Tasking and Sharing Sensing Assets Using Controlled Natural Language

Alun Preece[a], Diego Pizzocaro[a], Dave Braines[b], David Mott[b]

[a]School of Computer Science and Informatics, Cardiff University, Cardiff, UK
[b]Emerging Technology Services, IBM United Kingdom Ltd, Hursley Park, Winchester, UK

ABSTRACT

We introduce an approach to representing intelligence, surveillance, and reconnaissance (ISR) tasks at a relatively high level in controlled natural language. We demonstrate that this facilitates both human interpretation and machine processing of tasks. More specifically, it allows the automatic assignment of sensing assets to tasks, and the informed sharing of tasks between collaborating users in a coalition environment. To enable automatic matching of sensor types to tasks, we created a machine-processable knowledge representation based on the Military Missions and Means Framework (MMF), and implemented a semantic reasoner to match task types to sensor types. We combined this mechanism with a sensor-task assignment procedure based on a well-known distributed protocol for resource allocation. In this paper, we re-formulate the MMF ontology in Controlled English (CE), a type of controlled natural language designed to be readable by a native English speaker whilst representing information in a structured, unambiguous form to facilitate machine processing. We show how CE can be used to describe both ISR tasks (for example, detection, localization, or identification of particular kinds of object) and sensing assets (for example, acoustic, visual, or seismic sensors, mounted on motes or unmanned vehicles). We show how these representations enable an automatic sensor-task assignment process. Where a group of users are cooperating in a coalition, we show how CE task summaries give users in the field a high-level picture of ISR coverage of an area of interest. This allows them to make efficient use of sensing resources by sharing tasks.

Keywords: sensor assignment; sensor sharing; intelligence, surveillance, reconnaissance; ontologies; controlled natural language

1. INTRODUCTION

In the context of intelligence, surveillance, and reconnaissance (ISR) operations, there are typically multiple ways to achieve a task using sensor-provided data. For example, the NIIRS (National Image Interpretability Rating Scales) framework characterises various kinds of ISR tasks that can be achieved using visual sensing data of different types (visible, radar, infrared and multispectral).[1] Other kinds of sensor-provided data can be similarly characterised, for example acoustic and seismic data. Therefore, given an ISR task and a set of sensing assets in a particular area of interest, there may be many options for resourcing that task. In a coalition context, the problem is more complex, because the assets may be 'owned' by different partners. From the point of view of a user (for example, an ISR analyst) with a particular information need (for example, tracking high value targets in an area), the problem of identifying suitable ISR assets is difficult, without a great deal of knowledge about sensing capabilities and availability of coalition assets.

As part of a solution to this problem, several works have proposed the use of some form of knowledge base or mapping that relates sensor capabilities to task requirements (for example,[2–5]), to aid either automatic or semi-automatic identification of suitable assets for tasks. In our previous work,[6,7] we developed an approach founded on the Military Missions and Means Framework (MMF).[8] We created ontologies of task and asset types, and an automatic procedure for matching one to the other, through the capabilities required on one and provided by the other. This approach was intended to be extensible, and we later showed that additional matching knowledge from the NIIRS framework could be easily incorporated.[9]

Send correspondence to A.Preece, E-mail: A.D.Preece@cs.cf.ac.uk, Tel: +44 (0) 29 2087 4653

Our approach was intended mainly for automated assignment of assets to tasks, in a context where ISR assets are relatively scarce (demand exceeds supply), and many tasks may be competing for the same resource. While assets may be shared among tasks, we assumed that resourced tasks would not be shared among users. Because there was competition between tasks, we studied models for the pre-emption of existing tasks by new tasks.[10] A typical case would involve a new higher-priority task effectively "stealing" resources from an existing lower-priority task.

Going forward, we now aim to relax some of these assumptions to support a more interactive approach, where the human user works in a more cooperative manner with the system. Specifically, we want the user to be able to explore the various means of achieving a task, and consider the option of sharing existing (identical or similar) tasks rather than creating a new request for resources (thus avoiding common cases of pre-emption). The main idea is to show the user — subject to access policies — currently-resourced tasks in their area of interest, and to make is easier for them to "join" an existing task (again, subject to access policies) than to create a new one. For this to work, the representation of tasks and means of achieving them (combinations of assets) would need to be much more transparent to users than in our earlier work, and other work in this area.

We therefore decided to experiment with the use of a *controlled natural language* (CNL) to express the elements of our knowledge base, and allow us to generate human-understandable representations of ISR tasks and their resourcing. A CNL is a subset of a natural language, commonly English, with restricted syntax and vocabulary. Often they are used to provide an information representation that is easily machine processable (with low complexity and no ambiguity) while also being human-readable (see, for example,[11]). Our main goals were:

- to verify whether our MMF-based knowledge base could be expressed in CNL, with no loss of power to support automated asset-task matching; and
- to explore how a CNL-based representation of tasks and their resourcing could be used to create a human-understandable tool to promote task sharing among users.

The remainder of this paper is organised as follows: Section 2 reviews our MMF-based approach to task-asset matching, including the task and asset ontologies and the NIIRS-based matching procedure; Section 3 describes how we achieved the first goal above, expressing our knowledge base and the various facts associated with task-asset assignment in a CNL; Section 4 presents our prototype tool — in the form of a mobile tablet-based app — that allows users to explore the space of assigned tasks, and choose to share existing tasks; finally, Section 5 concludes the paper.

