




Layered ontology-based multi-sourced information integration for situation awareness

Jongmo Kim¹ · Junsik Kong¹ · Mye Sohn¹  · Gyudong Park²

Accepted: 8 January 2021 / Published online: 17 February 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

Abstract

Data and information produced in network-centric environments are large and heterogeneous. As a solution to this challenge, ontology-based situation awareness (SA) is gaining attention because ontologies can contribute to the integration of heterogeneous data and information produced from different sources and can enhance knowledge formalization. In this study, we propose a novel method for enhancing ontology-based SA by integrating ontology and linked open data (LOD) called a multi-layered SA ontology and the relations between events in the layer. In addition, we described the characteristics and roles of each layer. Finally, we developed a framework to perform SA rapidly and accurately by acquiring and integrating information from the ontology and LOD based on the multi-layered SA ontology. We conducted three experiments to verify the effectiveness of the proposed framework. The results show that the performance of the SA of our framework is comparable to that of domain experts.

Keywords Situation awareness · Ontology-based integration · Multi-sourced information integration · Linked open data

✉ Mye Sohn
myesohn@skku.edu

Jongmo Kim
dignityc@skku.edu

Junsik Kong
jskong94@skku.edu

Gyudong Park
iobject@add.re.kr

¹ Department of Industrial Engineering, Sungkyunkwan University, Suwon, Korea

² 2nd R&D Institute, Agency for Defense Development, Seoul, Korea

1 Introduction

The rapid development of information communication technology has spurred the transition to networked environments in which all the elements (or objects) are interconnected. This transition has occurred in many real-world applications, particularly in the military field [1]. In network-centric environments (NCEs), all the elements of warfare from sensors to shooters are connected, and their connections are becoming tighter. The data and information produced by the connected elements should be transmitted to the right person to make accurate decisions at the right time. However, the data and information produced in NCEs are large, heterogeneous, and dynamic, thereby making it difficult to recognize the exact situation using them. To overcome this challenge, various methods have been proposed to gain situation awareness (SA) in many research fields such as businesses and battlefields [1–4]. Among these methods, ontology-based SA has gained attention because ontology can contribute to the integration of heterogeneous data and information from multi-sources and can enhance knowledge formalizations using an explicit representation of the concepts and identification of relations between them [5]. A critical issue that many researchers have aimed to address using ontology-based SA is the syntactic and semantic heterogeneity of data and information to be resolved for the SA using integrated and formalized information. Thus, to represent a situation clearly and concisely, most SA ontologies have defined the relations to represent the complex structures of the situations and concepts to generalize the objects or elements included in a situation for enhanced interoperability [5].

There are two ways of constructing SA ontologies. In the first method, domain experts gather the multi-sourced domain knowledge required for the awareness of specific situations. Then, knowledge engineers use it to develop a single and complete SA ontology to elucidate the concepts and vocabulary used in the situation. This method results in rapid and precise SAs. However, it is very inflexible because the developed SA ontologies are suited to a specific situation due to their complexity [2]. The many developed SA ontologies may cause heterogeneity between them. In addition, the quality or concept formalization levels of the developed SA ontology may differ depending on the capability of the domain experts or knowledge engineers. The second method for constructing SA ontologies reuses and integrates the domain ontologies of multi-sources that have already been created by developing the core ontology as extension points to connect ontologies. Although this method can integrate domain knowledge efficiently using abstract concepts at the upper level, it still has difficulty solving syntactic, semantic, and structural heterogeneity in integrating domain ontologies developed for different purposes [6]. The latter method, i.e., the core ontology development method, is emerging as the primary method for constructing SA

ontologies due to the rapid increase in the scale and complexity of the information involved, which no longer allows any one ontology to cover it.

However, there are several limitations associated with the core ontology development methods. First, while integrating the domain ontologies, the existing core ontology only considers the relationships between the classes of the domain ontologies [7–9]. As a result, these methods cannot properly reflect data values and their relationships, which can change dynamically depending on the situation. It leads to the low utility of the core ontology. Second, it is assumed that the core ontology already recognizes the structure of the domain ontology in development [10]. However, this assumption is no longer suitable for networked environments, where the number of the domain ontologies increases exponentially [11]. Finally, the core ontology integrates all the structures of the domain ontologies or extracts a big part of the domain ontologies that meet predefined constraints [12]. The resultant integration of the core ontology may produce a huge ontology, including unnecessary information for SA. The enormous size and complexity of the core ontology require the decision-maker to reinterpret the integrated information and try out all available choices to filter out useless information.

To overcome these limitations, we developed a novel framework for SA using layered SA ontologies that can integrate heterogeneous and multi-sourced information. In this framework, we employed the domain ontology for modeling concepts and relationships between these concepts and integrated the entire range of domain knowledge involved by reusing the domain ontologies. In addition, to overcome the problems related to the large and dynamic nature of the data and information produced in NCEs, we utilized the linked open data (LOD) to discover domain ontologies to deal with a variety of situations, extract the partial structures from the domain ontologies as sub-ontologies, and integrate these extracted sub-ontologies to help understand a situation in a holistic view.

A preliminary version of this paper has been presented as a conference paper [13]. The novelties and major contributions of this study are as follows:

- Improvement of the layered SA ontology structure previously, we designed four-layered command and control (C2) ontologies. However, to utilize various ontologies as an integrated whole, the interaction between the layers must be clarified. In this study, we clarify the interaction between the layers and propose an integration process using the layered SA ontology.
- Development of the graph entropy-based sub-ontology extraction method we assumed that many domain ontologies have been developed in multi-source, and they are needed for the SA. Some of these domain ontologies may be related to the current situation, while some may not. In addition, even if related domain ontologies are needed, a small part of them is essential for the SA. Thus, we propose a method for extracting only the related sub-ontologies from domain

ontologies, because the large ontology generated by the integrated structure of domain ontologies places a heavy burden on decision-makers.

- Development of the LOD-based sub-ontology integration method integrating multiple sub-ontologies is necessary because they are extracted from different domain ontologies. Although the sub-ontologies are developed based on the upper ontology, they cannot be easily integrated because the domains and granularities are different. To resolve this issue, we exploit the simple knowledge organization system (SKOS) in LOD as the core structure of the integrated ontology and join the extracted sub-ontologies to it.
- Execution of performance evaluation to verify the effectiveness and feasibility of the proposed framework, we conduct three experiments. We conduct the first experiment to demonstrate the effectiveness of extracting sub-ontologies by scenario. To this end, the number of the classes in the domain ontologies will be compared with that in the sub-ontologies extracted from them. We conduct the second experiment to demonstrate the appropriateness of the integrated ontology's volume based on the SKOS. Finally, the third experiment attempts to demonstrate that the integrated ontologies are properly generated, and we compared the results of the SA inferred using the integrated ontology with those inferred by domain experts.

The rest of this paper is organized as follows: Section 2 summarizes the related works. Section 3 describes the structure and properties of the layered SA ontology. Section 4 describes the overall architecture of the integration framework and the procedure for the interaction between the ontology layers. Section 5 demonstrates the performance of the proposed system by several experiments. Finally, Sect. 6 presents the conclusion and suggestions for future research.

2 Literature review

2.1 Ontology-based SA in C2

C2 is an area where ontology is actively used for the SA. Most studies attempting to utilize ontology for representing the knowledge of C2 systems have proposed a three-layered structure comprising an upper layer, middle layer, and domain [14–17]. Smith [16, 17] proposed a three-layered structure comprising the universal core semantic layer (UCore SL), C2 core ontology, and domain ontologies. In this model, the C2 core ontology was proposed as a middle layer ontology dealing with only general terms that can be used by a

broad stakeholder base within the C2 domain. Ra [15] has proposed a method combining the characteristics of top-down and bottom-up approaches by utilizing the kernel ontology as the middle layer ontology. Kabilan [14] proposed a framework that used SUMO as its upper ontology. In this framework, the middle layer ontology was constructed based on the Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM). However, by following these three-layer structures, ontologies or their relationships cannot be changed dynamically. In addition, it is difficult to understand the specific relationships between instances across distinct domain ontologies.

