OWL-Based Ontology Construction Method for Battlefield Information Service

Xiaozhen Tao, Wendong Zhao, Yi Wei, Chang Tian
Institute of Communication Engineering, PLA University of Science and Technology
Nanjing, China
18699193875@163.com

***Abstract*—With the constant development of IT, on-demand battlefield information sharing becomes the urgent need for military to achieve information superiority and win network centric warfare. In this paper, a method is proposed for building ontology which can support Battlefield Information Service (BIS). Based on the requirement analysis of BIS, Battlefield Information Ontology (BIO) is constructed in OWL, including Top-Frame Ontology (TFO) and Domain-Branch Ontology (DBO). The value assignment properties are added to solve the assignment problem of concepts with different data types. And a new problem, the integrated reasoning of preconditions and outputs, is raised in this paper. A concept may be equal to another one under some precondition, which cannot be discovered in classical way of matching preconditions and the outputs separately. To express the above equivalent relation in ontology, a transformation model is proposed to make OWL have the ability to describe high-order relations, which only allows first-order relations formerly. An application example is given to show how the ontology supports semantic-based BIS publishing, query and match.**

Keywords-Battlefield Information Service; ontology; assignment; integrated reasoning

I. INTRODUCTION

With the constant development of IT in military field, network center warfare as a new mode of combat, has attracted wide attention of military experts. The U. S. army proposed the plan of Global Information Grid (GIG) in September 1999 [1]. GIG is thought to be the key to network center warfare, and is essential for achieving information and decision superiority. With the adoption of open grid service architecture (OGSA), spatial metadata, interoperation and other techniques, provision of battlefield information can be regard as a kind of network service [2]..

[3] says, Battlefield Information Service (BIS) should have two aspects of abilities: One is that any combat platform at anytime anywhere can obtain all the information it needs; the other one is that any combat platform at anytime anywhere can provide all the information it has. But in our opinion, a combat platform as information demander should be able to not only obtain all the information, but also minimize the acquisition of useless information; a combat platform as information supplier should be able to not only provide all the information it has, but also provide the information without misunderstanding. So more than dynamic BIS access, we also require an exact match of service and request, and the barrier-free communication between supplier and demander.

Ontology-based semantic technique provides an effective way to improve the accuracy of service match. We can build Battlefield Information Ontology (BIO) in a standard description language and take it as the common communication way between information supplier and demander. BIO can be used for service description, which is needed by service publishing and query. And it can provide semantic reasoning for service match.

In this paper we focus on the construction of BIO. In Section 2 the requirement analysis of BIS is done as a guide for ontology design. Section 3 introduces the method of BIO design and construction in detail. And an application example of BIS match is given in Section 4 to show how BIO works. Finally, Section 5 discusses conclusions as well as some issues that require further research.

II. RELATED WORKS

Ontology technology is an effective way for computer logical reasoning, which is the foundation of semantic service discovery. Essentially, ontology is a knowledge base of some domain. But it should be strictly organized and clearly described, as well as cover all aspects of the domain. In 1995, Gruber [4] offered 5 principles for ontology construction: clarity, coherence, extendibility, minimal encoding bias and minimal ontological commitment. And Perez [5] supplemented these principles in 1999. Based on these principles, researchers have found out kinds of ontology construction methods. But ontologies of different domains have different characteristics, and different applications have different requirement. We can only build ontology in a specific domain for a specific application rather than a large and all-inclusive one.

Until now, we already have some research on military ontology. For example, [6] builds ontological models in support of information fusion and to exploits these models to provide enhanced knowledge management assistance to military commanders. [7] uses the DoD Enterprise Architecture Reference Models as a mid-level ontology to support an automatic NCW service binding for a business-level service composition. And there also exist some ontologies specially for battlefield information sharing. [8] designs the two-level domain ontology architecture from viewpoint of the military intelligent processing. [9] constructs military domain ontology, situation ontology and military rule ontology, which can transform information superiority to knowledge superiority. And [10] develops a precise domain ontology to overcome interoperability issues of different C4I architectures exchanging information on a common battle field. Although the above methods can mostly construct a complete ontology as the common concept foundation for different battle units, we still have no idea of the better semantic description method for information service and query. In addition, when the service discovery based on PE (Precondition and Effect) is widely discussed, we find PE can make BIS described more exactly. Therefore, we will begin with the requirement analysis for BIS, and focus on the approval of PE when designing the ontology.

