

Study on Construction and Integration of Military Domain Ontology, Situation Ontology and Military Rule Ontology for Network Centric Warfare

Song Jun-feng, Zhang Wei-ming, Xiao Wei-dong, Xu Zhen-ning
(School of Information System and Management, The National University of Defense Technology,
Changsha 410073, China)
sjfemail@hotmail.com

Abstract

In network centric warfare, a suitable knowledge infrastructure is the premise of transforming information superiority to knowledge superiority, but current research works are few in this aspect. For building the knowledge infrastructure of NCW and capturing the knowledge superiority, in this paper, as the first step towards a practical knowledge infrastructure, MDOCA, SOCA and MROCA are proposed to construct military domain ontology, situation ontology and military rule ontology; the integration procedure of military domain ontology, situation ontology and military rule ontology is clarified; then a scenario knowledge infrastructure fragment is constructed by using approaches and procedure presented above.

1. Introduction

Network Centric Warfare (NCW)^[1,2] is the product of Information Age and military response to both the challenges and the opportunities created by the Information Age. For platform centric warfare, NCW is an essential transformation. By networking all battlespace entities, information superiority is obtained on the base of information sharing; knowledge superiority and decision superiority are enabled by using information superiority, so the combat power is improved dramatically.

In NCW, the sensor network's capability is much more stronger than ever, the force can get a mass of information about battlespace in real or near-real time. How to utilize the information about battlespace effectively and transform the information superiority into knowledge superiority is a key problem for NCW research.^[3] To solve this problem, firstly we need to establish a suitable knowledge infrastructure.

At present, researches concentrate on information superiority and proving information superiority's huge effectiveness to NCW.^[1,2,4] The research works about building knowledge infrastructure and capturing knowledge superiority are few.

In this paper, MDOCA(military domain ontology construction approach), SOCA(situation ontology construction approach) and MROCA(military rule ontology construction approach) are proposed to construct MDO(military domain ontology),

SO(situation ontology) and MRO(military rule ontology). The result of integrating MDO, SO and MRO is knowledge infrastructure for NCW.

2. Knowledge infrastructure and its component ontologies

Warfare takes place in physical domain, information domain and cognitive domain.^[1] For NCW, characteristics in physical domain, information domain and cognitive domain are different from platform centric warfare.

- In physical domain, all platforms are robustly networked achieving secure and seamless connectivity.^[1]
- In information domain, force has the capability to collect, share, access, and protect information.^[1] The force tries to capture information superiority and keep it.
- In cognitive domain, the force builds knowledge infrastructure so that knowledge representation, sharing and reasoning can be realized; the force tries to capture knowledge superiority and keep it. Based on information superiority and knowledge superiority, the force tries to capture decision superiority and keep it.

In NCW, the side that has information superiority, knowledge superiority and decision superiority can obtain full spectrum dominance.^[4]

In this paper, as the first step, knowledge infrastructure for NCW consists of 3 component ontologies: MDO, SO and MRO, but it can be expanded to include other component ontologies in future. In MDO and SO, description logic based ontology language is used to encode knowledge about military domain and situation; in MRO, Horn-like rule language is used to encode military rules.

3. Construct and integrate component ontologies

3.1. Ontology languages for constructing component ontologies

The familiar ontology languages are XOL, SHOE, OML, RDF(S), OIL, DAML+OIL and OWL. In terms of the expressivity, these languages can be arrayed in an

increasing order: XOL, RDF(S), SHOE, OML, OIL, DAML+OIL and OWL.^[5]

XML is used as the foundation for these languages' syntax. The formal foundation of XOL and SHOE is frame; the formal foundation of OML is conceptual graph; but frame and conceptual graph lack precise semantics. The expressivity of RDF(S) is so limited that it can only be regarded as primitive ontology language. To develop ontology language which can have more expressive power and characterize semantics more precisely, OIL^[6], DAML+OIL^[6] and OWL^[6,7] are all extended on the base of inheriting RDF(S)'s syntax and expressivity. European researchers establish OIL. DAML+OIL, established by Joint US/EU ad hoc Agent Markup Language Committee, is used as the starting point of OWL; OWL, established by W3C, is a new synthesis of research on ontology language.