2. KNOWLEDGE-BASED MATCHING OF ISR ASSETS TO TASKS

Our original knowledge base and task-asset matching procedure[6,7] was based on an ontology derived from the Military Missions and Means Framework (MMF).[8] To derive our ontology, we formalized concepts and relationships from the MMF documentation, the main ones being as shown in Figure 1. Missions are comprised of operations which are in turn comprised of tasks. Tasks require capabilities, which are provided by assets. Assets include platforms and systems; systems — including sensors — are mounted on platforms. The original ontology included a relationship "allocated to" to capture that an asset was assigned to resource a particular task. The ontology was implemented in OWL DL version 1[*] and the task-asset matching procedure was implemented using a combination of Pellet[†] and Java. Pellet was used to perform classification on the ontology, to infer all capabilities provided by all assets (sensor and platforms). The remainder of the matching procedure was implemented in Java, based on a set-covering algorithm.[6]

[*]http://www.w3.org/TR/owl-guide/

[†]An OWL description logic reasoner, http://clarkparsia.com/pellet/

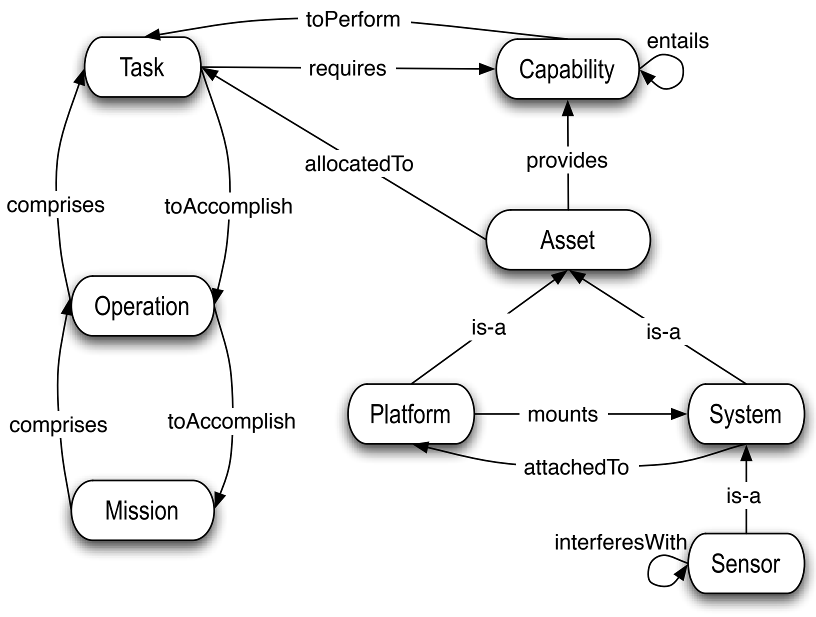


Figure 1. Missions and Means Framework ontology[6,7]

2.1 NIIRS-based Task-Asset Matching Knowledge

A revised matching procedure based on the NIIRS framework was introduced in.[9] Here we describe a revised implementation of the procedure in Prolog, for use with the CNL knowledge base introduced in the following section. The knowledge base contains a set of *intelligence clause* tuples of the form: $\langle IC, DS, FS, C, IT, NR \rangle$, where:

- IC is an *intelligence capability* which include the three capabilities in NIIRS — *detect*, *distinguish*, *identify* — and also *localize*;
- DS is a set of *detectable things* drawn from the NIIRS framework (for example, kinds of vehicle or building);
- FS is a set of more specific features of the detectable entities (for example, the roads or guard posts of a base, or the runways of an airport) — these are not considered further in this paper;
- C is a context, defining the preconditions that must hold for the intelligence clause to apply (for example, detection of a ship in the context of open water) — again, these are not considered further in this paper;
- IT is a type of *sensor-provided data* which includes the NIIRS types — *visible*, *radar*, *infrared*, *multispectral* — plus other types including *acoustic* and *seismic*; and
- NR is a NIIRS rating on a scale of 0 to 9 (for example, *visible-6* or *radar-4*).

These intelligence clause tuples are derived from the NIIRS framework, augmented with others drawn from the sensing literature (for example, on acoustic sensing).

2.2 Matching Procedure

We define a *task type* as a pair $\langle IC, DS \rangle$ where IC and DS are defined as above. Some example task types are *detect* $\{$*tank*$\}$ and *distinguish* $\{$*tank, jeep*$\}$. Note that task types feature either a single detectable (for *detect*, *identify* and *localize* tasks) or a pair of detectables (for *distinguish* tasks).

We say that a given task type $TT_i = \langle IC_i, DS_i \rangle$ *requires* a set of capabilities $CT_i = \{IT_j, NR_j\}$ if there is an intelligence clause fact $IC_j = \langle IC_j, DS_j, FS_j, C_j, IT_j, NR_j \rangle$ where $IC_i = IC_j$ and $DS_i \subseteq DS_j$.[‡]

We define a *bundle type* as a pair $\langle PT, SS \rangle$ where PT is a platform type and SS is a set of sensor types (according to the ontology in Figure 1) and the ontology contains the statement PT *mounts* ST_j for each $ST_j \in SS$ — that is, each sensor type in the set can be mounted on that platform type.[§]

We say that a given bundle type $BT_k = \langle PT_k, SS_k \rangle$ *provides* a set of capabilities CB_k if, for each $C_l \in CB_k$, our ontology includes either the statement PT_k *provides* C_l or the statement ST_j *provides* C_l for some $ST_j \in SS_k$. In other words, each capability in CB_k is provided either by the platform type or a sensor type in the corresponding bundle type BT_k.