III. Ontology Requirements Analysis

BIO should be able to support BIS publishing, query and match. BIS publishing needs to describe services according to ontology, so does BIS query. And BIS match needs ontology-based reasoning to find the most satisfying service.

A. Service Publishing and Query

Both of service publishing and query need service description, which should describe the inputs, outputs, preconditions, effects etc. The typical BIS model (shown in Fig.1) always takes one or several kinds of battlefield information as outputs. And it may have preconditions, which can make restrictions on outputs [11]. The necessary ontology support contains the following two aspects:

- BIS description must be understood by every combat platform. Therefore BIO should cover all the information concepts probably used. Each concept has fix and unique meaning without ambiguity. This meaning must be widely accepted, not be given at will.
- BIS makes restrictions on output target information in preconditions, which may give a value or a value range to a certain kind of target information. Therefore BIO should also provide the assignment way of target information.

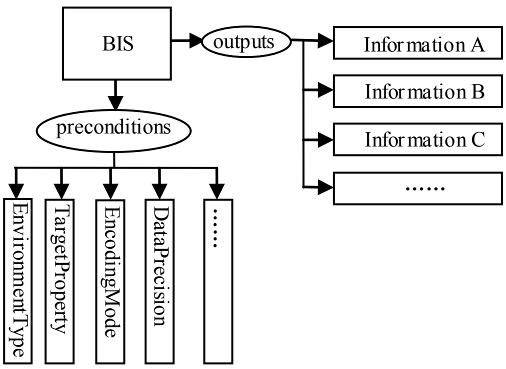

Figure 1. The basic model of battlefield service

B. Service Match

Whether a service can satisfy the request needs to be determined by service match. In classical keyword-based service match, only when descriptions from the service supplier and demander are identical, can we know they match. It only cares about the consistency of the words, rather than the meaning, resulting in losing a lot of actually matching services. To solve the above problems, semantic-based service match utilizes the concept relation in BIO to do logical reasoning. The needed reasoning support mainly includes:

- The inclusion or equivalence relation between outputs. Some services and queries describe the same battlefield information concept in different words, or the information provided by a service can cover the content demanded. With semantic reasoning, the above case should be correctly judged.
- The implication relation of preconditions. The precondition given by service demander must be able to imply the one given by service supplier. There should be some logical rules for precondition reasoning in ontology.
- The integrated reasoning of preconditions and outputs. In common service match, even based on semantic, preconditions and outputs will be separately matched[12]. But a case is usually encountered that an output concept under a certain precondition is equal to another concept. For example, if the output is *TargetInformation*, and the precondition is that *EnvironmentType* is *Air*, then their integration is equal to *AirTargetInformation*. BIO should support this kind of match.

IV. Ontology Construction

Given the concrete requirements for the ontology, we will discuss the construction process of BIO in this section. Web Ontology Language (OWL) is the most widely used ontology description language currently. The ontology satisfying all the requirements mentioned above can be carried out with the elements and their compositions provided by OWL.

A. Ontology Description Language

OWL language is developed in 2001, and became a W3C recommendation standard in February 2004. OWL provides three increasingly expressive sublanguages, the OWL Lite, the OWL DL and the OWL Full. OWL Lite can only support a classification hierarchy and simple constraints. OWL DL can provide the maximum expressiveness while retaining computational completeness and decidability. It includes all OWL language constructs, but they can be used only under certain restrictions. OWL Full includes all OWL language constructs without any restrictions. It supports users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees [13].

BIO cannot just contain a simple classification hierarchy obviously. However it's also unreasonable to get maximum expressiveness with no computational guarantees, because battlefield decision-making must have a result in time. Thus OWL DL should be chosen as the ontology description language (omit DL below). OWL describes the concepts and their relations in ontology mainly through the following elements [14]:

1) Class, individual, and property. A class denotes an abstract concept. An individual is an instance of the class. Individuals can be connected by properties. If a class is regarded as a set, then individuals will be the elements of the set, and a property will be a map from a set to another. Two classes may have subordination relation (*subClassOf*), equivalence relation (*equivalentClass*), and mutex relation (*disjointWith*). Two properties may have subordination relation (*subPropertyOf*).