The upper ontology has become a common method for ensuring interoperability between ontologies. It is defined as a domain-independent ontology describing general concepts that can be applied equally to all the domains [18–20]. However, these upper ontologies, such as the basic formal ontology (BFO) [18], DOLCE [19], and SUMO [20], often fail to provide semantic interoperability in C2 systems due to their very high abstraction level. The high abstraction is a property that upper ontology must have to cover the knowledge of domain ontologies, but the very high abstraction level does not provide the core concepts necessary for integrating heterogeneous domain ontologies. Most of the upper ontologies have these problems, and their structures are continuously improved to solve them.

A variety of core ontologies have been developed to support SA, decision making, joint operation, etc. [21–23]. However, these core ontologies fail to meet the above-mentioned four properties. To overcome these drawbacks, we propose a four-layered framework containing a linked data layer that models the relationships between instances among distinct domain ontologies.

2.2 Sub-ontology extraction

To reduce the burden of reasoning and reuse the ontologies, researchers have proposed extracting sub-ontologies related to the given problems from full-scale ontologies. These studies include the view extraction, query-based, graph-based, and traversal-based methods. Table 1 lists the descriptions and limitations associated with each method.

The traditional methods aim to support ontology engineers who want to build a new single ontology. Extraction criteria are mainly configured based on prior knowledge. In addition, engineers should interpret existing domain ontologies or their knowledge. However, it has limited ability to reflect many semantic relationships

Table 1 Comparison of the sub-ontology extraction methods

	Description	Limitation	References
View-Extraction	Selectively extract the part to be reused in the ontology like RBD's view Predefine classes and relationships to be extracted depends on situation	Intervention of ontology engineers and domain experts is required for the extraction process Unhandled situations that are not defined in advance	[5, 24, 25, 26, 27, 28]
Query-based	Develop the virtual ontologies using query results The goal is a query response, not reuse	Generate virtual ontology module unable to integrate to others	[22, 23, 29, 30]
Graph-based	Calculate the weight of edges through class dependency or set the weight manually Predefine the graph pattern and extract the most similar module	Semantic relations are not considered with the weight using class dependency The burden on the ontology interpretation occurs with the manual weight imposing	[20, 31, 32, 33, 34]
Traversal-based	Select initial class manually Extract all classes, properties, and instances connected with the initial class	Semantic relations are not sufficiently reflected with direct connection Oversized ontology module is extracted with indirect relations Requires manual initial class selection process	[15, 35, 36, 37]

implied in heterogeneous ontologies because they utilized prior knowledge to find them. Furthermore, most relationships are limited to synthetic interoperability.

2.3 Ontology alignment

Ontology alignment methods have been studied for integrating multiple ontologies into a single ontology. Ontology alignment is the process of generating a set of correspondences between entities, such as classes, properties, or instances, from different ontologies. As in a previous study [38], we divide the ontology alignment methods into three—terminological, structural, and external methods. Although matching systems are classified into different categories, they use multiple methods in combination rather than a single method.

Terminological methods perform ontology alignment by comparing terms in strings. The terms can be names, labels, and/or comments of ontology entities. AgreementMaker, which is a general-purpose multi-layer matching system, has been previously proposed [39]. In the first layer, it compares labels or comments of entities using the term frequency-inverse document frequency (TF-IDF) vector and edits the distance and substring for similarity calculation. Anchor-Flood aims to handle large ontologies efficiently by verifying the equality of two terms with WordNet and calculating similarity using the Jaro–Winkler edit distance [35]. However, the relationships between ontologies determined by terminological methods are generic and provide only limited information.

Structural methods align ontologies by exploiting the internal structure of entities comprising their attributes or an external structure comprising relations between entities. In a previous study [31], DSSim has been introduced, which is a system used with a multiagent ontology-mapping framework. It compares graph patterns to generate correspondences between entities. In a previous study [40], ASMOV has been proposed, which focuses on the field of bioinformatics. Unlike other systems, it recognizes sets of correspondences that lead to inconsistencies. It was verified by examining five types of internal and external patterns. Alignment with graph isomorphism [41], co-occurrence patterns [42], and feature-based internal structure similarity [43] have also been suggested. However, because the structural method utilizes the internal structure of each ontology, it is difficult to overcome the differences in the level of information representation among the ontologies.

The extensional methods align ontology classes by analyzing their extensions, such as instances. In previous studies [24, 44], RiMOM and RiMOM-IM have been proposed, both of which are based on the dynamic multi-strategy ontology matching framework. These systems perform alignment using instance matching with different strategies based on conditions like the amount of overlapped information. Smart-Matcher [45] exploits the similarities of instances and values. In addition, InsMT and InsMTL [46, 47] align ontologies by analyzing the descriptive information of instances. However, these methods are effective only when they are applied to a large number of instances, thereby limiting their applications.

To overcome the limitations associated with the existing methods, we used the SKOS—a knowledge base that represents well-defined concepts and hierarchical

relationships among them [48]. We attempted to align the ontologies developed for different domains with different goals by selecting and applying related concepts and relations in the SKOS.

3 Multi-layered SA ontology

3.1 Design principles of multi-layered SA ontology

We identified four properties for the SA from multi-sourced information produced in NCE: largeness, heterogeneity, dynamism, and complexity. The largeness means a large scale of data produced by enormous sensor nodes in NCE, and it is increasing exponentially. It is difficult for the ontology to cover this largeness because of low scalability and strictly defined schema. The heterogeneity refers to the differences among data models and information systems. There are various heterogeneities among them, so it is impossible to resolve all possible heterogeneities by only a few ontologies. The dynamism is the lively changed data, schema, and relations among entities in NCE. It is difficult to maintain up-to-date ontologies for changed data. The complexity refers to the entangled structure of sensor nodes and systems.

To deal with these problems, we combined the domain ontologies, core ontologies, and LOD. The heterogeneity can be solved by utilizing various types of ontologies, such as the top ontology, domain ontologies, and core ontologies, while the largeness, dynamism, and complexity can be solved by combining the ontologies and LOD. The ontologies and LOD are interwoven in a layered structure to create awareness of situations through interaction (subsequently referred to as layered SA ontology). To design the layered SA ontology and LOD, we considered the following properties:

- *Semantic interoperability* to solve the heterogeneity problem that occurred in NCE, the layered SA ontology should include artifacts that can be referred to as domain-independent concepts and their relationships, which can be used to develop domain ontologies under the same axiom. The layered SA ontology adopts an upper ontology to achieve semantic interoperability among different domain ontologies.
- *Completeness of knowledge representation* the multi-layered SA ontology should be able to accurately and completely represent all domain knowledge related to SA. To this end, the layered SA ontology allows the use of multiple domain ontologies that can represent only specific domain knowledge or data model.
- *Composability to support dynamic situation awareness* composability is the ability to select and combine only the domain ontologies, concepts, relations, and linked data that are necessary to be aware of dynamic situations. This is a paramount factor in correct decision making as it integrates the domain ontologies and LOD for dynamic situations.
- *Discoverability of multi-sourced information* to be aware of situations successfully in NCEs, it is necessary to establish a method for handling a large amount of multi-sourced data appropriately and efficiently. For this purpose, the multi-

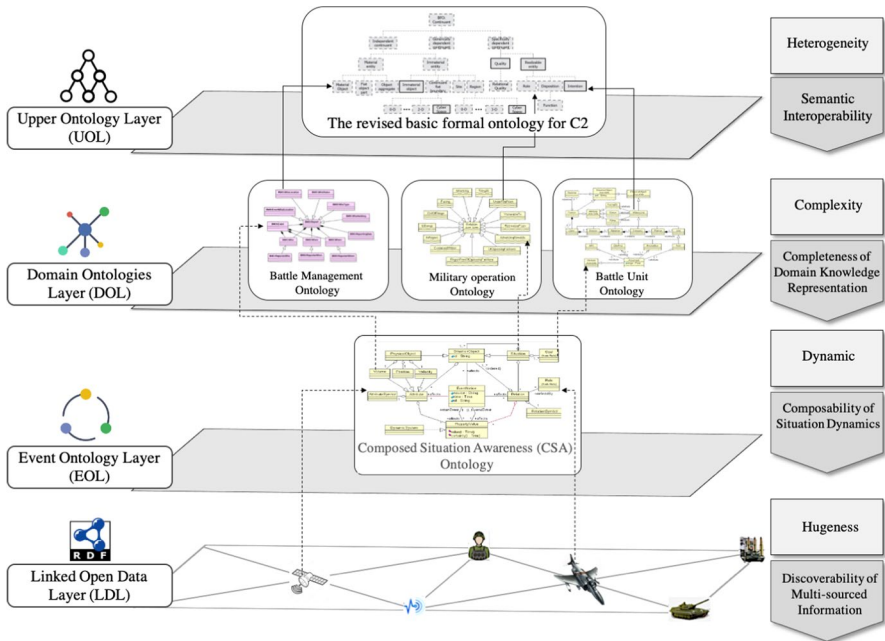


Fig. 1 Multi-layered SAW ontology structure

layered SA ontology enables the retrieval and collection of the data and information generated from multiple sources through the resource description framework (RDF)-based data link.