OWL provides three increasingly expressive sublanguages: OWL Lite, OWL DL, OWL Full. The expressivity of OWL Lite is limited, but the efficiency of reasoning is preferable; OWL DL is as expressive as possible on the premise of preserving completeness and decidability of reasoning; OWL Full is the most expressive with no computational guarantees of reasoning. The formal foundation of OWL Lite is *SHIF(D)*^[6], key inference problems in OWL Lite have EXPTIME complexity, and there are already several optimized reasoners for logic equivalent to OWL Lite, such as description logic systems FaCT^[8] and RACER^[9]; the formal foundation of OWL DL is *SHOIN(D)*^[6], key inference problems in OWL DL have NEXPTIME complexity, it's difficult to reason.

Individuals (*O*) are much more widely used in constructing ontology than inverse roles (*I*)^[10], but the formal foundation of OWL Lite contains constructor *I* and doesn't contain constructor *O*. To obtain a better expressivity of knowledge, we use a limited OWL DL as ontology language for constructing military domain ontology and situation ontology (the restriction is to forbid using inverseOf). This limited OWL DL is named as OWL LDL, the formal foundation of OWL LDL is *SHON(D)*, the reasoning algorithm for *SHON(D)* is given in [10].

For constructing military rule ontology, we choose SWRL^[11] as ontology language. SWRL is based on a combination of the OWL DL and OWL Lite with the Unary/Binary Datalog RuleML sublanguages of the Rule Markup Language. Jess^[12] can be used as inference engine for rules encoded in SWRL.

3.2. Domain, domain conceptualization, domain ontology, ontology

In this paper, we distinguish the definitions of ontology and domain ontology. The definitions of domain, domain conceptualization, domain ontology and ontology is given as follows.

Definition1. Domain is a section of the world about which we wish to express some knowledge; domain

conceptualization is to abstract a set of terms and a set of knowledge from the domain in terms of the tasks to be solved and the ontological commitment of ontology language used; domain ontology is {the set of domain terms, the set of domain knowledge}, it's explicit specification of domain conceptualization, usually, we use ontology language to write down this specification; ontology is explicit specification of conceptualization, there is only one ontology about the world, no application needs to use the whole ontology.

In practical application, domain ontology or the integration of several domain ontologies is needed.

Definition2. Suppose the domain *D* can be divided into *n* subdomains, then the domain ontology of *D* can be obtained by integrating domain ontologies of these *n* subdomains.

3.3. MDOCA

We can't construct *MDO* for the whole military domain at one stroke; the feasible way is to build domain ontologies for subdomains in military domain, then integrate these domain ontologies to get *MDO*.

From the above paragraph, we know that MDOCA contains two basic approaches:

- **MDOCA_ I** : Construct domain ontology of subdomain in military domain.
- **MDOCA_ II** : Integrate domain ontologies of several subdomains in military domain.

The procedure of MDOCA is:

First, use MDOCA_ I to construct domain ontology of each subdomain in military domain.

Second, use MDOCA_ II to integrate domain ontologies of those subdomains, the result of integration is *MDO*.

Third, if there is new subdomain, for which we need to construct domain ontology, then using MDOCA_ I to construct it.

Fourth, after domain ontology of new subdomain is constructed by using MDOCA_ I, using MDOCA_ II to integrate it with existing *MDO*, the result of integration is *MDO*.

By using MDOCA, we can obtain *MDO*, which is more and more perfect.

3.3.1. MDOCA_ I : The procedure of MDOCA_ I is introduced as follows.

First. Construct the set of domain terms—In terms of the tasks to be solved and the ontological commitment of OWL LDL, construct the set of domain terms. The ontology engineers do most of the work in this phase.

Definition3. OWL LDL makes the commitment that the world consists of class, datatype, object property, datatype property, individual, data value.

The ontology engineers construct the set of domain terms according to the following two principles:

1. Determine the terms in the set of domain terms according to the task to be solved; not everything in the domain should be expressed, only relevant things should be expressed.
2. Determine the terms in the set of domain terms according to the ontological commitment of OWL LDL; there are six types of terms in the set of domain terms according to the ontological commitment of OWL LDL, class, datatype, object property, datatype property, individual, data value.

Second. Construct domain ontology—By using OWL LDL's syntax and the terms in the set of domain terms, construct the set of domain knowledge; domain ontology is {the set of domain terms, the set of domain knowledge}, so here we have domain ontology. The ontology engineers do most of the work in this phase.