2) Property characteristics and restrictions. Properties in OWL are divided into *ObjectProperty* and *DatatypeProperty* according to the data type of their value range. And there may be some *Restriction* on properties, for example, the value of a property can be restricted on having all values in some range (*allValuesFrom*), or having a value in some range (*someVlauesFrom*), or having a specific value (*hasValue*).

3) Cardinality of set. The OWL can make restrictions on the set which includes all the values of a property for a certain individual. The number (*cardinality*), the minimum number (*minCardinality*) and the maximum number (*maxCardinality*) of the set can be restricted.

B. Organization of Basic Concepts

The most important function of BIO is providing a common battlefield information semantic model, with which combat platforms can communicate without ambiguity. First of all, BIO should include all of the battlefield information concepts Even if the quantity of battlefield information concepts is limited, it's unsuitable to store all of them and their relations in a single ontology document, which will reduce the efficiency of logical reasoning and increase the probability of logical contradiction. So battlefield information concepts should be organized as a two-tier structure, Top-Frame Ontology (TFO) and Domain-Branch Ontology (DBO).

TPO stores the classification of battlefield information. The construction of it should follow the principles below.

1) Cover all aspects of battlefield information. According to the requirements of battlefield situation awareness, the information of a target can be divided in two parts: one is the original information of the target itself, such as the identity, characteristic, and motion state; the other one is the information refer to the target, including context information and data reliability.

2) Select suitable concept granularity. To promise an efficient logical reasoning of TFO, we must limit the ontology scale. Specific details should be put into DBO. On the other hand, to simplify the process of reasoning, we hope TFO can complete some concept match independently. The bottom concepts in TFO should at least be able to serve as outputs of BIS.

By the principles, we collect military terms, organize the hierarchy of them, and connect different layers with the relation *subClassOf*. TFO is built as shown in Fig. 2 (part of it), which needs further refinement by military domain experts.

In TFO, some concepts involve much content. For example, *CombatMission*, there are more than 300 common combat missions. Some others need further explanation. For example, *GeographicArea* should be further described in its position, shape, size, etc. We build DBOs for such concepts.

We can obtain DBO through direct construction with the domain knowledge, or from existing domain ontology. But the existing ones often need some modification. The method is as follows:

- Step 1. Let C be the concept to be extended in TPO. Choose the most exhaustive ontology O of the related domain to be the candidate ontology.
- Step 2. Calculate the similarity s_i of C and each concept C_i in O. Give a threshold ε. If $\max\{s_i\}$ is smaller than ε, then add C to O with a suitable location (may see [15] for the method). If $\max\{s_i\}$ is larger than ε, then take the corresponding concept C_i as the equivalent one of C in O.
- Step 3. If there is a concept C_j (not C) in both O and TFO, then delete it from O. If C is a subclass of C_j, then replace C_j with C in domain or range of all properties, otherwise delete all the properties related to C_j. After modifying, check all properties and their values and delete illegal assignment sentences.
- Step4. Delete all the concepts unrelated to C (or C_i) from O. Here we give the definition of related concept.

Definition 1. Regard a concept (class) as a point, and a property as an edge connecting two concepts (take *subClassOf* as special property). If concept C_1 is accessible to concept C_2 according to Graph Theory, then it's called C_1 is a **related concept** to C_2.