Based on the four above-mentioned properties, we designed a multi-layered SA ontology structure. As shown in Fig. 1, the structure comprises four layers, namely the upper ontology layer (UOL), the domain ontology layer (DOL), the event-specific ontology layer (EOL), and the linked open data layer (LDL).

The upper ontology defines a domain-independent ontology schema that can be referred to when constructing domain ontologies. The domain ontology defines domain-specific knowledge, such as “Joint Doctrine” or “Army Task,” based on the upper ontology schema. The event-specific ontology is a dynamically generated ontology that depends on the situation through an interaction among the domain ontologies and linked data. Finally, the linked open data are a dataset using RDF, which enables the access, acquisition, and integration of resources across various domains.

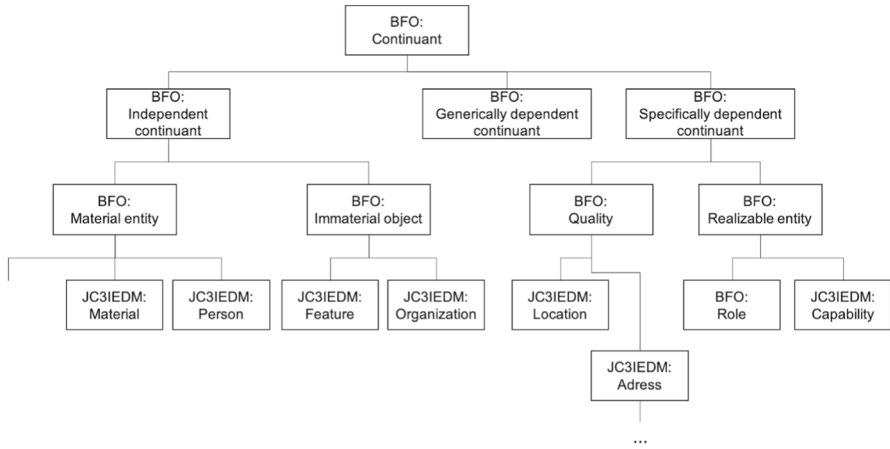


Fig. 2 The modified BFO structure using JC3IEDM

Table 2 Class hierarchy and abbreviations of modified BFO

Hierarchy of class “Continuant”	Abb	Hierarchy of class “Continuant”	Abb
Continuant	C c	Continuant	
Independent Continuant		Generically Dependent Continu	
Material Entity		Specifically Dependent Continant	
Material Object		Quality	SD _C
Fiat Object Part		Relational Quality	Q q
Object Aggregate	IE	Realizable Entity	RE re
Immaterial Entity	IO io	Role	
Immaterial Object	FB _C	Disposition	
Continuant Fiat Boundary	PB	Function	
Physical Boundary (0D to 2D)	CB	Intention	IN
Cyber Boundary	S		
Site	R		
Region	PR		
Physical Region (0D to 3D)	CR		
Cyber Region			

3.2 Detailed descriptions of the layers

3.2.1 3.2.1. Upper ontology layer

To address domain-specific problems, domain ontologies, such as combat

Table 3 Detailed description of added classes

	Description	DL-based expression
<i>Added</i>		
Immaterial object (IO)	Concepts such as “plan” or “command” that exist independently but do not have a physical form	$IO := \exists \text{SubClass Of} \cdot IE \cap \neg \text{Instance Of}^{-1} \cdot (\text{instance Of} \cdot S \cup \text{instance Of} \cdot FB_c \cup \text{instance Of} \cdot R)$
Cyber boundary (CB)	Boundary of object in cyber space, not physical space	$CB := \exists \text{SubClass Of} \cdot FB_c \cap \neg \text{Instance Of}^{-1} \cdot (\text{instance Of} \cdot PB)$
Cyber region (CR)	The region occupied by objects in the cyber space not physical space	$CR := \exists \text{SubClass Of} \cdot R \cap \neg \text{Instance Of}^{-1} \cdot (\text{instance Of} \cdot PR)$
Intention (IN)	The intent of a material entities or immaterial entities to perform the specific tasks	$IN := \exists \text{SubClass Of} \cdot SD_c \cap \neg \text{Instance Of}^{-1} \cdot (\text{instance Of} \cdot Q \cup \text{instance Of} \cdot RE)$
<i>Modified</i>		
Quality (Q)	Expand the scope of application of “Quality” and “Realizable entity” that express unique attributes, roles and functions of material entity as immaterial objects	Relation $1. IO \text{ has_quality } Q \Rightarrow \exists q, io (io \text{ rdfs:type } IO \wedge q \text{ rdfs:type } Q \wedge io \text{ has_quality } q)$

management and operational scenarios, have already been developed for the military. Therefore, it is difficult to ensure the semantic interoperability among the domain ontologies. The upper ontology (or top ontology) is a widely employed method for ensuring their semantic interoperability. It is defined as a domain-independent ontology describing general concepts that can be applied equally to all the domains. A majority of the upper ontologies, such as BFO, OpenCyc, and DOLCE, focus on modeling basic and common concepts, such as “object” and “process.”

To reduce the development burden of the upper ontology, the existing upper ontologies have been analyzed. As a result, by extending the BFO, an upper ontology that can represent the hierarchy among objects (e.g., units) and express time and space (related to operations) was developed. By analyzing the BFO, we determined that it specialized in expressing scientific facts and was very highly abstractive. Simultaneously, we discovered that the BFO could not represent essential concepts among domain ontologies due to the very high abstraction level of the BFO. The very high abstraction level is a key factor for the top ontology, but it prevents essential concepts from forming sturdy relations among domain ontologies. To avoid this problem, we utilize the JC3IEDM, which is the proposed data model for interoperable communication among C2 systems [49]. While the BFO focused on presenting generalized concepts and their hierarchical structures that everyone could relate to, the JC3IEDM focused on interoperability for easily exchanging information among heterogeneous systems. Figure 2 shows the modified BFO structure using the JC3IEDM.

In addition to utilizing the JC3IEDM, we modified some entities in the BFO to improve utility and explicit representation of the upper ontology in NCE. Table 2 lists the classes and their hierarchy in the modified BFO, where upper- and lower-case letters denote classes and instances, respectively.

Table 3 lists detailed descriptions of the new and modified classes and the description logic (DL)-based expressions. Because Q and RE are “specifically dependent continuants” applicable to existing specific entities, it is necessary to extend or modify the existing relations. Table 3 also lists the modified relations for Q and RE.

3.2.2 Domain ontology layer

A domain ontology is a specified conceptualization comprising formal description of concepts, relations between concepts, and axioms on the relations in the domain of interest [12]. Domain ontologies provide a clear understanding of the concepts and their relations and knowledge of the domains. Many domain ontologies, such as battleships, weapons, and terrain, have already been developed. In addition, for accurate decision making, various ontologies are required, such as situation ontologies defining temporal information, geospatial information, and weather information; ontologies describing units and capabilities of each organization; and ontologies for military operations. Therefore, a domain ontology can represent domain knowledge necessary for precise SA in a clear and complete form.