We say that a bundle type BT_k *matches* a task type TT_i if $CB_k \supseteq CT_j$ according to the above definitions — that is, if the set of capabilities provided by the bundle type contains the set of capabilities required by the task type. We say that a bundle type BT_k *minimally matches* a task type TT_i when no sensor type can be removed from the bundle type such that the *matches* relationship still holds.

2.3 KB Table Generation and Asset Assignment

The types of task and asset are relatively static; for this reason is it reasonable to pre-generate a *lookup table* of all possible task type / bundle type minimal matches. Doing so allows us to deploy the knowledge base on a mobile device as described in,[10] where we compute the size of a realistic complete table to be in the order of 20,000 entries, which can comfortably be stored in 12MB on a smartphone or tablet. Note that the size of the deployed table can easily be reduced by including only task types and asset types relevant to a particular mission context. It would not make sense, for example, to include maritime ISR tasks and assets in a land-based context. We refer to the individual entries in the lookup table as *assignment templates.* These are of the form: $\langle IC, DS, BT, UF \rangle$, where:

- IC is an intelligence capability, as above;
- DS is a set of detectable things, as above;
- BT is a bundle type, as above, such that BT minimally matches the task type formed by $\langle IC, DS \rangle$;
- UF is a *utility function* compatible with BT and the task type formed by $\langle IC, DS \rangle$.

The utility function associated with an entry provides a means of assessing how effective a particular instance of the bundle type is likely to be in achieving a particular instance of the task type. Usage of the lookup table in asset assignment is covered in detail in.[10] In outline, the assignment procedure is as follows:

1. A user creates a task T_i, from which the system derives the corresponding task type $TT_i = \langle IC_i, DS_i \rangle$.
2. The system retrieves all entries $\langle IC_j, DS_j, BT_j, UF_j \rangle$ where $IC_i = IC_j$ and $DS_i \subseteq DS_j$.
3. The system determines all possible *bundle instances* that conform to all retrieved bundle types BT_j and uses the corresponding utility functions UF_j to derive a utility for each.
4. The preceding step is performed as part of a distributed allocation protocol based on[12] that attempts to maximise overall utility in the face of multiple competing tasks.[¶]

Having reviewed our task-asset matching and assignment procedures, we will now look at the knowledge base in greater detail, by describing its reformulation in controlled natural language.

[‡]In some cases, a hierarchy of detectables allows some inference here; for example, any clause involving *detect* and a kind of detectable D is considered to cover all more specialised kinds of D also — so *detect* { *car* } covers specialised kinds of car (*jeep, SUV, saloon*, etc).

[§]The ontology contains the additional relationship *interferes with* to cover cases where types of sensor are incompatible. No valid bundle type can contain pairs of sensor types involved in an *interferes with* relationship.

[¶]Which, as explained in Section 1, may involve pre-empting one or more existing tasks to accommodate the new task.

3. RE-FORMULATING THE KNOWLEDGE BASE IN CONTROLLED ENGLISH

Several controlled natural languages exist; for our work, we selected a form of Controlled English known as ITA Controlled English, which arose from our research in the International Technology Alliance (ITA).[13] ITA CE was chosen to aid knowledge reuse and the potential for system integration, because it is already in use in a number of related research projects. Hereafter we refer to ITA CE simply as CE. The purpose of CE is that it provides a human-friendly information representation format that is directly processable by machine agents with a clear and unambiguous underlying semantics. In terms of machine-processability, we aimed to demonstrate that CE can play the same role as OWL DL in formalising our ontology definitions[‖], while being more easily understandable by humans (in support of our second goal in Section 1).

Here we introduce CE in the context of reformulating the ontology definitions introduced in the previous section. All of the CE examples used in this paper are directly processable by machine agents and, we hope to illustrate, more consumable by human readers than non-CNL equivalent technical representations. The improvement of CE syntax to allow further linguistic variety and expressivity without undermining the unambiguous semantic grounding is a topic of current research.

3.1 Representing the Core MMF Ontology in CE

CE is used to define both models and instances. Model definitions take the form of concept definitions. CE `conceptualise` sentences are intended to define by concepts by example; that is, they provide generalised examples of how to say things about concepts. Relationships between concepts are also considered to be concepts, though we will refer to them in this paper simply as relationships. A CE model may also include the definition of logical inference rules (not shown in this paper) which are used to express further information about the concepts and relationships and how they are logically related. Concepts may be specialisations of other concepts (indicated by `is a` declarations). The following sample definitions cover part of the original MMF ontology (Figure 1):

```
conceptualise a ~ capability ~ C.

conceptualise the mission M
  ~ comprises ~ the operation O.

conceptualise the operation O
  ~ comprises ~ the task T.

conceptualise the task T
  ~ requires ~ the capability C.

conceptualise the asset type A
  ~ is rated as ~ the NIIRS rating R and
  ~ provides ~ the capability C.

conceptualise a ~ system type ~ S that
  is an asset type.

conceptualise a ~ sensor type ~ S that
  is a system type.

conceptualise a ~ platform type ~ P that
  is an asset type.

conceptualise the platform type P
  ~ mounts ~ the system type S.
```

[‖]Which in turn is similar to other efforts including SensorML,[14] OntoSensor[5] and the W3C Semantic Sensor Network Incubator Group.[2]

Here are some sample platform and sensor type definitions that extend the higher-level definitions; these are intended to be illustrative and are not complete. CE sentences beginning with '`Note:`' are annotations referring to terms in the preceding sentence.

```
conceptualise a ~ UAV ~ U that is a platform type.

conceptualise a ~ MALE UAV ~ M that is an UAV.
Note: MALE = Medium Altitude, Long Endurance.

conceptualise a ~ Predator A ~ P that is a MALE UAV.

conceptualise an ~ EO camera ~ E that is a sensor type.
Note: EO = Electro-optical.
```