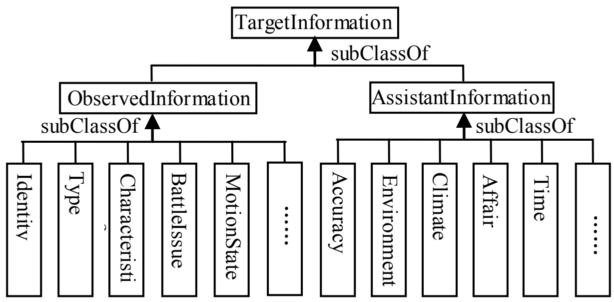

Figure 2. Part of top structure ontology

Following the above steps, we can obtain DBO from an existing ontology, and promise the consistency and simplicity of the result. After the organization of basic concepts, we get the accordance for terms used in BIS description. The computer can recognize the meaning of each concept, which will be used to compare the descriptions of the demander and provider and determine whether they match. Besides semantic match of concepts, logical reasoning for preconditions is also needed. We should add some properties to complete the expression and logical reasoning for preconditions.

C. *Design of Assignment Properties*

The preconditions of BIS always make value restrictions on parameters, maybe giving a specific value, or limiting to a range of values. The description of such preconditions requires BIO provide assignment properties for concepts. Besides, logical reasoning for preconditions determines the implication relations by comparing parameter values. Different data types need different comparison ways. Therefore we should also point out the data type of concept value meanwhile adding assignment properties.

In OWL, the value of a property can be restricted to a certain range by *Restriction*. But if we make restrictions on every concept, it will be a laborious work. It's better to classify the concepts by data type and restrict the range for each class[16]. When we have a new concept, we can describe its data type just by putting it into the corresponding class.

We define a property *hasOneValue* for top concept *TargetInformation*. Then define classes *IntTargetInformation*, *FloatTargetInformation*, *StringTargetInformation*, etc. as subclasses of *TargetInformation*, and respectively restrict their ranges of *hasOneValue* to the corresponding data type. Description sentences are shown below:

```
<owl:DatatypeProperty rdf:ID="hasOneValue">
   <rdfs:domain rdf:resource="#TargetInformation"/>
</owl: DatatypeProperty>
<owl:Class rdf:ID="IntTargetInformation">
   <rdfs:subClassOf rdf:resource="#TargetInformation"/>
   <owl:equivalentClass><owl:Restriction><owl:onProperty>
         <owl: DatatypeProperty rdf:about="#hasOneValue"/>
         </owl:onProperty>
      <owl:allValuesFrom rdf:resource
         ="http://www.w3.org/2001/XMLSchema#int"/>
</owl:Restriction></owl:equivalentClass></owl:Class>
```

If *TargetQuantity* is defined as a subclass of *IntTargetInformation*, then we can imply the value of *hasOneValue* for *TargetQuantity* must be an integer.

HasOneValue offers the way to assign the concept with a specific value. Adding properties *hasMaxValue* and *hasMinValue* as well, we can give a range to a concept. Now the preconditions of BIS can be almost described. Nevertheless there is a special case that the value of some concepts may be not a concrete data, but an object needing further description. For example, the value of *GeographicArea* is a certain spatial area, which needs to be described with extra properties. In order to deal with this case, we define a class *ObjectTargetInformation* as the subclass of *TargetInformation*, and define a property *hasObject* as its assignment property. The comparison function for objects should be extra provided in ontology. We add a property *CompareFunction*, of which the value is a link to a comparison function.

Ontology diagram in Fig.3 shows the total design of assignment properties.

D. *Integrated Reasoning Method*

For common service match, which matches the preconditions and the I/O separately, the ontology built in section 3.2 and 3.3 can offer sufficient support already. But in actual applications, such a situation often appears: some concept is equal to another one under a certain precondition. For example, a red apple is an apple whose color is red. From human's point of view, *RedApple* is obviously *Red Apple*. But in ontology used by computers, *RedApple* appears as a single concept which can't be divided into *Red* and *Apple* automatically. So it's unlikely to be equal to $(Color=Red)\wedge Apple$. If the above situation is expressed in the form of service, then the precondition of Service A is *Color=Red*, the output is *Apple*, and Service B has no precondition, the output is *RedApple*. Although these two services actually match, common service match won't get the right conclusion.

To solve the above problem, we should add corresponding relations in ontology, and do integrated reasoning of preconditions and outputs. To facilitate discussion, we define an expression form of binary relations.