However, because domain ontologies are developed by considering the characteristics and objectives of each domain, heterogeneity necessarily arises among them. In our framework, we assumed that all the domain ontologies are developed based on the structure of the upper ontology developed in UOL. The d^{th} domain ontology (DO_d) in the pool of the domain ontologies is represented as a 4-tuple as follows:

$$DO_d = (Cl_d, In_d, P_d, R_d) \quad (1)$$

where Cl_d is a set of classes UC and DC_d . Here, UC is the set of the upper ontology classes, and DC_d is the set of classes of DO_d . We use the expression “uc:dc” to denote the reference relation between the element dc_d of DC_d and the element uc of UC , and it is represented in a DL-based expression as follows: $\forall dc(dc_d \in DC_d) \Rightarrow \exists uc(uc \in UC \wedge uc : dc_d)$. In_d and P_d represent the set of instances and properties of the DO_d , respectively. Finally, R_d is the set of relations between specific classes, instances, and properties. It is represented as follows:

$$R_d = \{(s, r, o) | s \in Cl_d \vee In_d, r \in P_d, o \in Cl_d \vee In_d \vee DV\} \quad (2)$$

where s , r , and o are the subject, relation, and object of the relation pattern, respectively.

3.2.3 EOL

The EOL is a key layer that combines necessary domain ontologies and related linked data according to a specific event. If necessary, the event-specific ontology can directly refer to the upper and domain ontologies to support semantic interoperability. From our perspectives, the integrated SA ontology is located in the EOL because it obtains the necessary domain knowledge from the domain ontologies in the DOL to take notice of the given situations and collect necessary data and information from the LDL. The integrated SA ontology collects appropriate domain knowledge, data, and information for a given situation in the DOL and LDL and combines them to perform SA for flexible and precise decision making.

3.2.4 LDL

In the LDL, individual instances or resources in stored multi-sources are linked to other instances in an RDF triple format even if they are stored in sources in completely different domains. These instances and their links serve as a bridge among domain ontologies to support the SA ontology when it should combine them. The link information between the instances makes the inference of the relationships among heterogeneous domains possible. For example, we consider a scenario where enemy battleships are found during maritime reconnaissance. Then, friendly battleships must be found that can make an effective attack immediately. To this end, from the weapons system database, the C2 system first identifies the effective range and target information of the weapons system that the

Table 4 Notations and definitions used in this paper

Notations	Definition
i, j, k, l	Index of terms, matched RDFs, entities, and DOs
T^0	A set of initial terms that are identified by the preprocessing module, $T^0 = \{t_1, t_2, \dots, t_i, \dots, t_l\}$, where t_i is the i^{th} term in literal form from the event
RDF_i^m	A set of matched RDFs obtained by keyword matching t_i , $RDF_i^m = \{rdf_{i1}, rdf_{i2}, \dots, rdf_{ij}, \dots, rdf_{iR}\}$, where rdf_{ij} is the j^{th} obtained RDF and $rdf_{ij} = \langle s, p, o \rangle$ is the triple pattern of the subject (s), predicates (p), and object (o)
E_i	A set of found entities matched t_i , $E_i = \{e_{i1}, e_{i2}, \dots, e_{ik}, \dots, e_{iE}\}$, where e_{ik} is the k^{th} found entity by keyword matching t_i
pc_{ip}	p^{th} projected class with E_i on ADO_i , $pc_{ip} \in ADO_i$
CDO_i	A set of discovered domain ontologies using E_i , $CDO_i = \{CDO_{i1}, CDO_{i2}, \dots, CDO_{iJ}, \dots, CDO_{iD}\}$, where CDO_{iJ} is the J^{th} discovered domain ontology. CDO_{iJ} is represented as a 4-tuple similar to the domain ontology
ADO_i	A selected appropriate domain ontology with $E_i, ADO_i \in CDO_i$
SDO_i	An extracted sub-ontology from ADO_i , SDO_i is represented as a 4-tuple similar to the domain ontology

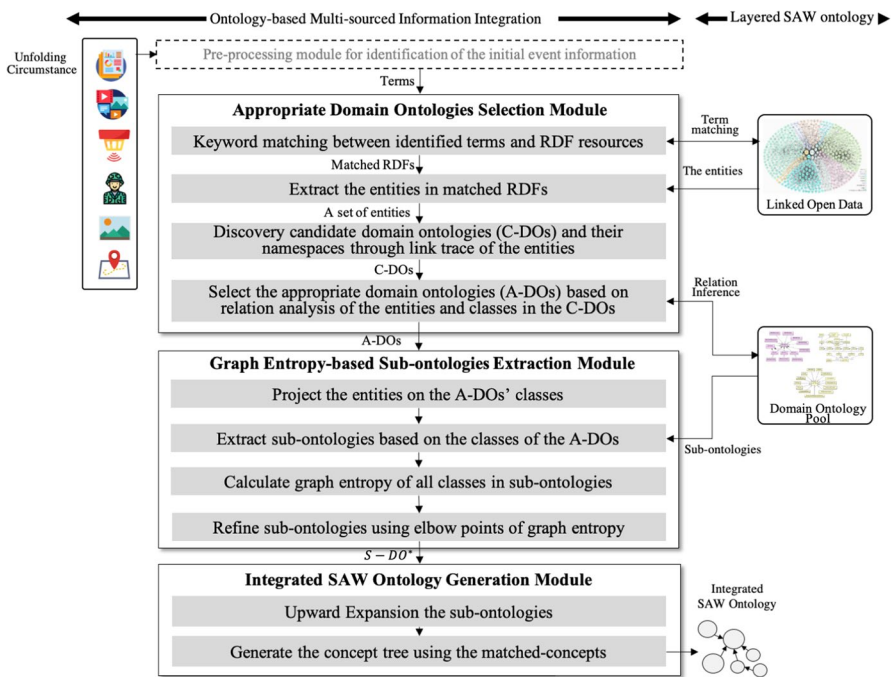


Fig. 3 Multi-sourced information integration procedure via interactions of multi-layered SAW ontology

battleships are equipped with. Information about the battleships is stored in the battleship database, and the types of the enemy battleships are identified through the reported enemy database. Finally, the geolocation of the friendly battleships and the performance of the weapons system with which they are equipped are considered to identify the friendly battleships that should perform the mission.

This example indicates that the linked data make it possible to discover large multi-sourced data and information and collect them employing a domain-independent strategy. The linked data also provide clues to the SA ontology that can integrate domain ontologies and data. We assume that an instance of the linked data is associated with at least one domain ontology so that the SA ontology can utilize the linked data to identify and integrate the domain ontologies.

4 Multi-sourced information integration framework for SA

Table 4 lists the definition of notations used in this paper.

4.1 Overall architecture

Using the layered SA ontology proposed in the previous section, we developed a framework that can acquire and integrate the appropriate domain knowledge and information required for a specific SA. Figure 3 shows the overall framework. The framework consists of four modules, namely the preprocessing, appropriate domain ontology selection, graph entropy-based sub-ontologies extraction, and integrated SA ontology generation modules. However, the preprocessing module is beyond the scope of this research, and we do not discuss it further. We assume that the preprocessing of information on events happening under the unfolding circumstances has already been performed using WordNet and/or the Semantic Sensor Network Ontology. As a result, the preprocessing module generates a set of refined terms related to the events under the unfolding circumstances as its output.