Based on the corresponding relation between OWL LDL and *SHON(D)*, the class, datatype, object property, datatype property, individual, data value encoded in OWL LDL can be corresponded into concept, concrete datatype, abstract role, concrete role, individual, value encoded in *SHON(D)*; the axiom and fact encoded in OWL LDL can be corresponded into Tbox, Abox, role hierarchy encoded in *SHON(D)*.

Third. Call a suitable reasoning system to check consistency—Call reasoning system, which can accomplish the reasoning tasks of OWL LDL (system which implements the reasoning algorithm for *SHON(D)*, such as Pellet^[13]), to check consistency. This phase can prevent inconsistent knowledge to occur. The machines do most of the work in this phase.

3.3.2. MDOCA_II: The basic idea of MDOCA_II is embodied in theorem 1.

Theorem1. There are m domains, and we have constructed domain ontologies for these domains. Suppose we need to integrate these domain ontologies: $do_1, do_2, \dots, do_m$, these domain ontologies are encoded in OWL LDL and identified by URI: $uri_1, uri_2, \dots, uri_m$; $\forall i \in [1, m]$, $do_i = \{v_i, k_i\}$, v_i is the set of domain terms, k_i is the set of domain knowledge, then

1. $Integration(do_i)$, the integration of m domain ontologies is :

$$Integration(do_i) = \{ \bigcup_{i=1}^m v_i, (\bigcup_{i=1}^m k_i) \cup IBK \}.$$

In the equation above, IBK (integration built knowledge) is the set of knowledge encoded by using OWL LDL and terms in $\bigcup_{i=1}^m v_i$, IBK

is different from $\bigcup_{i=1}^m k_i$ ($\bigcup_{i=1}^m k_i$ reflects union of knowledge from m domains), IBK reflects the newly appeared knowledge for the

integration of m domains, and knowledge in $(\bigcup_{i=1}^m k_i) \cup IBK$ should keep consistency.

2. $\#(\bigcup_{i=1}^m v_i) = \sum_{i=1}^m \#(v_i)$, “#” denotes cardinality of set.

Proof: $Integration(do_i)$, the integration of m

domain ontologies, is domain ontology of the domain that is the integration of m domains; and $\forall i \in [1, m]$, $do_i = \{v_i, k_i\}$; so we can use the terms defined in m domain ontologies as terms of $Integration(do_i)$, then

we obtain the following equation,

$$Integration(do_i) = \{ \bigcup_{i=1}^m v_i, \text{ the set of domain knowledge for the integration of } m \text{ domains} \}.$$

The set of domain knowledge for the integration of m domains can be divided into two parts. The first part is $\bigcup_{i=1}^m k_i$, which reflects union of knowledge from m domain; the second part is IBK, which reflects the newly appeared knowledge for the integration of m domains, knowledge in IBK can't be achieved in single domain of m domains. Knowledge in $(\bigcup_{i=1}^m k_i) \cup IBK$ should keep consistency, or it denotes contradiction.

For using URI naming mechanism, every term is unique, so $\#(\bigcup_{i=1}^m v_i) = \sum_{i=1}^m \#(v_i)$.

End.

Based on theorem 1, during the process of integrating domain ontologies, because $\bigcup_{i=1}^m v_i$ and

$\bigcup_{i=1}^m k_i$ can be obtained by utilizing the existing domain ontologies, so the key of integrating domain ontologies is how to construct IBK. Based on this idea, the procedure of MDOCA_II is introduced as follows.

First. Construct IBK—Use uri_IBK to identify IBK (uri_IBK can be a newly created URI, or an existing URI), suppose term t is defined in do_i , if t is to be quoted or defined in IBK, then in IBK, we can use $uri_i\#t$ to denote term t ; after all the quotations and definitions are completed, we have $Integration(do_i)$.

The ontology engineers do most of the work in this phase.

Second. Check consistency—Call reasoning system (just as reasoning system used in MDOCA_I) to check consistency. This phase can prevent inconsistent knowledge to occur. The machines do most of the work in this phase.

3.4. SOCA and MROCA

A situation corresponds to the limited parts of reality we perceive, reason about, and live in. Compared with relative static *MDO*, *SO* is dynamic with time changing. We can treat situation ontology as domain ontology about situation, and situation ontology is integration of many domain ontologies of subdomains of situation. So the procedure of *MDOCA* can be used as procedure of *SOCA* with a little change (the change is to consider situation, instead of military domain, as domain).