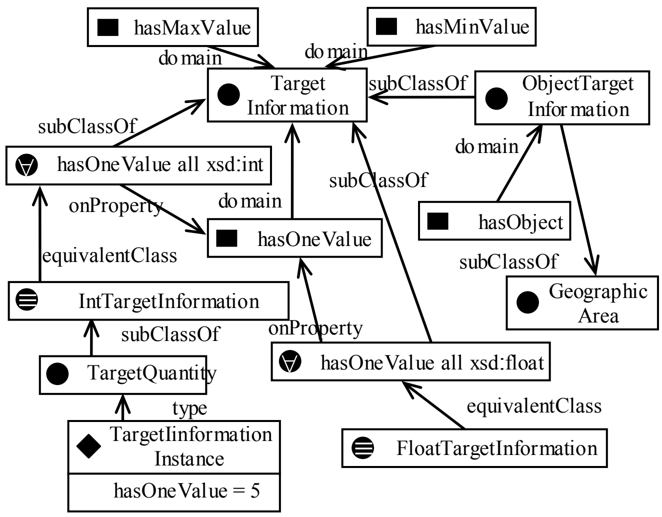

Figure 3. Design of ontology properties

Definition 2. A **binary relation** can be expressed as $\sigma=(R,x_1,x_2)$, where R is a predicate, x_1 and x_2 are two elements. It's called "x_1 and x_2 have relation R" or "x_1Rx_2."

On the definition, "the color is red" can be expressed as

$$\sigma_1=(\text{Is},\text{Color},\text{Red}). \quad (1)$$

When x_1 and x_2 are simple elements, σ is a first-order binary relation [17]. If x_1 or x_2 involves relations, then σ will be a second-order or even higher binary relation. For example, "Apple, of which the color is red, is equal to red apple" can be expressed as

$$\sigma_2 = (\text{EqualTo}, \sigma_1 \wedge \text{Apple}, \text{RedApple}) . \quad (2)$$

It is just the relations like σ_2 that should be added to the ontology. Unfortunately, a significant limitation of OWL is in allowing only first-order binary relations [18]. That is to say the relations like σ_2 can't be described in theory. However, we can transform high-order binary relations to first-order ones by defining new classes and properties. From intuitive observation, if we define " $\sigma_1 \wedge \text{apple}$ " as a class, then σ_2 becomes a first-order binary relation like σ_1. In daily life, it's common to see examples such as *RedApple, YellowBanana* and so on. Expressing all these high-order relations in transformation method is obviously impossible. Yet BIO has special characteristics, which offer the possibility of using transformation method:

1) The high-order predicate is unique. Here we just want to express the equivalence relation between the output with a precondition and the single output. So the predicate is just *EqualTo*.

2) The relation quantity is limited. According to the existing system (such as USA data link system) and actual requeirmnts of combat, environment type, identity attribute and few others are always taken as the criterions the classification of battlefield information. So there are just a limited number of relations we need to describe, such as *GroundTargetInformation*, *EnemyTargetInformation*, *AirTargetInformation*, etc.

The preconditions of BIS can be expressed as

$$\sigma_p = (\text{hasOneValue}, \text{PreConcept}, \text{Value}) , \quad (3)$$

where *hasOneValue* is the assignment property former defined, *PreConcept* is the concept restricted in the precondition, and *Value* is the given property value. Let $OutConcept_1$ and $OutConcept_2$ be the outputs. The relation to be added is

$$\sigma_e = (\text{EqualTo}, \text{OutConcept}_1 \wedge \sigma_p, \text{OutConcept}_2) . \quad (4)$$

Addressing the simplified problem, we propose the transformation method:

- Step 1. Define a new Class *PreConcept_Value* as the subclass of *PreConcept*. And restrict it to including and only including individuals whose value of *hasOneValue* is *Value*.
- Step 2. Define a property *has_PreConcept*, of which the domain is $OutConcept_1$ and the range is *PreConcept*.
- Step 3. For an individual I of $OutConcept_1$, if there is a value of *has_PreConcept* belonging to *PreConcept_Value*, then I belongs to $OutConcept_2$. And restrict $OutConcept_2$ to including and only including such individuals.