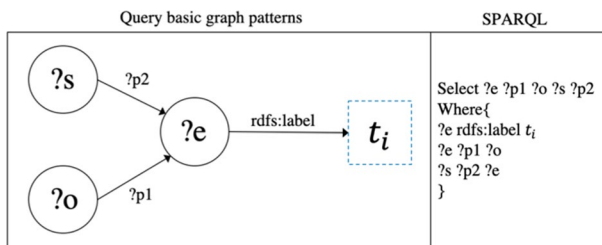


Fig. 4 Keyword matching considered RDF triple patterns and SPARQL

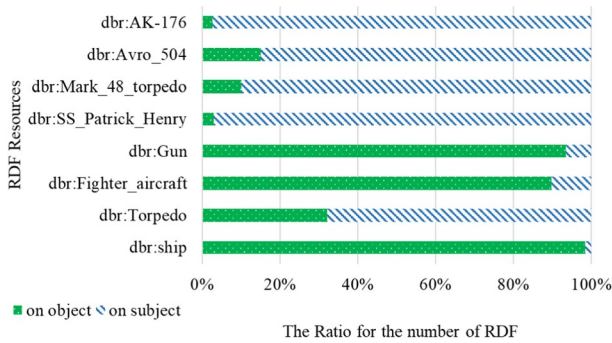


Fig. 5 Appearance frequency of the RDF resources

4.2 Appropriate domain ontologies selection module

This module is executed to discover and select appropriate domain ontologies (ADOs) from the DOL. The ADO is defined as an ontology that includes domain knowledge best suited to process current events using the initial terms. The LDL efficiently exploits a number of the DOs existing in the DOL using its own properties and discoverability. The process of finding ADOs using initial terms is as follows:

- Keyword matching between identified terms and RDF resources

In this step, RDF resources associated with the initial terms (T^0) are obtained through keyword matching. Keyword matching is performed between the t_i and the RDF value, which represents the RDF resources linked by the predicate “rdfs:label.” As shown in Fig. 4, the RDF resources are represented by RDF URIs and their literal data are expressed by predicates such as “rdfs:label.” For this reason, we selected a keyword matching target with t_i as the RDF value in the RDF triple pattern. As a result, we obtained a matched RDF resource (RDF_i^m). RDF_i^m is defined as a set of

RDF_i^m : a set of matched RDFs obtained by keyword matching the t_i
 doe: List of calculated degree of entity ($doe(rdf_{ij})$)
 Med: a median value of all calculated $doe(rdf_{ij})$

```

doe = []
For all i,j:
  Doe ← Calculate  $doe(rdf_{ij})$ 
Med = Median(Doe)
For all i:
  For all j:
    If  $doe(rdf_{ij}) > Med$ :
       $e_{ik} = rdf_{ij}$ 
       $E_i \leftarrow e_{ik}$ 
Return  $E_i$ 

```

Fig. 6 Algorithm of entities extraction

```

 $E_i$ : A set of found entities matched the  $t_i$ 
hops: Number of nodes to perform link tracking
NP: List of discovered namespaces

Link_Tracing_Function( $e, hops$ )
Iteration = 0
NP = []
While Iteration  $\leq$   $hops$ :
  Query = "Select ?o Where {" + str( $e$ ) + "?p ?o}"
  Request Query
  URIs  $\leftarrow$  Query.results()
  For url in URIs:
    If Ontology_Validation_Function(url) = True:
      NP  $\leftarrow$  Query.results(url)
  Return NP
End Function

Ontology_Validation_Function( $url$ )
If '#' in url:
  Text = Access(url)
  If <owl:Ontology> and <owl:Class> in Text:
    Return True
Else:
  Return False
Else:
  Return False
End Function

Begin Discovery CDOs and their namespaces
DOs_NP = []
For all i:
  For all k:
    DOs_NP  $\leftarrow$  Link_Tracing_Function( $e_{ik}, hops = 2$ )
Return DOs_NP

```

Fig. 7 The algorithm of discovery CDOs and their namespaces

RDFs, whose subjects or objects are an RDF resource identified by keyword matching with t_i . Figure 4 shows the keyword matching process.

•Entity Extraction from Matched RDFs

This is the process of extracting entities (E_i) from the RDF_i^m . Here, entities represent physical or nonphysical individual things existing in the real world. For example, among RDF resources, "dbr:Ship" is not an entity because it expresses the concept of certain things, while "dbr:SS_Patrick_Henry" is an entity because it is an actual cargo ship. To identify the entities from RDF_i^m , we explored the properties of the entities from the viewpoint of the subject and object in the triple patterns; Fig. 5 shows the results.

As shown in Fig. 5, most entities, such as dbr:AK-176 and dbr:Avro_504, corresponding to individual things are used as objects for the RDF triple, while most subjects, such as dbr:Gun and dbr:Fighter_aircraft, are used as nonentities, i.e., concepts

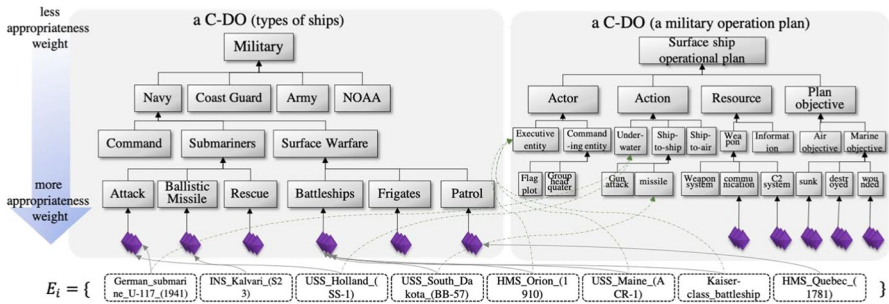


Fig. 8 Appropriateness of the relationship between entities and domain ontologies

of things. Based on the properties of the entities, we devised a metric that can measure whether a specific RDF resource (rdf_{ij}) is an entity. The metric is as follows:

$$doe(rdf_{ij}) = \frac{1}{\log(\text{num}(rdf_{ij}^o))} \times \frac{1}{\text{len}(RDF_i^m)} \times \sum_{j=1}^R \text{num}(rdf_{ij}^s) \quad (3)$$

where $\text{num}(rdf_{ij}^o)$ is the number of RDF with rdf_{ij} on an object, and $\text{num}(rdf_{ij}^s)$ is the number of RDF with rdf_{ij} on a subject. $Doe(rdf_{ij})$ is used to find the element (e_{ik}) of E_i . We show it in Fig. 6.

•Discovery Candidate Domain Ontologies (CDOs) and Their Namespaces

To find CDOs, we first collected the links of all elements of E_i . We assumed that e_{ik} is associated with the classes or instances of at least one DO. The algorithm for obtaining CDOs and their namespaces is shown in Fig. 7. This algorithm is based on the fact that the structure of the class URIs belonging to a specific ontology is combined with the namespace URI of the ontology, symbol “#,” and class name. Using this property, we identified the URI that is the class or instance of the ontology and RDF resources connected with e_{ik} and extracted the namespace URI by parsing it based on the symbol “#.” Finally, to determine whether the extracted namespace URI is an ontology, our framework verifies whether there is at least one <owl: Ontology> and <owl: Class> tag in the address. Once the existence of tags is confirmed, the ontology of the corresponding namespace is classified as CDOs.

•Select the Appropriate Domain Ontologies (ADOs)

To avoid the inefficiency caused by the use of multiple CDOs and prevent the overload of information experienced by users, the ADOs that are most closely related to the current events of CDOs should be selected. The ADOs are selected by link analysis between the elements of E_i and CDOs. An example is provided to illustrate the link analysis method. There are two CDOs and set E_i with seven elements. As shown in Fig. 8, some elements of E_i are linked to the classes or instances of the

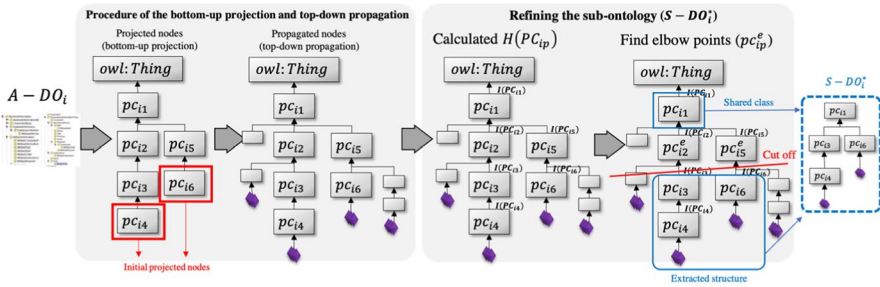


Fig. 9 Sub-ontology extraction process using graph entropy in ADO_i

CDOs, although they vary to some extent. Obviously, a higher number of links with elements lead to a greater ability of the CDO to explain events. In Fig. 8, the CDOs in the left window have a greater number of concepts related to the event than those in the right window. However, if we simply determine ADOs by the number of the links, the difference in information quantitation of the classes or instances in the CDOs may be neglected. To solve this problem, we propose a metric that can calibrate the number of the links with the information entropy.

