imint/niirs.htm

[8] Ability to find or discover the presence of an item of interest, based on its general shape, contextual information, etc.

[9] Ability to determine that two detected objects are of different types or classes based on one or more distinguishing features

[10] Ability to name an object by type or class, based primarily on its configuration and detailed components

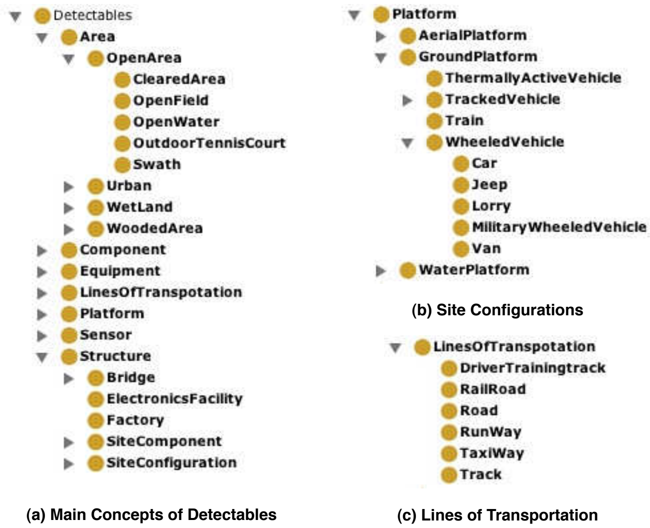

Fig. 1. Taxonomy of "Detectables"

or part of a building such as *BoilerHall*. *Factory* is not classified under either of them since it could be a single building structure or a multiple building configuration. Also we have introduced an object property *hasFeature* to describe the distinctive features of *Detectables*. For example, piers and hangars are both detectable concepts but they also are parts of a port, which makes them features of the port. Furthermore, this classification helps us formally define the concepts $detectable$, $distinguishable$, and $identifiable$:

1. **Detectable**: If the concept of interest has any sub-concept then it is detectable (e.g., *WheeledVehicle*).
2. **Distinguishable**: If a set of concepts are detectable, then they are also distinguishable. For example, if we detect a *Jeep* and a *Car*, then we can distinguish between them based on their shape.
3. **Identifiable**: If the concept of interest has no sub-concepts, then it is identifiable. For example, one can say a *SAAB 9-3 sedan* is identifiable.

3.3 Formalisation of Interpretation Tasks

We define a criterion as a 6-element tuple $FIT(T, W, F, C, I, V)$, where T represents the type of the *interpretation task* to perform (e.g., detect, distinguish, identify, and so on); I is the type of capability/intelligence (e.g., imagery spectra in NIIRS) that could be used to perform the interpretation task; $W = \{w_1, w_2, \ldots, w_i\}$ is a set of detectables (e.g., {*port, hospital*}) that can be observed using the capability/intelligence I; $F = \{f_1, f_2, \ldots, f_j\}$ is a set of features (e.g., {*pier, warehouse, loading bay, ambulance*}) describing W; C represents the context of the detectables; V is a numeric value that represents the quality of the intelligence (e.g., the rating of an imagery source in NIIRS). Below we provide examples of this formalism based on NIIRS criteria.

With an image rated *Visible NIIRS 1*, one can *detect* a medium-sized port facility and/or distinguish between taxi-ways and runways at a large airfield [12]. So, from this criterion, we can derive if there is a port facility in the image then one can detect it. Also according to the *Radar NIIRS 1*, one can *detect* a port facility based on its features (i.e., presence of piers and warehouses). Example 1 and Example 2 shortly describe how tasks could be presented in our formalism to exploit $features$ and *context* of criteria while inferring different capability.

Example 1 The task of detecting a port can be formalised as *FIT(detect,* {*Port*}*,* {}*,* {}*, image(Visible), 1)*. In this case, a reasoner can infer detection of a port can be achieved by using *Visible NIIRS 1*. However, in many cases, using explicit features of ports (e.g., piers and warehouses), we can detect objects more accurately. Therefore, the representation *FIT(detect,* {*Port*}*,* {*Pier,Warehouse*}*,* {}*, image(Radar), 1)* allows a reasoner to use some explicit features of a port while detecting it.