High-order binary relation can be described by the above steps (shown in Fig. 4). Step 1 takes all of the individuals satisfying σ_p together to a new class *PreConcept_Value*, which transforms $OutConcept_1 \wedge \sigma_p$ to $OutConcept_1 \wedge$ *PreConcept_Value*. Step 2 defines the relation $\wedge$ between $OutConcept_1$ and *PreConcept_Value* as a property *has_PreConcept*. Step 3 firstly takes the individuals satisfying $OutConcept_1 \wedge$ *PreConcept_Value* together as a class. Then expresses the equivalence relation (σ_e) between $OutConcept_2$ and the class by the phrase "include and only include."

We take *AirTargetInformation* and *TargetInformation* $\wedge$ *EnvironmentType=Air* for example to show the concrete description sentences. First of all, define a class *EnvironmentType_Air*, of which the description is:

```
<owl:Class rdf:ID="EnvironmentType_ Air ">
  <rdfs:subClassOf rdf:resource="#EnvironmentType"/>
  <owl:equivalentClass><owl:Restriction><owl:onProperty>
      <owl: DatatypeProperty rdf:about="#hasOneValue"/>
    </owl:onProperty>
    <owl:hasValue rdf:resource="#Air"/>
</owl:Restriction></owl:equivalentClass></owl:Class>
```

Then we define a Property *HasEnvironmentType*, whose domain is *TargetInformation*, range is *EnvironmentType*:

```
<owl:ObjectProperty rdf:ID="HasEnvironmentType">
  <rdfs:domain rdf:resource="#TargetInformation"/>
  <rdfs:range rdf:resource="#EnvironmentType"/>
</owl: DatatypeProperty>
```

Finally make restriction on *AirTargetInformation*:

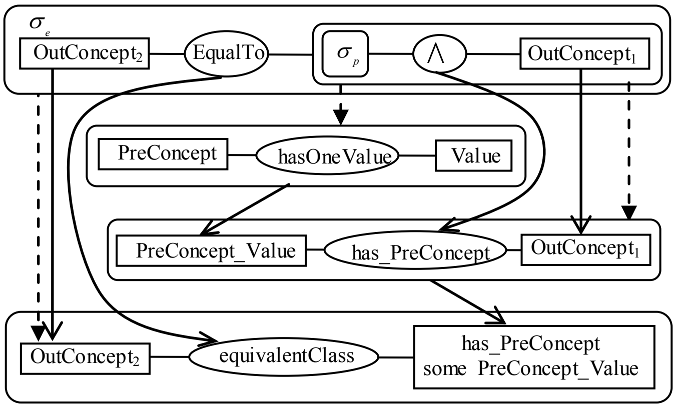

Figure 4. The transforming process of high-order relations

```
<owl:Class rdf:about="AirTargetInformation">
  <owl:equivalentClass><owl:Restriction><owl:onProperty>
      <owl: ObjectProperty rdf:about="#HaEnvironmentType"/>
    </owl:onProperty>
    <owl:someValuesFrom rdf:resource="#EnvironmentType_Air"/>
</owl:Restriction> </owl:equivalentClass></owl:Class>
```

The reasoning process is described as follows:

a) Necessity. Given *AirTargetInformation* as a subclass of *TargetInformation*, all the individuals of *AirTargetInformation* belong to *TargetInformation*.

b) Sufficiency. The individual of *TargetInformation* is connected with the one of *EnvironmentType* by *HasEnvironmentType*. When the precondition says the individuals of *EnvironmentType* has the value *Air* of property *hasOneVlaue*, these individuals belong to class *EnvironmentType_Air*. As former said, these individuals are connected with the ones of *TargetInformation* by property *HasEnvironmentType*. So under such condition the individuals of *TargetInformation* can meet the restriction that there is a value of *HasEnvironmentType* belongs to *EnvironmentType_Air*. So they belongs to *AirTargetInformation*.

c) Synthesizing *a)* and *b)*, we get the equivalence relation.

V. APPLICATION EXAMPLE

In this section, an example of BIS is given to show the process of BIO application.

There are two BISs, Service1 and Service2, and a Request, as shown in Tab. 1. Taking Service1 for example, the description method is given below. To keep compatible with OWL, we adopt OWL-S to describe service. Here are the main description sentences.