Once the conceptual model is defined subsequent assertions can be made according to these concepts and relationships. The CE syntax `there is a...` is used to create instances. As an example, here is a "prototypical" instance `Predator A platform type` which defines all of the "prototypical" instances of sensor types that can be mounted on that kind of platform (specified using the `mounts` relationship):

```
there is a Predator A named 'Predator A platform type' that
  mounts the sensor type 'EO camera sensor type' and
  mounts the sensor type 'TV camera sensor type' and
  mounts the sensor type 'FLIR camera sensor type' and
  mounts the sensor type 'LADAR sensor type'.
```

3.2 Representing Task-Asset Matching Knowledge in CE

To support the NIIRS-based matching of tasks to assets, we extend the above definition of task as follows, to include a NIIRS-style intelligence capability and one or more kinds of detectable thing. We also include a spatial area-of-interest, a time period, and a priority to allow tasks to be ranked if assets are scarce:

```
conceptualise the task T
  ~ requires ~ the intelligence capability IC and
  ~ is looking for ~ the detectable thing DT and
  ~ operates in ~ the spatial area SA and
  ~ operates during ~ the time period TP and
  ~ is ranked with ~ the task priority PR.
```

Here are some sample intelligence capabilities and detectable things:

```
there is an intelligence capability named detect.
there is an intelligence capability named identify.
there is an intelligence capability named localize.

there is a detectable thing named 'wheeled vehicle'.
there is a detectable thing named 'tracked vehicle'.
there is a detectable thing named 'field artillery'.
```

Here is a sample task instance:

```
there is a task named t1265 that
  requires the intelligence capability detect and
  is looking for the detectable thing 'wheeled vehicle' and
  operates in the spatial area r942 and
  operates during the time period t1789 and
  is ranked with the task priority medium.
```

The time period `t1789` and spatial region `r942` are assumed to be defined by separate instances containing the relevant spatio-temporal values, but these are not shown here.

As described in Section 2.1, the NIIRS-based approach allows automatic matching of tasks to asset capabilities, by means of encoding NIIRS knowledge as a set of *intelligence clause* facts. In CE these facts are modelled as follows:[**]

```
conceptualise the intelligence clause IC
  ~ fulfills ~ the intelligence capability IC and
  ~ is looking for ~ the detectable thing DT1 and
  ~ provides ~ the capability C and
  ~ is rated as ~ the NIIRS rating R.
```

Here is an example *intelligence clause*, for the case that wheeled vehicles can be identified with visible imagery at NIIRS rating 4 or better:

```
there is an intelligence clause named ic003 that
  fulfills the intelligence capability identify and
  is looking for the detectable thing 'wheeled vehicle' and
  provides the capability 'visible sensing' and
  is rated as 'visible NIIRS rating 4'.
```

NIIRS ratings are associated with platform and sensor types, by means of the `provides` relationship (as with the `mounts` example above, instances of the `provides` relationship are defined between “prototypical” instances of the relevant sensor and platform types):

```
there is an EO camera named 'EO camera sensor type' that
  provides the capability 'visible sensing'.

there is a Predator A named 'Predator A platform type' that
  is rated as the NIIRS rating 'visible NIIRS rating 6' and
  is rated as the NIIRS rating 'RADAR NIIRS rating 4'.
```

3.3 Assignment Representation in CE

Entities associated with the KB lookup table described in Section 2.3 are modelled as follows:

```
conceptualise the assignment template AT
  ~ fulfills ~ the intelligence capability IC and
  ~ is looking for ~ the detectable thing DT and
  ~ can be satisfied by ~ the bundle type BT and
  ~ is ranked by ~ the utility function UF.

conceptualise the bundle type BT
  ~ is deployed on ~ the platform type P and
  ~ uses ~ the sensor type S.
```

Here is an example assignment template (and associated instances) in CE, based on the earlier examples:

[**]Here we show only the 4 elements of the tuple $\langle IC, DS, FS, C, IT, NR\rangle$ from Section 2.1; FS and C are outside the scope of this paper and omitted.

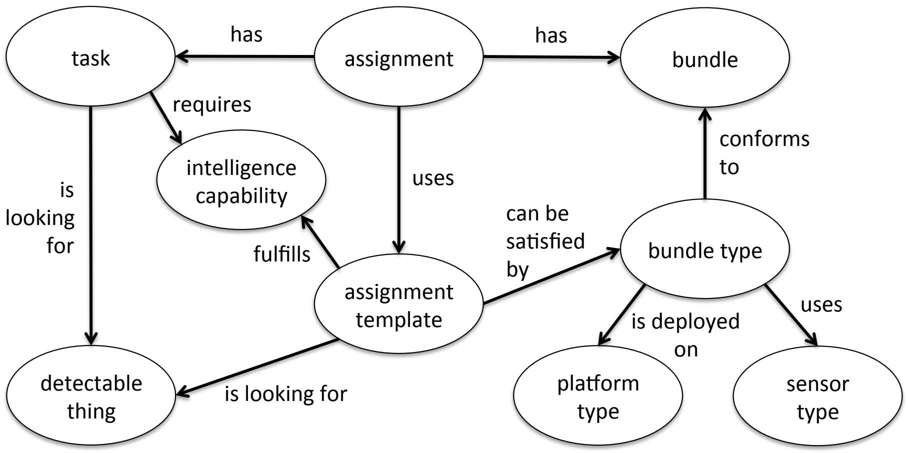


Figure 2. Extract of model showing main relationships between `assignment` and linked concepts

```
there is an assignment template named at349 that
  fulfills the intelligence capability identify and
  is looking for the detectable thing 'wheeled vehicle' and
  can be satisfied by the bundle type bt312 and
  is ranked by the utility function CDP.

there is a bundle type named bt312 that
  is deployed on the platform type 'Predator A platform type' and
  uses the sensor type 'EO camera sensor type'.

there is a utility function named CDP.
Note: CDP = Cumulative Detection Probabilty.
```