Example 2 Some tasks are highly sensitive to the context. For example, distinguishing between a taxiway and a runway using imagery intelligence can only be achieved if the context of the task enables clear images to be taken. If the context is *airfield*, which means that the task will be executed over an airfield, it is possible to distinguish between a taxiway and a runway. This can be represented as *FIT(distinguish,* {*Taxiway,Runway*}*,* {}*,* {*AirField*}*, image(Visible), 1)*. Similarly, to detect individual vehicles in a row at a known motor pool using radar intelligence, we have *FIT(detect,* {*Vehicle*}*,* {}*,* {*Motor-Pool*}*, image(Radar), 4)*.

We believe the proposed FIT formalism can be used to formalise knowledge from other intelligence domains too. For example, Guo *et al.* [11] propose an approach to detect and distinguish vehicles based on their *acoustic signatures*. Therefore, detect and distinguish tasks in our framework can also be formalised using *acoustic signatures* instead of NIIRS. In this case, if an acoustic signature of value 5 enables us to detecting a vehicle, we should formalise our statement as *FIT(detect,*{*Vehicle*}*,*{}*,*{}*,5,Acoustic)*.

An extensive knowledge base has been created using the representation above by formalizing the NIIRS corpus. In the next section, we present a set of rules that are implemented to draw conclusions from this knowledge base to find diverse but feasible set of capabilities to perform a task. This makes the assignment of assets to tasks more flexible and agile; we reason about multiple ways in which assets can satisfy the requirements of a task.

3.4 Rules to Derive Capabilities

In this section, we present a set of rules to make inferences from the created knowledge base using the FIT formalism. These rules derive minimal, but necessary and sufficient capabilities needed to achieve a particular task. For example, let X be a set of objects that need to be observed. Detecting an element $x_i \in X$ is defined using the rules below.

$$
\begin{aligned}
detect(x_j, i_j, v_j) &\leftarrow distinguish(x_j, i_j, v_j) &(1)\\
distinguish(x_j, i_j, v_j) &\leftarrow identify(x_j, i_j, v_j) &(2)\\
detect(x_j, i_j, v_j) &\leftarrow FIT(detect, w, f, c, i_j, v_j) \ \wedge \ x_j \in w &(3)\\
identify(x_j, i_j, v_j) &\leftarrow FIT(identify, w, f, c, i_j, v_j) \ \wedge \ x_j \in w &(4)\\
distinguish(x_j, i_j, v_j) &\leftarrow FIT(distinguish, w, f, c, i_j, v_j) \ \wedge \ x_j \in w &(5)
\end{aligned}
$$

These rules can be interpreted as follows. Rule 1 states that the object of interest x_i can be detected using intelligence i_j and the quality of intelligence v_j (corresponds to ratings in $NIIRS$ terminology) if it can be distinguished using i_j and v_j. Similarly, Rule 2 states that x_i can be distinguished using i_j and v_j if it can be identified using i_j and v_j. Rules 3, 5 and 4 state that you can detect, identify or distinguish an object x_i if you can find a related FIT statement in which x_i is a member of the set w of detectables declared in the statement.

3.5 Example Results

We have developed a proof-of-concept prototype using CIAO Prolog[11] to show how these rules draw conclusions from the knowledge base. For this purpose, we first query the system for required capabilities of the tasks *detect*, *distinguish*, and $identify$. Then, in this section we summarize the inferred capabilities. For example, a query to *detect a large airplane* returns the following result set.

```
?- detect(largeAirliner,Results).
Results = [(image(infrared),2),(image(radar),2),(image(visible,3))]
```

The inferred solution recommends three capabilities that could be used to perform the task using one of *visible*, *infrared*, or *radar* imagery with a minimum $NIIRS$ of 3, 2, and 2 respectively. However, *detection of a small airplane* can only be achieved using an infrared imagery with a minimum $NIIRS$ of 3.

```
?- detect(smallAirliner,Results).
Results = [(image(infrared),3)]
```

Therefore, according to the definitions of the interpretation tasks, *distinguishing between a large plane and a small plane* could only be done using infrared image with a minimum $NIIRS$ of 3. This is because, infrared $NIIRS$ 3 is the smallest common denominator in the above two queries to detect a large plane and a small plane. Below is the result of the query that confirms the expected result.

```
?- distinguish([largeAirliner,smallAirliner],Results).
Results = [(image(infrared),3)]
```

[11] http://clip.dia.fi.upm.es/Software/Ciao/

4 Capability-Requirement Matching

In [9, 16], we proposed the *Sensor Assignment to Missions* (SAM)[12] framework to improve asset-to-task assignments based on current Semantic Web technologies together with semantic matchmaking [15]. The core of the approach is a set of interlinking ontologies to describe scenarios (i.e., missions, operations, tasks), assets (i.e., sensors and platforms), capabilities of the assets, and the requirements of the tasks. These ontologies are represented in OWL DL [7].