1) Preconditions:

```
<GeographicArea rdf:ID="GeographicAreaInstance">
   <hasObject rdf:resource=#Area1"/>
</GeographicArea >
<EnvironmentType rdf:ID="EnvironmentTypeInstance">
   <hasOneValue rdf:resource=#Air"/>
</EnvironmentType >
<TargetIdentity rdf:ID="TargetIdentityInstance">
   <hasOneValue rdf:resource=#Enemy"/>
</TargetIdentity >
```

TABLE I. THE EXAMPLES OF BIS PUBLISHING, QUERY AND MATCH

Service or Request	Precondition	Output
Service1	GeographicArea=Area1 EnvironmentType=Air TargetIdentity=Enemy	TargetInformation
Service2	GeographicArea =Area2 TargetIdentity =Enemy ∨ Friend	UnderwaterTarget-Information
Request	GeographicArea =Area3 TargetIdentity =Enemy	TargetVelocity TargetHeading TargetDepth TargetLongitude TargetLatitude

Among them, the value of *GeographicArea* is an object *Area1*, which needs a further description:

```
<RoundArea ID="Area1">
   <CenterPosition><Longitude>37.5</Longitude>
   <Latitude >28.6</Latitude></CenterPosition >
   <Radius>36</Radius>
</RoundArea>
```

2) Outputs:

```
<profile:hasOutput rdf:resource="#TargetInformation"/>
```

In service match, the preconditions match is done first. The description of *Area3* in Request is:

```
<RectangularArea ID="Area3">
   <CenterPosition><Longitude >37.5</Longitude>
      <Latitude >28.6</Latitude></CenterPosition>
   <LongEdgeDirection>25</LongEdgeDirection>
   <LongEdge>36</LongEdge><ShortEdge>25</ShortEdge>
</RectangularArea>
```

The geographic areas described in Service1 and in Request are shown in Fig. 5. According to the result of comparison function in ontology, *Area1* covers *Area3*. The restriction on *TargetIdentity* is both *Enemy* in Service1 and Request. Service1 have restriction on *EnvironmentType*, but Request not, which needs integrated reasoning with outputs. By the integrated reasoning rules in ontology, the output *TargetInformation* combined with *EnvironmentType=Air* is equal to *AirTargetInformation*. Then compare it with the outputs of Request. According to the subordination relations in ontology, *TargetDepth* doesn't belong to *AirTargetInformation*. Finally we can get the conclusion that Service1 doesn't match Request. Although *TargetInformation* includes *TargetDepth*, when combined with the precondition, it doesn't meet the request actually. From this we can see the necessity of integrated reasoning.

For Service2, according to the result of comparison function in ontology, *Area2* can also cover *Area3*. The value of *TargetIdentity* includes *Enemy* as well. So the preconditions completely match. And the output *UnderwaterTargetInformation* can cover all the outputs demanded by Request. Therefore, Service2 is the matching service.

VI. CONCLUSION AND EXPECTATION

In this paper, we propose an OWL-based construction method of BIO. According to the requirements analysis of BIS, we build TFO and DBO to organize battlefield information

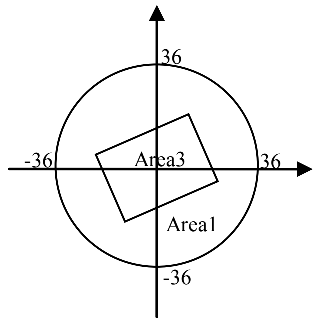

Figure 5. Geographic area covering diagram

concepts. The value assignment attributes are added to solve the assignment problem of concepts with different data types. And a new problem, the integrated reasoning of preconditions and outputs, is raised in this paper. We also provide the corresponding solution. BIO can provide semantic support for BIS publishing, query and match. It will help to increase the utilization rate of battlefield information resource and offer an effective way to semantic-based interoperation of combat platforms.

However we discuss the ontology requirements of BIS just in consideration of the typical service model. In further research, we should focus on more complicated service models.

Besides, we can also add some processing rules to ontology for data fusion and situation awareness, which will further enhance the intelligence of battlefield information system.