The weighted frequency-based appropriateness (APR_{il}) of the CDO_{il} is calculated as follows:

$$doAPR_{il} = \sum_{k=1}^E wf_{ikl} \tag{4}$$

where $wf_{ikl} = \sum_m^{all} \frac{Depth(n_{ilk_m})}{Depth(C-DO_{il})}$. Here, $Depth(CDO_{il})$ is the total number of the links from the leaf node to the top node in the CDO_{il} . $Depth(n_{ilk_m})$ is the number of the links from the node n_{ilk_m} to the top concept (owl:things) in the CDO_{il} . n_{ilk_m} is the m^{th} node of the CDO_{il} , which is related to the element e_{ik} of E_i and can be a class or instance of the CDO_{il} .

Among $CDO_{il}(forall)$, we determine ADO_i to be the CDO_{il} with the maximum value APR_{il} .

4.3 Graph entropy-based sub-ontologies extraction module

This module is created to extract sub-ontologies based on graph entropy, which can measure the structural complexity of graphs [36]. We extracted sub-ontologies using the fact that the higher the entropy, the more complicated the graph. The first step while extracting the sub-ontologies was to project all elements of E_i onto the ADO_i . Next, using the initial projected nodes (pc_{ip}^l) on the ADO_i , we performed the bottom-up projection. And then we performed the top-down propagation on the ADO_i to extract SDO_i . The overall procedures of the bottom-up and top-down propagations on the ADO_i are shown in the left window of Fig. 9. Using the sub-ontologies,

which are lightweight ontologies comprising only the data and information closely related to the current event, a rapid and accurate SA becomes possible.

In the third step, for all p , we computed the graph entropy for all sub-graphs with pc_{ip} as the top node. Here, the graph entropy of pc_{ip} represents the structural complexity of the graph, whose top node is pc_{ip} . The graph entropy of pc_{ip} is calculated as follows:

$$E(pc_{ip}) = - \sum_{o=1}^O \frac{|N_{ipo}|}{|V_{ip}|} \log \left(\frac{|N_{ipo}|}{|V_{ip}|} \right) \quad \text{for all } p \quad (5)$$

where $|N_{ipo}|$ is the number of the nodes connected to pc_{ip} by the depth of o , and $|V_{ip}|$ is the total number of the nodes connected to pc_{ip} .

•Refine Sub-Ontologies Using Elbow Points of Graph Entropy

For all p , we determined the elbow node of $SDO_i (pc_{ie})$ using the value of $E(pc_{ip})$. The deviation in the graph entropy $ED(pc_{ip})$ represents the entropy deviation between the two nodes related to $c_{ip} SubClassOf^{-1} pc_{ip}$. Then, pc_{ie} is determined as the node with the maximum $ED(pc_{ip})$ value, and it is defined as follows.

Definition 1 Elbow node of $pc_{ip}^l (pc_{ie})$ pc_{ie} is a single node with the greatest $ED(pc_{ip}^l)$ among the superclasses of pc_{ip}^l . It is represented by a DL-based expression as follows:

$$pc_{ie} := \forall (\exists SubClassOf^{-1} \cdot pc_{ip}^l) \cdot (\exists GED \cdot (\exists SubClassOf^{-1})) \cap \exists SubClassOf^{-1} \cdot pc_{ip}^l \quad (6)$$

where $YGEDZ$ signifies $ED(Y) \geq ED(Z)$, and $SubClassOf^{-1}$ is the inverse of $SubClassOf$.

Next, we removed all the ancestor nodes of pc_{ie} (for all e) to eventually obtain multiple separated SDO_i with pc_{ie} as the top node. SDO_i is defined as follows.

Definition 2 i^{th} Sub-ontology (SDO_i) SDO_i is a subset of ADO_i , where the class is the union of pc_{ie} and its subclasses. A DL-based expression of the SDO_i is as follows:

$$SDO_i \subseteq ADO_i, \text{ where } Cl_i \equiv pc_{ie} \cup SubClassOf.pc_{ie} \quad (7)$$

For all i , it is necessary to integrate multiple SDO_i separated by pc_{ie} to obtain a complete ontology. To integrate multiple SDO_i (for all i), we identified a node shared by all pc_{ie} (for all e) and connected it to all pc_{ie} to obtain a tree-structured complete ontology. The shared node for all pc_{ie} is defined as follows.

Definition 3 Shared node (pc_{is}) pc_{is} is a common superclass of all pc_{ie} that has the greatest depth. It is represented by the following DL-based expression:

E_i : A set of found entities matched the t_i
 I : Information Entropy
 ADO_i : a selected appropriate domain ontology with E_i
 SDO_i : an extracted sub-ontology from ADO_i
 GE: a list of calculated graph entropy

```

Function Sub-ontology_Extractor( $E_i, ADO_i$ )
For  $e_{ik}$  in  $E_i$ :
    PC  $\leftarrow$  Find.RelClass( $e_{ik}, ADO_i$ )
For pc in PC:
    While pc has 'rdfs: subClassOf':
        pc  $\leftarrow$  Find.ClassByProperty(pc, 'rdfs: subClassOf')
         $SDO_i.append(pc)$ 
        PC.append(pc)
For pc in PC:
    While pc has 'rdfs: subClassOf-1' or 'rdf: type-1':
        comp  $\leftarrow$  Find.CompByProperty(pc, 'rdfs: subClassOf-1' or 'rdf: type-1')
         $SDO_i.append(comp)$ 
Return  $SDO_i, PC$ 
End Function
Function Sub-ontology_Refiner( $SDO_i, PC$ )
For  $pc_{ip}$  in PC:
    ie  $\leftarrow$  Calculate( $l(pc_{ip})$ )
    IE.append(ie)
    IED.append(ie - IE[-1])
     $pc_{ie} \leftarrow PC[IED.index(MaxValue)]$ 
For class in  $SDO_i$  :
    If class in Find.ClassByProperty( $pc_{ie}, 'rdfs: subClassOf^{-1}'$ ):
        RemoveAllConnections(class)
Return  $SDO_i, pc_{ie}$ 
End Function
Function Shared-Node_Linkler( $SDO_i, pc_{ie}$ )
For class in  $SDO_i$  :
    If class in Find.ClassByProperty( $\forall pc_{ie}, 'rdfs: subClassOf^{-1}'$ ):
        SC.append(class)
        sn  $\leftarrow$  Find.ClassByProperty(SC,  $\neg'rdfs: subClassOf^{-1}'$ )
         $SDO_i.append(sn)$ 
Return  $SDO_i$ 
End Function
Begin graph Entropy based Sub-ontologies Extraction Process
 $SDO_i = []$ 
For all i:
     $SDO_i, PC =$  Sub_ontology_Extractor( $E_i, ADO_i$ )
     $SDO_i, pc_{ie} =$  Sub-ontology_Refiner( $SDO_i, PC$ )
     $SDO_i^* =$  Shared-Node_Linkler( $SDO_i, pc_{ie}$ )
Return  $SDO_i^*$ 
End Process
    
```

Fig. 10 Refine sub-ontologies using elbow points of graph entropy

$$pc_{is} := \exists SubClass Of^{-1} \cdot \forall pc_{ie} \cap \neg \exists SubClass Of^{-1} \cdot (\exists SubClass Of^{-1} \cdot \forall pc_{ie}) \tag{8}$$

Finally, we obtained the refined ontology of SDO_i , named SDO_i^* , which is defined as follows.