This approach was inspired by the Missions and Means Framework (MMF) [20]. MMF was developed by the US Army Research Laboratory to provide means for specifying a military mission in order to evaluate the utility of alternative means (i.e., assets) to accomplish the goals. Based on MMF we have defined an architecture to infer the types of assets that are fit for the purpose (i.e., can meet the information requirements of the task). We use semantic reasoning and a matchmaking mechanism to derive these asset types. Figure 2 depicts the architecture of the system.

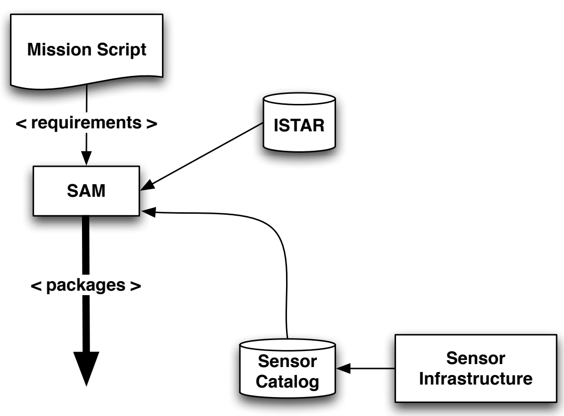

Fig. 2. SAM architecture

The architecture is composed of two main components, SAM the reasoner and the sensor infrastructure, and some data sources (viz., ISTAR ontology, and sensor catalogue). The *ISTAR*[13] ontology represents the domain knowledge of intelligence, surveillance, target acquisition, and reconnaissance aspects (e.g., types of intelligence). Figure 3 depicts the main concepts of the ISTAR ontology. The left-hand side decomposes a mission into a collection of tasks with specific information requirements (e.g., surveillance) and the right-hand side represents capabilities provided by assets (e.g., target detection provided by an UAV) as a

[12] http://www.csd.abdn.ac.uk/research/ita/sam
[13] http://www.csd.abdn.ac.uk/research/ita/sam/downloads/ontology/ISTAR.owl

composition of the functions provided by sensors and platforms. Requirements of tasks are broadly categorized into two sections: intelligence (i.e., kinds of intelligence disciplines such as imagery intelligence) and operational (i.e., desired capabilities of a task such as constant surveillance) requirements.

The *sensor catalogue* contains the attributes of assets (i.e., location, energy, current status, and so on.). These assets are particular instances of the asset types described in sensor and platform ontologies of the ISTAR ontology. The attributes of assets are retrieved from a *sensor infrastructure* [2].

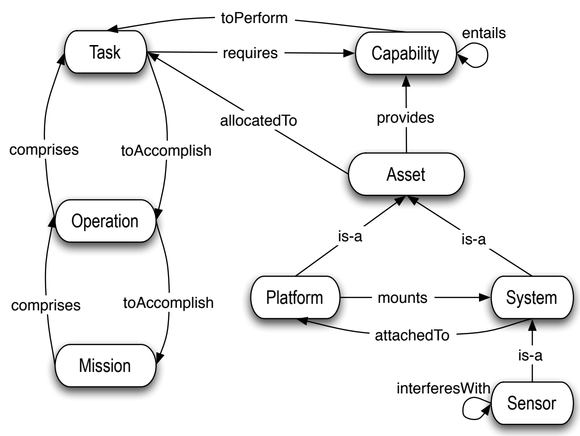

Fig. 3. Main concepts and relations in the ISTAR ontology. Reproduced from [9]

The reasoner checks the requirements of a given task and suggests asset types that are feasible and logically sound for the task. These solutions are logically sound due to the logical properties of OWL-DL [7] and the inference mechanisms used. We use Pellet[14] as a DL reasoner for inferences. Some solutions recommended by the reasoner are collection of asset types. This is because a task may not be satisfied with only one asset. For example, to achieve the goals of the task, visual and audio information are needed but there is no single asset to provide both. *SAM* uses a set-covering algorithm to compute this. Since a solution may contain more than one asset type, we refer to a solution collectively as an asset package. Furthermore, using subsumption[15] relationships, the reasoner finds all the plausible assets types for a particular task. We believe these solutions can be used in many useful ways, such as to analyse the feasibility of a mission with respect to an assets inventory, to assist in planning and re-planning stages of the mission, and so on.