In our earlier ontology, a property *allocatedTo* was used to link tasks to assets. This proved inadequate to handle the richer relationship between asset bundles and tasks. For this purpose, we define the concept of an `assignment` as follows:

```
conceptualise an ~ assignment ~ A that
  has the task T as ~ task ~ and
  has the bundle B as ~ bundle ~ and
  has the value US as ~ utility score ~.

conceptualise the assignment A
  ~ uses ~ the assignment template AT and
  ~ is provided by ~ the coalition partner CP and
  ~ is owned by ~ the user UO and
  ~ is joined by ~ the user UJ.
```

The main relationships between `assignment` and other concepts are shown graphically in Figure 2. Thus, an assignment is a concept that relates a task to a bundle, and has an associated utility score. An assignment uses an assignment template (which defines the utility function used to generate the utility score). These definitions use three additional concepts: a sensor bundle (that conforms to a particular bundle type[††]), a user (individual person, for example an ISR analyst), and a coalition partner (which has ownership of particular assets). Incomplete definitions for these are shown below:

[††]A bundle also contains specific asset instances, not shown here.

```
conceptualise the bundle B
  ~ conforms to ~ the bundle type BT.

conceptualise a ~ user ~ U.

conceptualise a ~ coalition partner ~ CP.
```

Assignments are generated by the procedure outlined in Section 2.3. An example assignment is shown below:

```
there is an assignment named a43288 that
  has the task t1265 as task and
  has the bundle b17352 as bundle and
  has '0.7' as utility score and
  uses the assignment template at349.

there is a bundle named b17352 that
  conforms to the bundle type bt312.
```

We also associate "provenance" information with an assignment, in terms of the coalition partner (typically a country) that provides the assets in the bundle, and the user who originated the task (the task "owner"). As described in the next section, we allow assignments to be shared among users — for this purpose, we define the relationship "is joined by". Some example provenance and sharing information on an assignment is shown below:

```
the assignment a43288
  is provided by the country UK and
  is owned by the user Sue41 and
  is joined by the user Bill356 and
  is joined by the user Tommy9 and
  is joined by the user Zack99.
```

Figure 2 can be seen as an extension of our original ontology depicted in Figure 1, emphasising concepts of importance in the task-asset assignment process. Originally, there was no need to explicitly model the new concepts in Figure 2 because most of them were internal to the assignment procedure. Having them made explicit allows us to make the assignment process more transparent, as per our second goal stated at the end of Section 1. In view of this, the next section shows how the CE knowledge base and instance representations can be used in a tool to facilitate task-asset assignment and sharing.

4. TASK-ASSET ASSIGNMENT AND SHARING USING A TABLET-BASED APP

Having established the feasibility of using CE to express our task-asset assignment knowledge base and instances, we now consider how the user in the field may exploit the capabilities afforded by our knowledge-based system. We created a client interface, implemented as an app on a mobile device, as part of an implemented illustration-of-concept sensor-task assignment system called SAM (Sensor Assignment to Missions). The original version of this app[10] was implemented on a smartphone (Apple iPhone) and emphasised simplicity and minimality in terms of features:

1. Allow a user to create an ISR task in an area-of-interest, by means of a convenient user interface, and submit the task for asset assignment.
2. Achieve a separation between *what* information the user requires and *how* the information is obtained (by which kind of sensing).

We have now created an enhanced version of the app with extended functionality, offering users more information and choice:

3. Allow a user to view all tasks with assigned assets in an area of interest (subject to access policies).
4. Allow the sharing of tasks among users (again, subject to access policies).

The primary motivation for feature (3) was to give a decision-maker an overview of how well-covered an area in in ISR terms, not just in terms of what tasks are currently being resourced, but also the likely permanence of these tasks (by allowing the user to view details of the takes such as their ownership and priority, which affect the likelihood of a task being pre-empted). The motivation for feature (4) was to reduce task pre-emption, by making it easier for a user to share an existing task than to create a new request for resources (because tasks are only pre-empted when there is a competing task in the same area-of-interest). These features benefit from a redesign of the app to run on a tablet (the Apple iPad).