Definition 4 Refined ontology of SDO_i (SDO_i^*) SDO_i^* is a subset of SDO_i , where the class is the union of pc_{ie} , its subclasses, and pc_{is} . The DL-based expression of SDO_i^* is as follows:

$$SDO_i^* \supseteq SDO_i, \text{ where } Cl_i \equiv pc_{ie} \cup SubClass\ Of \cdot pc_{ie} \cup pc_{is} \quad (9)$$

The refinement process of the sub-ontology is shown in the right window of Fig. 9 and summarized as an algorithm in Fig. 10.

4.4 Integrated SA ontology generation module

This module performs the integration of all SDO_i^* (for all i). However, although the sub-ontologies are developed based on the upper ontology in the UOL, they cannot be easily integrated because the domains and granularities of the ontologies differ. To facilitate integration, we used the SKOS core vocabulary structure—a general model focusing on labeling and the hierarchies of the RDF resources. The SKOS-based SDO_i^* integration process is divided into the two following stages.

4.4.1 Stage 1. Upward expansion of the sub-ontologies using “skos:concept” and “skos:broader”

To integrate multiple sub-ontologies, we searched for common superclasses of the shared nodes (pc_{is}), which were included in the sub-ontologies, using the hierarchical relations of the SKOS. We performed it in two substages.

In the first substage, we performed keyword matching between pc_{is} and the RDF value, which represents the RDF resources that are linked by the predicate “rdfs:label.” If the keywords matched, we identified the URI of pc_{is} and used it to explore the SKOS to confirm that pc_{is} is the concept. To identify only RDF resources represented by the SKOS, we used skos:Concept, which is defined as an RDF resource. If we confirm that pc_{is} is the concept, we initiate the second substage.

In the second substage, we identified the hierarchical links (i.e., hypernyms) of pc_{is} using “skos:broader.” of the SKOS. However, garbage hypernyms, which are irrelevant to SDO_i^* and cause computational burdens, can be identified because the SKOS contains general domain-independent concepts. To overcome these problems, we developed a bag of words (BoW_i) using the classes of the domain ontology and their synonyms. We identified their synonyms by referring to a lexical ontology such as WordNet. If multiple hypernyms were identified using BoW_i , the above steps were repeated to find the hierarchical links for all hypernyms except those with the same RDF value as pc_{is} . This process was repeated until there were no more multiple hypernyms with skos:broader. When the process was completed, we obtained the q expanded node en_{iq} of SDO_i^* (for all i, q). We used the expanded nodes to integrate all SDO_i^* into a single complete ontology.

Fig. 11 The generation process of refined *CT*

```

CT: Concept tree

Function top_node_generation(TerminalNodes)
For e in TerminalNodes:
    cot = ‘owl:Thing’
Return cot
Function trimming_nodes(CT, cot, pcis)
For pcis:
    Branchis ← FowardSearchTo(cot)
For cos:
    If cos not in Branchis:
        Remove(CT, cos)
    Retrun CT
Begin the modification of concept tree process
For all s:
    If cos has not any upper class:
        TerminalNodes ← cos
        CT ← top_node_generation(TerminalNodes)
    CT ← trimming_nodes (TerminalNodes)
    
```

4.4.2 Stage 2. Generate the concept tree using expanded nodes

We first generated CTs, which comprise the hypernyms of *SDO_i^{*}* and their relations. A CT is defined as follows.

Definition 5 *CT* A *CT* is composed of the hypernyms of *SDO_i^{*}* and their relations. It is represented as follows:

$$CT = (Co, Ed) \tag{10}$$

where Co is a set of concepts that has *co_s* and *pc_{is}* as elements ($\{co_s, pc_{is}\} \in Co$). *co_s* is the *sth* hypernym of *en_{iq}* that occurs in two or more domain ontologies simultaneously ($co_s \in \cup_{i,q}^{all} en_{iq}$) and *pc_{is}* is a leaf node. Ed is a set of edges among *co_s* and is used to generate a CT as a directed acyclic graph pattern. We excluded the hypernyms of *en_{iq}*, which appear only in one domain ontology, because they do not help in inferring relations with other *SDO_i^{*}*.

To generate the integrated SA ontology, we refined the CT considering two aspects. First, a single top node that guarantees the tree shapes of the CT was required. This requirement was satisfied by adding the most general class, “thing.” Second, the size of the CT was reduced by eliminating concepts that are not related to the integration of *SDO_i^{*}*. To achieve this, we removed all the classes of CT that fell outside the path from the top class “thing” to the bottom class *pc_{is}*. We summarized this process as an algorithm in Fig. 11.

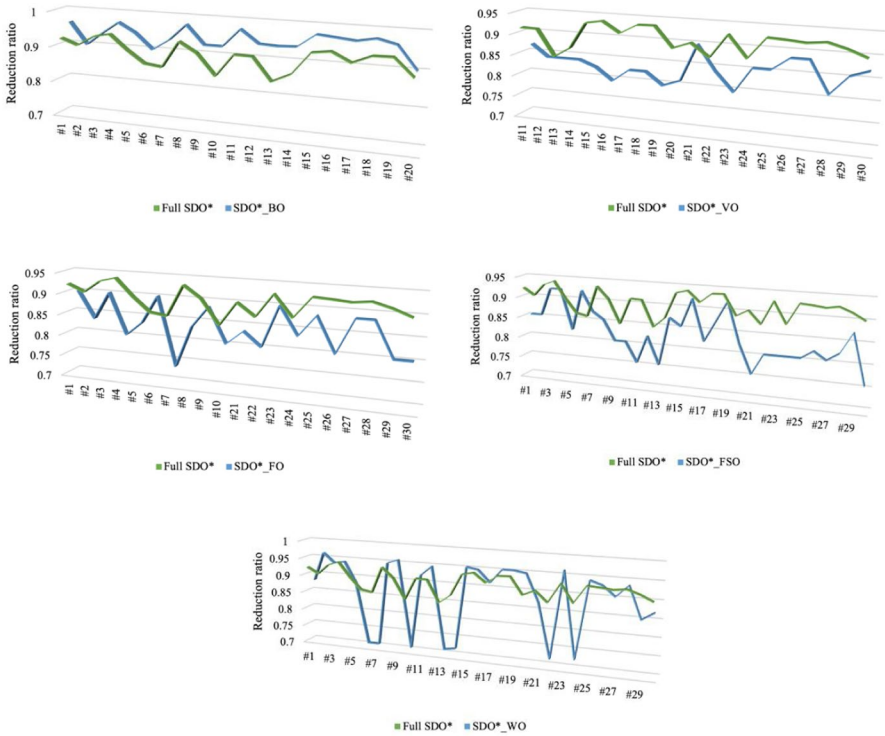


Fig. 12 The reduction ratio of the classes by the scenario

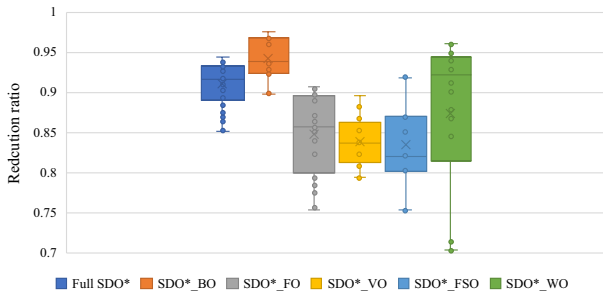


Fig. 13 The box plot of the reduction ratiion by the ontology

5 Illustrative scenario and experiments

We conducted some experiments based on illustrative scenarios to demonstrate the effectiveness of the layered SA ontologies and our framework. In these experiments, we compared the SA results obtained by decision-makers with those by our framework from the viewpoint of accuracy and time. For the experiment, we generated a scenario considering the following tactical environment.