[14] http://clarkparsia.com/pellet/

[15] A concept A subsumes a concept B if the definitions of A and B logically imply that members of B must also be members of A.

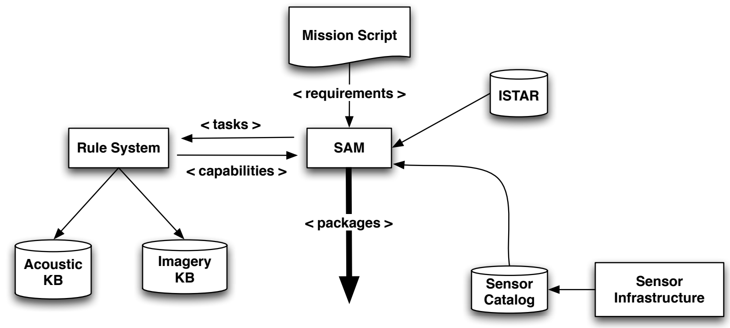

Fig. 4. SAM architecture with integrated rule system

We have extended the SAM architecture by incorporating the rule system discussed in Section 3 as shown in Figure 4. With the resulting integrated system, users can specify their information needs at higher level. That is, they do not have to express every capability requirement of a task explicitly; instead they simply let the rule system infer multiple capabilities in which the task could be accomplished. These inferences allow the system to compute many different asset types that could be used to satisfy the requirements of a given task.

5 A Case Study

In this section, we introduce an example scenario and demonstrate how the system proposed in Section 4 computes feasible asset types for tasks in a realistic situation. Let us suppose a mission where an international peacekeeping force has to maintain a safe corridor between two countries. In order to perform this mission, many operations need to be carried out. Let one of those operations be "Perimeter Surveillance", which could be broken down into a set of tasks. Some possible tasks for the operation are:

1. **Detect human activity in the region.** This task is a part of the operation because a suspicious gathering near or in the region of the safe corridor may imply a critical breach in perimeter.
2. **Detect vehicle movement.** This may imply the movement of troops or militia.
3. **Identify vehicle of particular type.** For example armoured vehicles might imply an imminent treat.

Let us consider the task *identify vehicle*. A high-level requirement of identifying a vehicle task could be *identifying jeeps*. The *SAM* tool discussed in section 4 allows users to specify their requirements in this manner (e.g., *detect vehicles, identify jeeps,* etc.) as shown in figure 5.

When the *SAM* tool receives such requirements, they are automatically passed onto the rule system discussed in section 3.4. Within the rule system, the

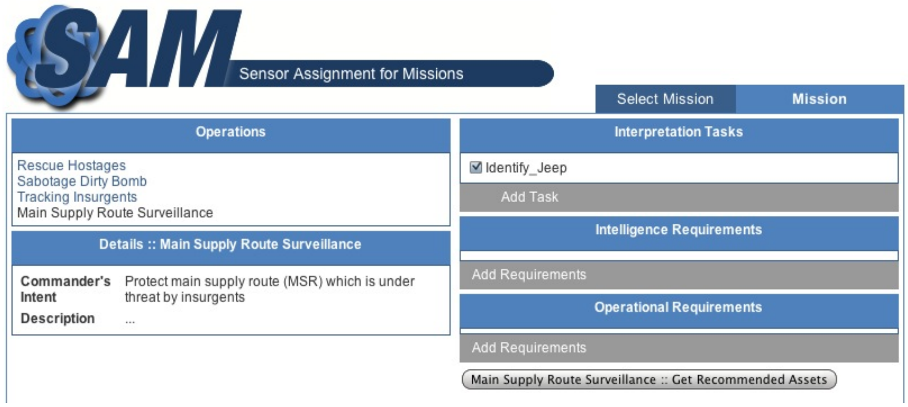

Fig. 5. SAM tool

appropriate rule is executed (e.g., *detect* rule is fired for detect activities whereas for identifying activities, *identify* rule is fired.). The rule traverses through the knowledge-bases (KBs) known to the rule system, and infer minimum capability ratings required to satisfy requirements. These KBs are created with respect to the formalism described in section 3.3. In order to satisfy the requirement *identify jeeps*, the rule system derives **{VisibleNIIRSRating6, RadarNIIRSRating6, ACSignature7}** as the required ratings.