REFERENCES

[1] S. A. Davidson, W. Mucheng, S. Mohan, "GIG End-to-End Policy Based Network Management: A new approach to large-scale distributed automation," MILITARY COMMUNICATIONS CONFERENCE (MILCOM 2011), Baltimore, MD, pp.2011-2018, Nov. 2011.

[2] V. Florian, G. Neagu, "OGSA Compliant Service Administration to Support Workflow Execution on Grid," 2011 International Conference on Emerging Intelligent Data and Web Technologies (EIDWT), Tirana, Albania, pp. 159-164, Sept. 2011.

[3] J. L. Burbank, P. F. Chimento, B. K. Haberman, "Key Challenges of Military Tactical Networking and the Elusive Promise of MANET Technology," IEEE Communications Magazine, Vol. 44, No. 11, pp. 39-45, Nov. 2006.

[4] T. R. Gruber, "Toward Principles for the Design of Ontologies used for Knowledge sharing," International Journal of Human–Computer Studies, Vol. 6, pp. 907-928, 1995.

[5] A. G. Perez, V. R. Benjamins, "Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem Solving Methods," Proceedings of the IJCAl2 99 workshop on 0ntologies and Problem-Solving Methods(KRR5), pp. l-15, 1999.

[6] B. Brisset, "Ontological engineering for threat evaluation and weapon assignment: a goal-driven approach," 2007 10th International Conference on Information Fusion, Quebec, Que, pp. 1-7, July 2007.

[7] H. Sang-Kyu, L. Kwang-Je; B. Young-Tae, "A Method for Dynamic NCW Service Selection Based on EA Ontology," 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, Daegu, pp. 300 – 305, May 2009.

[8] J. Meiying, Y. Bingru; Z. Dequan; S. Weicong, "Research on Domain Ontology Construction in Military Intelligence," Third International Symposium on Intelligent Information Technology Application, IITA 2009, Nanchang, China, pp. 116 – 119, Nov. 2009.

[9] S. Junfeng, Z. Weiming, X. Weidong, Xu Zhenning, "Study on construction and integration of military domain ontology, situation ontology and military rule ontology for network centric warfare," The 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, Changsha, China, pp. 368 – 373, April 2005.

[10] A. S. Alghamdi, Z. Siddiqui, S. S. A. Quadri, "A Common Information Exchange Model for Multiple C4I Architectures," 2010 12th International Conference on Computer Modelling and Simulation (UKSim), Cambridge, pp. 538 – 542, March 2010.

[11] L. Yi, X. Jijuan, "An Approach to Battlefield Data Management of Targets and Environment," 2011 International Conference on Information Technology, Computer Engineering and Management Sciences (ICM), Nanjing, China, pp. 190-194, Sept. 2011.

[12] X. Binhong, Z. Yingjun, C. Lichao, "An Approach to Match Semantic Web Services Based on Multidimension," 2010 IEEE Asia-Pacific Services Computing Conference (APSCC), Hangzhou, China, pp. 683-689, Dec. 2010.

[13] OWL Web Ontology Language Overview. http://www. w3. org/TR/owl-features/.

[14] I. Bedini, C. Mathus, P. F. Patel-Schneider, A. Boran, B. Nguyen, "Transforming XML Schema to OWL Using Patterns," 2011 Fifth IEEE International Conference on Semantic Computing (ICSC), PaloAlto, CA, pp. 102-109, Sept. 2011.

[15] D. Jun, Research on some Key Issues for Domain Ontology Construction Base on the KDD, Nanjing: Nanjing University of Posts and Telecommunications, Oct. 2010.

[16] A. Dean, H. James, Semantic Web for the Working Ontologist, Beijing: Posts & Telecom Press, 2009.

[17] H. Ishikawa. Patel-Schneider, "Transformation of General Binary MRF Minimization to the First-Order Case," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No. 6, pp. 1234-1249, Jun. 2011.

[18] E. L. Martin, L. H. David, L. James, Handbook of Multisensor Data Fusion Theory and Practice, 2nd ed., New York: CRC Press, 2009, pp. 438-496.