In *NCW*, we need not only knowledge about military domain and situation, but also knowledge about military rule, so we choose *SWRL* to encode Horn-like rules. The procedure of *MROCA* is:

First. Military Rule extraction—There are many explicit and implicit rules for combat, as we choose *SWRL* to encode rules, we should extract those rules that can be transformed into Horn-like rules. The ontology engineers do most of the work in this phase.

Second. Encoding—Encode those rules in *SWRL*.

3.5. Procedure of Integrating component ontologies

After *MDO*, *SO* and *MRO* are constructed, we need to integrate them to obtain knowledge infrastructure. Consider *MDO*, *SO* and *MRO* as domain ontologies, based on theorem 1, the procedure of integration is:

First. Construct IBK of *MDO*, *SO* and *MRO*—Use *uri_IBK* to identify IBK (*uri_IBK* can be a newly created URI, or an existing URI), suppose term *t* is defined in *XO* (*XO* is either *MDO*, or *SO*, or *MRO*), if *t* is to be quoted or defined in IBK, then in IBK, we can use *uri_XO#t* to denote term *t*; after all the quotations and definitions are completed, we have the integration of *MDO*, *SO* and *MRO*. The ontology engineers do most of the work in this phase.

Second. Check consistency—Call suitable reasoning system (which can accomplish the reasoning tasks of the composition of *OWL LDL* and *SWRL*, such as *Vampire*^[14]) to check consistency. This phase can prevent inconsistent knowledge to occur. The machines do most of the work in this phase.

4. Application example

To illustrate the approaches and procedure, we construct a *SKIF* (scenario knowledge infrastructure fragment) by using them. That scenario reflects air situation at a certain time and certain airspace between red side and blue side; all the data used are hypothetical.

To be compact, here we use description logic's syntax and *SWRL*'s abstract syntax to encode *SKIF*; the corresponding relation between description logic's syntax and *OWL LDL* can be found in [6], the corresponding relation between *SWRL*'s abstract

syntax and *SWRL*'s *RDF/XML* syntax can be found in [14].

The procedure of constructing this *SKIF* is as follows:

First, use *MDOCA_I* to construct domain ontologies for the domain of fundamental elements in battlespace, the radar domain, the air to air missile domain one by one.

Second, use *MDOCA_II* to integrate those domain ontologies obtained in the first step, then we have *MDO α*. *MDO α* can be encoded in three parts:

1. Domain ontology for the domain of fundamental elements in battlespace — URI is *uri_BattleSpace*; there are concepts: *Sensor*, *DecisionMaker*, *Shooter*, *Platform*, *RedSide*, *BlueSide*; there is concrete role: *hasTime*; the range of *hasTime* is *dateTime* defined in *XMLSchema*.

```
Sensor;  
DecesionMaker;  
Shooter;  
Platform;  
RedSide;  
BlueSide;  
T ⊆ ∀hasTime.http://www.w3.org/2000/10/XMLSchema#dateTime;
```

2. Domain ontology for the radar domain — URI is *uri_Radar*; *Radar* is *Detector*; there are three types of *Radar*: *Radar1* (its maximal sense range is 80, its maximal sense angle is 60), *Radar2* (its maximal sense range is 100, its maximal sense angle is 60), *Radar3* (its maximal sense range is 200, its maximal sense angle is 360); there are concrete roles: *hasMaxSenseRange*, *hasMaxSenseAngle*; the range of these two concrete roles is the same, *decimal* defined in *XMLSchema*.

IBK (integration needed): To integrate with domain ontology for the domain of fundamental elements in battlespace, we define that *Detector* in this domain ontology is *Sensor* in domain ontology for the domain of fundamental elements in battlespace.

Note: The unit of *hasMaxSenseRange* value is kilometer, the unit of *hasMaxSenseAngle* value is degree.

```
Detector ⊆ uri_BattleSpace#Sensor;  
Radar ⊆ Detector;  
Radar1 ⊆ hasMaxSenseRange:80 ⊓ Radar ⊓ hasMaxSenseAngle:60;  
Radar2 ⊆ hasMaxSenseRange:100 ⊓ Radar ⊓ hasMaxSenseAngle:60;  
Radar3 ⊆ hasMaxSenseRange:200 ⊓ Radar ⊓ hasMaxSenseAngle:360;  
T ⊆ ∀hasMaxSenseRange.http://www.w3.org/2000/10/XMLSchema#decimal;  
T ⊆ ∀hasMaxSenseAngle.http://www.w3.org/2000/10/XMLSchema#decimal;
```

3. Domain ontology for the air to air missile domain — URI is *uri_AAMissile*; *AAMissile* is *Attacker*; there are two types of *AAMissile*: *Missile1* (its maximal kill range is 40), *Missile2* (its maximal kill range is 50); there is concrete role: *hasMaxKillRange*; the range of *hasMaxKillRange* is *decimal* defined in *XMLSchema*.