This result set represents the fact that, in order to identify a jeep, one needs assets that could either provide visual, radar, or acoustic capability at a particular rating or above. These results are handed back to the *SAM* tool as shown in figure 4. *SAM* tool then passes these results to the ontology-based reasoner to identify the potential assets types that satisfy these capability requirements. It is important to note that these capability ratings are provided by an asset: sensors provide the capabilities such as radar, acoustic whereas platforms provide the capabilities such as altitude, range capabilities required to compute a particular rating. We represent this using the following logical formula.

$$\begin{aligned}\textbf{Asset([P,S]):providesCapabilityRating([C,R])} \leftarrow\ & \textbf{Platform(P):canProvideRating([C,R])} \wedge \\ & \textbf{Platform(P):carriesSensor(S)} \wedge \\ & \textbf{Sensor(S):providesCapability(C)}\end{aligned}$$

Therefore, in order to infer suitable asset types the reasoner first has to identify the suitable platform and sensor types based on the above capability ratings. We have created an ontology to represent these ratings and rating types concepts. The figure 6 depicts the NIIRS [12] imagery types and NIIRS imagery rating concepts of this ontology. We have imported this ontology into our ISTAR[13] ontology and associated these concepts with sensors and platforms types. At the reasoner level, we then use Pellet[14] to identify platform types that could provide a particular rating or above (using subsumption relationships among the rat-

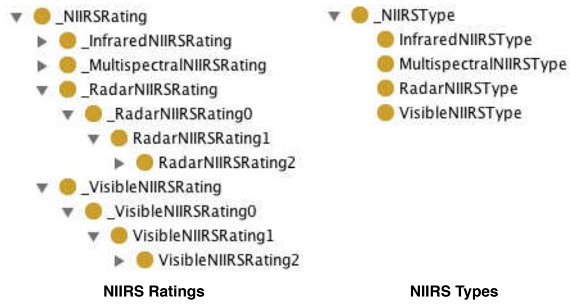

Fig. 6. NIIRS imagery types and ratings

ings) and sensor types that could be used to satisfy the capabilities for a specific rating. The reasoner then uses a set covering algorithm to compute all possible asset types that could be used to satisfy the task requirements. For example, to *identify a jeep* following asset types are recommended.

Asset Type	Explanation
iRobotPackbot **with** AcousticArray	Provides an **acoustic** signature of **value 9**
Raven **with** DaylightTV	Provides a **visual** rating of **value 7**
Reaper **with** DaylightTV	Provides a **visual** rating of **value 6**
Reaper **with** SAR	Provides a **radar** rating of **value 6**
GlobalHawk **with** EOCamera	Provides a **visual** rating of **value 6**
GlobalHawk **with** SAR	Provides a **radar** rating of **value 6**
HarrierGR9 **with** EOCamera	Provides a **visual** rating of **value 7**
NimrodMR2 **with** EOCamera	Provides a **visual** rating of **value 6**

Table 1. Assets capable of identifying a jeep

6 Conclusions and Future Work

In this paper, for the assets-to-tasks assignment problem, we have proposed an approach motivated by the importance of the litheness to the assignment problem. We have combined an ontology-based and rule-based reasoning mechanisms to achieve this. We have proposed a formalism to represent tasks. A well known knowledge corpus is formalised to create a knowledge-base, based on this formalism. A set of rules has been implemented to draw conclusions from this knowledge-base and we have validated the flexibility of this inference process by examples and a case study. In this architecture, the rule-based system is used to infer the information providing capabilities whilst an ontology-based reasoner is used to produce sound asset types that are necessary and sufficient to meet the information requirements of the tasks.

We have demonstrated the usefulness of the proposed approach by means of an example scenario. Our experiments imply that the research is promising even though it is currently in its early stages. Hence, we plan to investigate the following issues as a future work. First, we want to generalize the task representation so that a number of other domains could be represented using the same formalism. Second, the current version of the rule-based reasoning depends on the rule engines such as Prolog and Jess[16]. This is partly due to the existing limitations of the rule languages and tools catered for Semantic Web (e.g., SWRL does not support negation). We are currently investigating other rule representations that enable us to formalise rules using first order logics constructs.