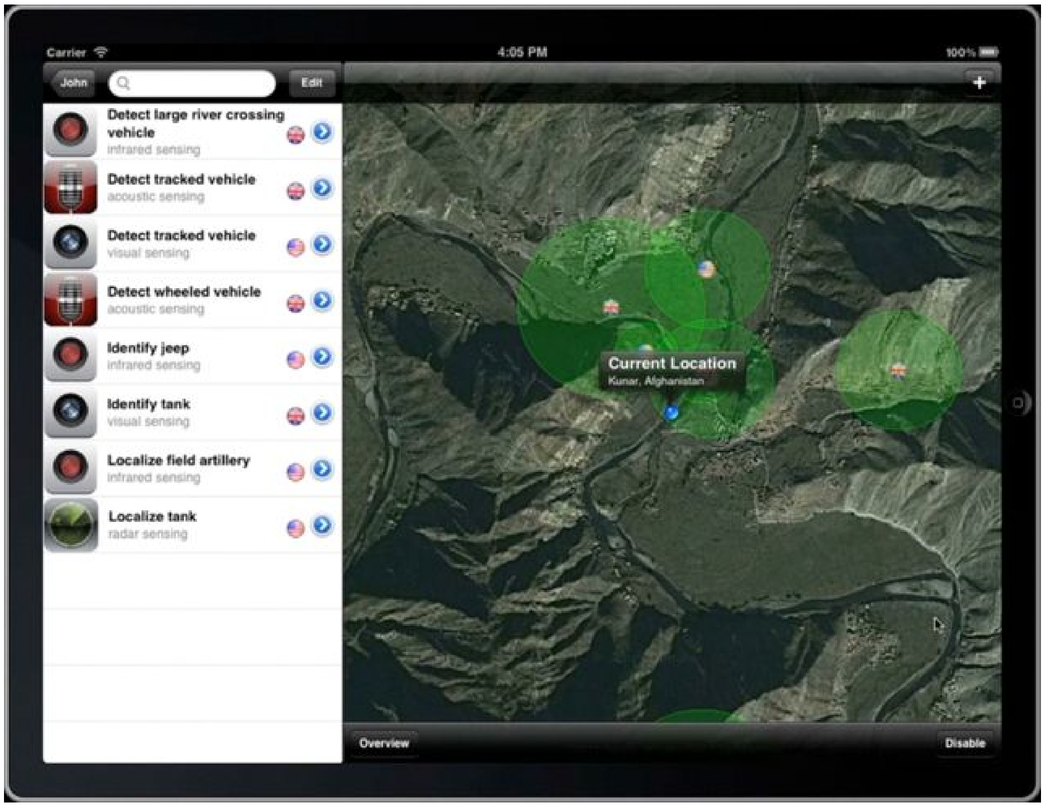


Figure 3. Sensor Assignment to Missions (SAM) iPad app

Figure 3 shows the main panel for the iPad app: the *task panel*, illustrating the implementation of feature (3). The task list is on the left, and the locations of tasks are shown as circular regions on the map to the right. The list of tasks is contextualised by the area of interest, which is by default determined by the iPads location (via GPS) but can be set manually also. This display is from the viewpoint of a logged-in user: each user belongs to a single coalition parter, and sees a list of tasks visible according to their access policies.[‡‡]

The task list is searchable using the box on the top-left allowing users to filter the displayed list by intelligence capability or type of detectable thing. Tasks are summarised view with an icon that provides a quick indication of the kind of sensing capability assigned (for example, by means of acoustic, visual, infrared, seismic, or radar sensing). Information for this list can be obtained by querying a CE Store for `assignment` instances. The elements of the CE model used to generate the task list are shown in Table 1. As noted above, the `capability` is displayed

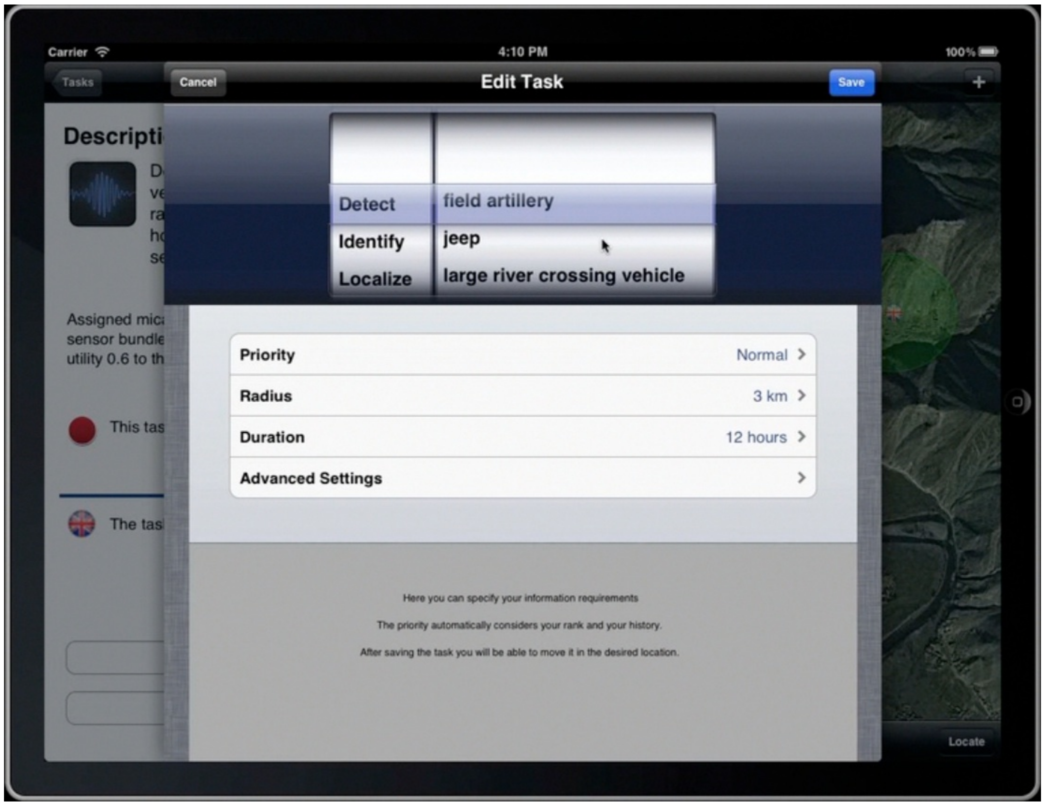
[‡‡]Access policies are rule-based, and can take account of factors including the user's coalition partner membership, their rank, membership of a particular group within the coalition, and also the partner ownership, rank, and group associated with the task. Full discussion of these policies is outside the scope of this paper.

both in text and as an icon. The `coalition partner` is displayed as a “flag” icon to the right of each item, and also at the centre of the task on the map. (These could be more fine-grained than country-level the flags are merely for illustrative purposes.) These elements are returned in JSON by a query to the CE Store from the client app.