IBK (integration needed): To integrate with domain ontology for the domain of fundamental elements in battlespace, we define that *Attacker* in this domain ontology is *Shooter* in domain ontology for the domain of fundamental elements in battlespace.

Note: The unit of *hasMaxKillRange* value is kilometer.

Attacker $\sqsubseteq$ uri_BattleSpace#Shooter;
 AAMissile $\sqsubseteq$ Attacker;
 Missile1 $\sqsubseteq$ hasMaxKillRange:40 $\sqcap$ AAMissile;
 Missile2 $\sqsubseteq$ hasMaxKillRange:50 $\sqcap$ AAMissile;
 $\top \sqsubseteq \forall$ hasMaxKillRange.http://www.w3.org/2000/10/XMLSchema#decimal;

Third, use MDOCA_ I to construct domain ontology for the military plane domain.

4. Domain ontology for the military plane domain — URI is uri_MP; there are concepts: *Plane*, *FighterPlane*, *AWACSPane*, *FighterPlane F1* (carry *M1 AAMissile*, *R1 Radar*), *FighterPlane J2* (carry *M2 AAMissile*, *R2 Radar*), *AWACSPane Y3* (carry *R3 Radar*); there are abstract roles: *hasMissileType*, *hasRadarType*; there are concrete roles: *hasDirection*, *hasX*, *hasY*; the domain of *hasDirection*, *hasX*, *hasY* is the same, *Plane*; the range of *hasDirection*, *hasX*, *hasY* is the same, *decimal* defined in *XMLSchema*.

Note: To be compact, here we just consider that all the planes are on the same height, so we can use coordinate of two dimensions to locate the plane; the unit of *hasX* and *hasY* value is kilometer; the unit of *hasDirection* value is degree.

Plane;
FighterPlane $\sqsubseteq$ *Plane*;
AWACSPane $\sqsubseteq$ *Plane*;
 $F1 \sqsubseteq \text{FighterPlane} \sqcap \exists \text{hasMissileType.M1} \sqcap \exists \text{hasRadarType.R1}$;
 $J2 \sqsubseteq \text{FighterPlane} \sqcap \exists \text{hasMissileType.M2} \sqcap \exists \text{hasRadarType.R2}$;
 $Y3 \sqsubseteq \text{AWACSPane} \sqcap \exists \text{hasRadarType.R3}$;
 $\geq 1 \text{hasDirection} \sqsubseteq \text{Plane}$;
 $\top \sqsubseteq \forall \text{hasDirection.http://www.w3.org/2000/10/XMLSchema\#decimal}$;
 $\geq 1 \text{hasX} \sqsubseteq \text{Plane}$;
 $\top \sqsubseteq \forall \text{hasX.http://www.w3.org/2000/10/XMLSchema\#decimal}$;
 $\geq 1 \text{hasY} \sqsubseteq \text{Plane}$;
 $\top \sqsubseteq \forall \text{hasY.http://www.w3.org/2000/10/XMLSchema\#decimal}$;

Fourth, use MDOCA_ II to integrate domain ontology for the military plane domain with the existing *MDO* α , then we can obtain a better *MDO* β .

After the fourth step, we have IBK1, which contains knowledge needed for integration.

5. IBK1 — Create a new URI, uri_IBK1, to identify IBK1; *FighterPlane* and *AWACSPane* in domain ontology for the military plane domain belong to *Platform* in domain ontology for the domain of fundamental elements in battlespace; *M1* and *M2* in domain ontology for the military plane domain respectively correspond to *Missile1* and *Missile2* in domain ontology for the air to air missile domain; *R1*, *R2*, *R3* in domain ontology for the military plane domain respectively correspond to *Radar1*, *Radar2*, *Radar3* in domain ontology for the radar domain.

uri_Mp#FighterPlane $\sqsubseteq$ uri_BattleSpace#Platform;
 uri_Mp#AWACSPane $\sqsubseteq$ uri_BattleSpace#Platform;
 uri_Mp#M1 $\sqsubseteq$ uri_AAMissile#Missile1;
 uri_Mp#M2 $\sqsubseteq$ uri_AAMissile#Missile2;
 uri_Mp#R1 $\sqsubseteq$ uri_Radar#Radar1;
 uri_Mp#R2 $\sqsubseteq$ uri_Radar#Radar2;
 uri_Mp#R3 $\sqsubseteq$ uri_Radar#Radar3;

IBK1, domain ontology for the military plane domain, *MDO* α form *MDO* β .