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 document 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 authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

1. I. F. Akyildiz, Y. S. W. Su, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40:102–114, 2002.
2. 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 *Proceedings of the 1st International Technology Alliance Conference*, 2007.
3. 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.
4. M. Botts, A. Robin, J. Davidson, and I. Simonis. OpenGIS© sensor web enablement architecture document. Technical report, Open Geospatial Consortium Inc, 2006.
5. J. Byers and G. Nasser. Utility-based decision-making in wireless sensor networks. In *MobiHoc '00: Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*, pages 143–144, Piscataway, NJ, USA, 2000. IEEE Press.
6. W. F. Clocksin and C. S. Mellish. *Programming in Prolog: Using the ISO Standard*. Springer, 5 edition, 2003.
7. M. Dean and G. Schreiber. OWL web ontology language reference. W3C recommendation, W3C, February 2004.
8. T. M. Doll. Optimal sensor allocation for a discrete event combat simulation. Technical report, Naval Postgraduate School, Monterey, CA 93943-5000, June 2004. Thesis: MSc in Operations Research.

[16] http://www.jessrules.com/

9. M. Gomez, A. Preece, M. P. 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 *EKAW '08: Proceedings of the 16th international conference on Knowledge Engineering*, pages 347–363, Berlin, Heidelberg, 2008. Springer-Verlag.
10. C. Goodwin and D. Russomanno. An ontology-based sensor network prototype environment. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN 2006)*, Nashville TN, USA, 2006.
11. B. Guo, M. S. Nixon, and T. R. Damarla. Acoustic information fusion for ground vehicle classification. In *The 11th International Conference of Information Fusion*, Cologne, Germany, July 2008.
12. J. M. Irvine. *National Imagery Interpretability Rating Scales (NIIRS)*, pages 1442–1456. Marcel Dekker, Oct. 2003.
13. M. P. Johnson, H. Rowaihy, D. Pizzocaroz, A. Bar-Noy, S. Chalmers, T. L. Porta, and A. Preece. Frugal sensor assignment. In *4th IEEE International Conference on Distributed Computing in Sensor Systems*, June 2008.
14. Joint publication, "Joint Publication 2-01: Joint and National Intelligence Support to Military Operations", 2004.
15. M. Paolucci, T. Kawamura, T. R. Payne, and K. P. Sycara. Semantic matching of web services capabilities. In *ISWC '02: Proceedings of the First International Semantic Web Conference on The Semantic Web*, pages 333–347, London, UK, 2002. Springer-Verlag.
16. 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 *Proceedings of SPIE Defense Transformation and Net-Centric Systems 2008*, to appear, mar 2008.
17. D. Roberts, G. Lock, and D. C. Verma. Holistan: A futuristic scenario for international coalition operations. In *Proceedings of Knowledge Systems for Coalition Operations (KSCO 2007)*, 2007.
18. A. Robin, S. Havens, S. Cox, J. Ricker, R. Lake, and H. Niedzwiadek. OpenGIS© sensor model language (SensorML) implementation specification. Technical report, Open Geospatial Consortium Inc, 2006.
19. J. Scholtz, J. Young, J. L. Drury, and H. A.Yanco. Evaluation of human-robot interaction awareness in search and rescue. Technical report, The MITRE Corporation, Bedford, MA, USA, 2004.
20. 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*, pages 655–663, 2003.
21. C. M. Sperberg-McQueen, E. Maler, J. Paoli, F. Yergeau, and T. Bray. Extensible markup language (XML) 1.0 (third edition). first edition of a recommendation, W3C, Feb. 2004. http://www.w3.org/TR/2004/REC-xml-20040204.
22. S. J. Tutton. Optimizing the allocation of sensor assets for the unit of action. Technical report, Naval Postgraduate School, Monterey, CA 93943-5000, June 2003. Thesis: MSc in Operations Research.
23. K. Whitehouse, F. Zhao, and J. Liu. Semantic streams: A framework for composable semantic interpretation of sensor data. In *Wireless Sensor Networks*, pages 5–20. Springer, 2006.