Model element	Source
`intelligence capability`	from `task` associated with `assignment`
`detectable thing`	from `task` as above
sensing `capability`	from `sensor type` associated with `assignment` via `bundle type` and `assignment template`
`coalition partner`	from `assigment` directly

Table 1. Elements of the CE model used in the task list (Figure 3)

A user may create an ISR task using the form shown in Figure 4, corresponding to the task model described in Section 3.2. The app allows a user to specify the area-of-interest of the task in terms of a point on the map, and radius.

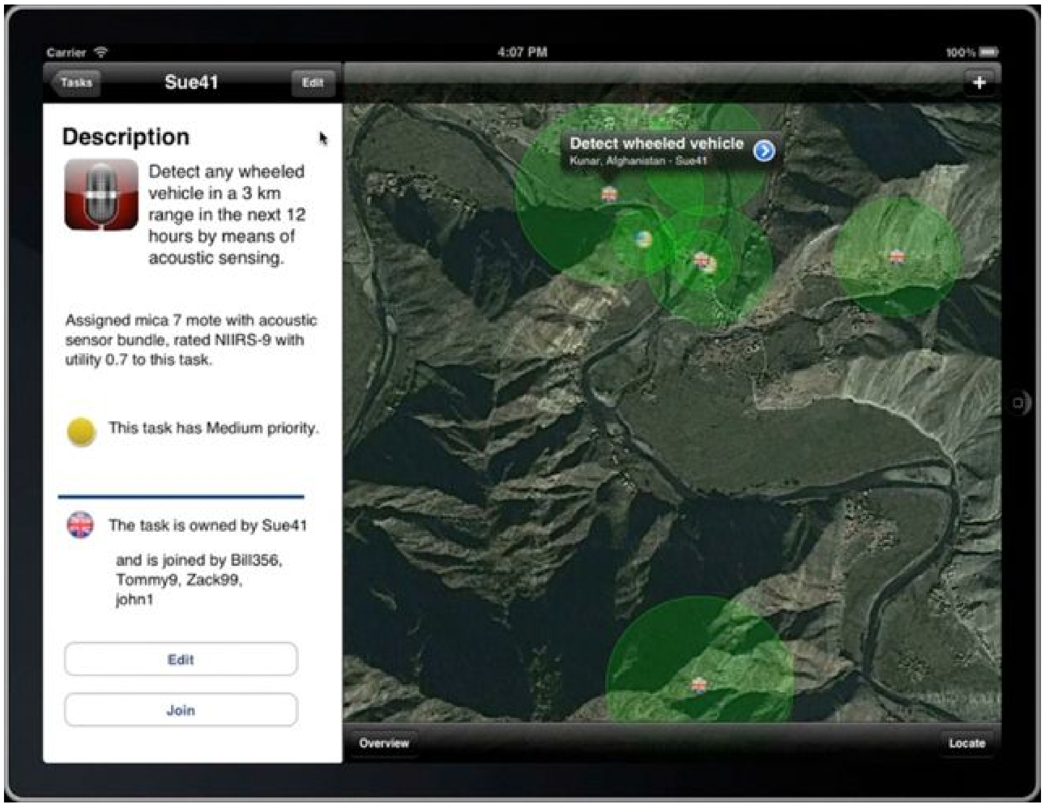


Priority	Normal
Radius	3 km
Duration	12 hours
Advanced Settings	

Here you can specify your information requirements
The priority automatically considers your rank and your history.
After saving the task you will be able to move it in the desired location.

Figure 4. iPad app task input form

The task panel on the left of Figure 3 shows only a summary of tasks. More detail on each can be obtained by selecting an individual task, which results in an *assignment view* like the one shown in Figure 5. Assuming the task has assigned sensors, the display presents a text summary derived from the aforementioned CE model elements plus the additional ones listed in Table 2. These additional items can be obtained with the task list data in JSON by a query to the CE Store — the amount of data is relatively small so it is worth obtaining it all by a single request. With this amount of detail, the user can understand how the task is currently being resourced. Inclusion of the NIIRS rating is intended to give an indication of the quality of the data the resourcing bundle can collect. We are considering other ways to convey quality information, as it may not be reasonable to assume users are familiar with the NIIRS scale. As noted above, the task priority is an indication of how likely the task is to be pre-empted (lower priority tasks are more prone to this). Information on the task owner and other users who have joined the task is intended to have a “social” effect, as the logged-in user is likely to know other users in the region.



Detect any wheeled vehicle in a 3 km range in the next 12 hours by means of acoustic sensing.

Assigned mica 7 mote with acoustic sensor bundle, rated NIIRS-9 with utility 0.7 to this task.

This task has Medium priority.