Fifth, use SOCA to construct situation ontology at a certain time and certain airspace, integrate it with *MDO* β , and then we obtain *SKIF* α .

6. Situation ontology at a certain time and certain airspace — URI is uri_SOF; time is 2002-12-22T21:53:33Z; two *F1* (*Blue011*, *Blue012*) of blue side; one *J2* (*Red001*) and one *Y3* (*Red002*) of red side; *Blue011* at (-50,100), *Blue012* at (0,200), *Red001* at (-50,50), *Red002* at (0,0); the plane of blue side fly along 270 degree direction, the plane of red side fly along 90 degree direction.

uri_BattleSpace#hasTime(now,2002-12-22T21:53:33Z);
 uri_BattleSpace#BlueSide(*Blue011*);
 uri_BattleSpace#BlueSide(*Blue012*);
 uri_BattleSpace#RedSide(*Red001*);
 uri_BattleSpace#RedSide(*Red002*);
 uri_MP#F1(*Blue011*);
 uri_MP#F1(*Blue012*);
 uri_MP#J2(*Red001*);
 uri_MP#Y3(*Red002*);
 uri_MP#hasDirection(*Blue011*,270);
 uri_MP#hasDirection(*Blue012*,270);
 uri_MP#hasDirection(*Red001*,90);
 uri_MP#hasDirection(*Red002*,90);
 uri_MP#hasX(*Blue011*,-50);
 uri_MP#hasY(*Blue011*,100);
 uri_MP#hasX(*Blue012*,0);
 uri_MP#hasY(*Blue012*,200);
 uri_MP#hasX(*Red001*,-50);
 uri_MP#hasY(*Red001*,50);
 uri_MP#hasX(*Red002*,0);
 uri_MP#hasY(*Red002*,0);

Situation ontology at a certain time and certain airspace and *MDO* β form *SKIF* α .

Sixth, use MROCA to construct *MRO* fragment, integrate it with *SKIF* α , and then we obtain *SKIF* β , as shown in figure 1.

7. *MRO* fragment — URI is uri_MROF; if *x* and *y* belong to the same side and *x* locks *y*, then *x* should hold fire to *y*; if *x* and *y* belong to different sides and *x* locks *y*, then *x* should fire at *y*.

uri_BattleSpace#RedSide(*x*) $\wedge$ uri_BattleSpace#RedSide(*y*) $\wedge$
 lock(*x*,*y*) $\rightarrow$ holdFire(*x*,*y*);
 uri_BattleSpace#BlueSide(*x*) $\wedge$ uri_BattleSpace#BlueSide(*y*) $\wedge$
 lock(*x*,*y*) $\rightarrow$ holdFire(*x*,*y*);
 uri_BattleSpace#RedSide(*x*) $\wedge$ uri_BattleSpace#BlueSide(*y*) $\wedge$
 lock(*x*,*y*) $\rightarrow$ fire(*x*,*y*);
 uri_BattleSpace#BlueSide(*x*) $\wedge$ uri_BattleSpace#RedSide(*y*) $\wedge$
 lock(*x*,*y*) $\rightarrow$ fire(*x*,*y*);

Note: The big circle is sense range of *Red002*, the bigger sector is sense range of *Red001*, the smaller sector is kill range of air to air missile carried by *Red001*.