The task is owned by Sue41

and is joined by Bill356, Tommy9, Zack99, john1

Figure 5. Details of an assigned task

Model element	Source
`range`	from `task` associated with `assignment` — not shown in Section 3.2
`duration`	from `task` as above
`platform type`	from `bundle type` associated with `assignment` via `assignment template`
`sensor type`	from `bundle type`
`NIIRS rating`	from `platform type` as above
`utility score`	from `assignment` directly
`priority`	from `task`
the `is owned by user`	from `assignment`
the set of `is joined by user` instances	from `assignment` as above

Table 2. Additional elements of the CE model used in the assignment view (Figure 5)

Based on this detailed task assignment information, the user may use the buttons on the bottom left to choose to join or edit the existing task. The intent here is that it is a more efficient use of network resources for a user to join an existing task than to create a new task which will compete for resources with existing tasks. Currently, editing an existing task effectively creates a new task (though is a quicker procedure for the user) though we are considering ways to avoid this in particular cases where the edited task can still be resourced by the currently-assigned bundle.

In summary, as a result of the latest implementation work we have demonstrated that all information provided by the current user interface can be generated through processing of the underlying CE sentences as the direct underlying information representation format, without the need for conversion to a secondary form such as OWL. This is achieved through the use of the CE Store processing environment which provides a set of APIs to define and query information in the form of CE sentences.

The iPad app was demonstrated to subject-matter experts from the US, UK, and NATO communities in September and October 2011. Highly positive feedback was obtained, especially concerning the improved transparency of the iPad version compared to the earlier iPhone functionality. This confirmed to us that features (3) and (4) above are considered important for a realistic deployment of our approach. Work is ongoing to further refine and improve the app.

5. CONCLUSION

In this paper, we set out to show that a controlled natural language could be used to express a knowledge base to support automated asset-task matching in an ISR context. We translated and extended our original model (based on the Military Missions and Means Framework) into ITA Controlled English with no loss of power to support our automated asset-task matching procedure. We showed how the new model can be used in the context of a user-facing mobile app to assist understanding of the currently-resourced set of ISR tasks, and informed choices on whether to create a new task or share an exiting task.

CE provides a good basis for translation between human languages and may offer advantages in a coalition context. Our tablet app demonstrates how a multi-modal interface can be driven by underlying CE-based information, with the information being presented to the user in a manner that is conducive to their task, for example labelled points on a map rather than CE sentences with lat/long values. Because the system is based on CE, if a user wishes to go outside the designed scope of the application they can contribute CE sentences that conform to the underlying model in order to interact with the system if required.

We believe that the current form of ITA CE provides a good start and does appeal to non-technical users, but we wish to invest further into the refinement of the syntax and an increased expressivity to allow information to be expressed more eloquently without sacrificing the underlying machine-processability. Since the underlying CE model is extensible the opportunity for "local knowledge" to be added via the app is very real (both in terms of new facts and in terms of model refinements/extensions).

Acknowledgements

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

REFERENCES

1. Irvine, J. M., "National Imagery Interpretability Rating Scales (NIIRS)," in [*Encyclopedia of Optical Engineering*], 1442–1456, Marcel Dekker (Oct. 2003).
2. Lefort, L., Henson, C., and Taylor, K., "Semantic Sensor Network XG final report," (2011). http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/.
3. Mullen, T., Avasarala, V., and Hall, D. L., "Customer-driven sensor management," *IEEE Intelligent Systems* **21**, 41–49 (Mar/April 2006).
4. Simonis, I. and Echterhoff, J., "OGC Sensor Planning Service Implementation Standard," tech. rep., Open Geospatial Consortium (2011). http://www.opengis.net/doc/IS/SPS/2.0.
5. Russomanno, D., Kothari, C., and Thomas, O., "Building a sensor ontology: A practical approach leveraging ISO and OGC models," in [*Proceedings of the International Conference on Artificial Intelligence*], 637–643 (2005).

6. Gomez, M., Preece, A., Johnson, M., de Mel, G., Vasconcelos, W., Gibson, C., Bar-Noy, A., Borowiecki, K., Porta, T. L., Pizzocaro, D., Rowaihy, H., Pearson, G., and Pham, T., "An ontology-centric approach to sensor-mission assignment," in [*Proc 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2008)*], 347–363, Springer (2008).
7. Preece, A., Gomez, M., de Mel, G., Vasconcelos, W., Sleeman, D., Colley, S., Pearson, G., Pham, T., and Porta, T. L., "Matching sensors to missions using a knowledge-based approach," in [*SPIE Defense Transformation and Net-Centric Systems 2008 (SPIE Defense and Security Symposium, SPIE Proceedings Vol 6981)*], SPIE (2008).
8. Sheehan, J. H., Deitz, P. H., Bray, B. E., Harris, B. A., and Wong, A. B. H., "The military missions and means framework," in [*Proceedings of the Interservice/Industry Training and Simulation and Education Conference*], 655–663 (2003).
9. de Mel, G., Sensoy, M., Vasconcelos, W., and Preece, A., "Flexible resource assignment in sensor networks: A hybrid reasoning approach," in [*1st International Workshop on the Semantic Sensor Web (SemSensWeb 2009)*], (2009).
10. Pizzocaro, D., Preece, A., Chen, F., Porta, T. L., and Bar-Noy, A., "A distributed architecture for heterogeneous multi sensor-task allocation," in [*Proc 7th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS'11)*], (2011).
11. Sowa, J., "Common logic controlled english," (2004). http://www.jfsowa.com/clce/clce07.htm.
12. Shehory, O. and Kraus, S., "Methods for task allocation via agent coalition formation," *Artificial Intelligence* **101**, 165–200 (1998).
13. Mott, D., "Summary of ITA controlled english," (2010). https://www.usukita.org/papers/5658/details.html.
14. Botts, M. and Robin, A., "OpenGIS sensor model language (SensorML) implementation specification," tech. rep., Open Geospatial Consortium Inc (2007).