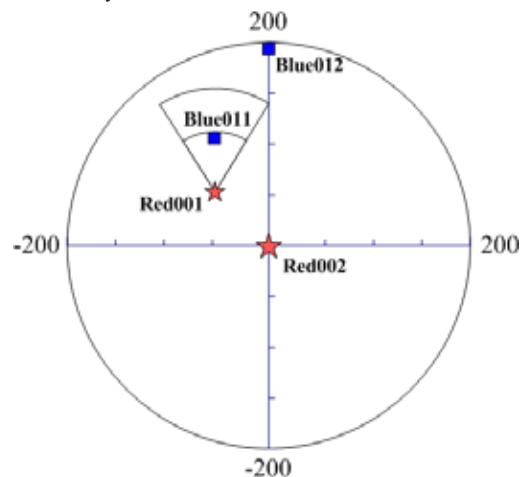


Figure1. SKIF β

In traditional platform centric warfare, platform uses its own sensor and shooter to fight; it's difficult to realize information sharing. As example shown in figure 1, Red001 can locate Blue011, but it can't locate Blue012; Red002 can locate Blue011 and Blue012, but it can't share this information with Red001.

In NCW, we can realize information sharing and obtain a mass of information. As example shown in figure 1, Red001 can share the target location information found by Red002, so it can know the air situation more comprehensively.

To utilize the vast information obtained by sensor network, we need knowledge infrastructure. In this paper, we propose approaches and procedure to construct knowledge infrastructure. Based on knowledge infrastructure, knowledge representation, sharing and reasoning can be realized, so knowledge superiority can be obtained and kept. As example shown in figure 1, *SKIF* β can be used for realizing knowledge representation, sharing and reasoning. For example, based on *SKIF* β , we can design automatic warning program for Red001 using reasoning system; so the decisionmaker on Red001 can be informed about the air situation comprehensively and timely — Blue011 is in the range of Missile2, Missile1 carried by Blue011 and Blue012 has maximal kill range of 40 kilometer, etc.

5. Conclusion and expectation

In this paper, we utilize the research works in description logic, ontology language and ontology integration; for establishing knowledge infrastructure of NCW and capturing knowledge superiority, we use the integration of *MDO*, *SO* and *MRO* as knowledge infrastructure of NCW, and we propose approaches to construct and integrate *MDO*, *SO* and *MRO*.

Future work will include a detailed study of what kinds of other component ontologies are needed to be included in knowledge infrastructure for NCW. An implementation is also planned for experimentation, and at present, we would like to choose protégé^[15] as our basic tool.

6. Acknowledgement

Supported by the National Natural Science Foundation of China under Grant No.60172012 and the Natural Science Foundation of HuNan Province under Grant No.03JJY3110.

7. References

[1] Department of Defense, USA, *Network Centric Warfare, DoD Report to Congress*, DoD, Washington DC, 2001.

[2] David S. Alberts, John J. Garstka and Frederick P. Stein, *Network centric warfare: developing and leveraging information superiority*, DoD CCRP, Washington DC, 1999.

[3] Martin C. Libicki and Stuart E. Johnson, *Dominant Battlespace Knowledge*, National Defense University Press, Washington DC, 1995.

[4] Joint Chiefs of Staff, USA, *Joint Vision 2020*, <http://www.dtic.mil/jointvision/>.

[5] Song Junfeng, etc, "Research on the formal foundation of OWL DL", *Mini-Micro Systems*, to appear.

[6] Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen, "From SHIQ and RDF to OWL: The making of a web ontology language", *Journal of Web Semantics*, 2003, 1(1), pp. 7-26.

[7] Mike Dean and Guus Schreiber, *OWL Web Ontology Language Reference*, <http://www.w3.org/TR/owl-ref/>, 2004.

[8] Ian Horrocks, "Using an expressive description logic: FaCT or fiction?", *Proceedings of the 6th Int. Conf. on Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, Trento, Italy, 1998, pp.636-647.

[9] Volker Haarslev and Ralf Möller, "RACER system description", *Proceedings of IJCAR2001*, Springer, Siena, Italy, 2001, pp. 701-705.

[10] Ian Horrocks and U. Sattler, "Ontology reasoning in the *SHOQ(D)* description logic", *Proceedings of the 17th Int. Joint Conf. on Artificial Intelligence*, Morgan Kaufmann, Washington, USA, 2001, pp. 199-204.

[11] Ian Horrocks, etc, *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, <http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/>.

[12] Jess, <http://herzberg.ca.sandia.gov/jess/>.

[13] Pellet, <http://www.mindswap.org/2003/pellet/index.shtml>.

[14] Dmitry Tsarkov, etc, "Using Vampire to Reason with OWL", *Proc. of the 2004 International Semantic Web Conference*, Springer, 2004, pp. 471-485

[15] Protégé, <http://protege.stanford.edu/>.