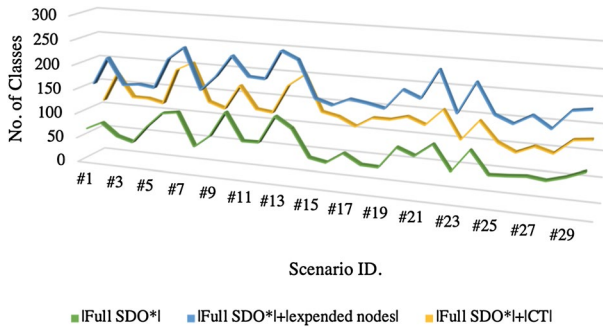


Fig. 14 The comparison results of the number of the classes before SKOS expansion, after SKOS expansion, and refined SKOS expansion

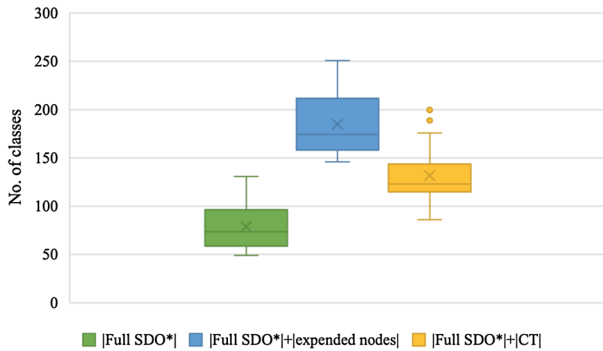


Fig. 15 The box plot of the number of the classes before SKOS expansion, after SKOS expansion, and refined SKOS expansion

5.1 Illustrative scenarios and environmental setup

In a tactical environment, commanders receive a significant amount of heterogeneous information from multiple sources and should make accurate decisions using this information. However, due to the heterogeneity and largeness of the collected information, a significant amount of time and effort is needed to have an accurate SA. For example, a captain can receive data and/or information from other information systems or sensor nodes in a situation. For an experiment, we considered 30 scenarios in which the captain acquired information from various sources. In addition, we developed five domain ontologies—the battleship ontology (BO), operation ontology (OO), vehicle ontology (VO), force structure ontology (FSO), and weapons system ontology (WO). The BO represents the category of battleship in a hierarchical structure, with the available weapons, capabilities, and specifications. The OO includes specifications, requirements, codes, and relationships among the operational commands that the military should perform in specific situations or contexts. The FSO represents the structure of the classes contained in the Army,

Navy, Air Force, and Marine Corps. Finally, the WO indicates the classification of weapons and the requirements for their operation, the ranges, sizes, and associated classes. In these scenarios, SA is achieved by domain knowledge identified using three-domain ontologies and using sub-ontologies, which are integrated in the SKOS knowledge base.

5.2 Experiments and results

We conducted the first experiment to demonstrate the effectiveness of the extraction of sub-ontologies by scenario. We compared the number of the classes in the domain ontologies with that in the sub-ontologies extracted from them. We used the following formula for comparison to calculate the reduction ratio:

$$1 - \frac{|DC_d \in DO_d|}{|DC_d \in SDO_d^*|} \quad (11)$$

Figures 12 and 13 summarize the results of the experiment.

Figure 12 shows that the degree of reduction of the domain ontology differed depending on the scenario, and the average standard deviation was 0.0032825. However, the reduction ratios of the full SDO^* were stable regardless of the scenarios. In addition, as shown in Fig. 13, the deviation of the reduction ratio of the full SDO^* was the second-lowest, following the BO with the average of 0.9111. It indicates that the reduction efficiency was very high. The results of the experiment reveal that SDO^* of appropriate size can be reliably and efficiently extracted from various domain ontologies regardless of the scenarios.

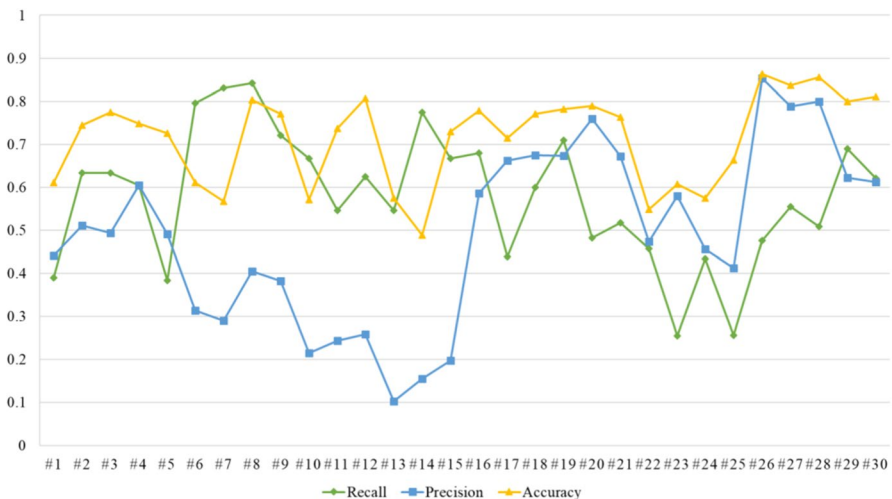


Fig. 16 The comparison results with respect to recall, precision, and accuracy

We conducted the second experiment to demonstrate that the volume of the integrated ontology based on the SKOS was appropriate. To achieve this, we counted and compared the number of the classes of the SDO^* ($ISDO^*$), the number of the classes expanded using the SKOS ($ISDO^*$ +expanded nodes), and the number of the classes of the CT ($ISDO^*$ +|CT|). Figures 14 and 15 show the results of the comparison.

As shown in Fig. 14, from the viewpoint of the number of the classes, the expanded SDO^* based on the SKOS was 2.4 times larger than that of the SDO^* , whereas the refined CT was 1.74 times larger than that of the SDO^* . This signifies that the SKOS-based expansions correctly identify the general concepts of SDO^* classes and simultaneously remove irrelevant concepts at the appropriate level. A qualitative evaluation was performed in the third experiment. In addition, from Fig. 15, there are scenarios in which it is difficult to take notice of the situation using the integrated ontology refined by CT. To avoid side effects due to the outliers of the scenarios, we can use the median rather than the mean of the number of the classes. Therefore, the number of the classes of the refined CT was 1.62 times larger than that of the SDO^* .

In the third experiment, to demonstrate that the integrated ontology was properly generated, we compared the results of the SA inferred with the integrated ontology with those by domain experts. To achieve this, we developed BoW_i , which comprised concepts and terms related to the scenarios. We instructed the domain experts to select the concepts and terms necessary for the scenario-specific SA using the BoW_i . Figure 16 shows the results of the comparison from the viewpoint of recall, precision, and accuracy.

The mean recall and precision were 0.2278 and 0.4909, respectively, while the variances were 0.0232 and 0.042. From Fig. 16, the recall and precision fluctuated greatly depending on the scenarios, and their mean was not high. It means that our model is based on a much larger pool of information and evidence than those used by most domain experts. The information overload due to the use of the integrated ontology can be overcome by creating scenarios that imply data that require reasoning. However, the mean and the variance of the accuracy were 0.7139 and 0.01, respectively. This indicates that the integrated ontology cannot be optimized as much as the information selected by the domain experts; however, it can be interpreted as containing sufficient information for SA.

6 Conclusion and future work

In this paper, we proposed a novel framework for SA using layered SA ontologies that can integrate heterogeneous and multi-sourced information. In this framework, we used domain ontologies to model concepts and relations between concepts and combined a comprehensive range of domain knowledge by reusing the domain ontologies. In addition, to overcome limitations associated with the large and dynamic data and information produced in NCEs, we extended the functionality of the LOD to identify situation-specific domain ontologies, extracted the partial structures from the domain ontologies, and integrated them. In addition, we proposed a framework

that could acquire and integrate the appropriate domain knowledge and information for situation awareness. The framework comprised four modules, namely the preprocessing, appropriate domain ontologies selection, graph entropy-based sub-ontologies extraction, and integrated SA ontology generation modules. Each module contains a very challenging research topic that has never been attempted before and will be improved in future work.

To demonstrate the effectiveness of our framework, we performed three experiments: First, we conducted an experiment to demonstrate the effectiveness of extracting sub-ontologies by scenario. To this end, we compared the number of the classes in the domain ontologies with the number of the classes in the sub-ontologies extracted from them. We conducted the second experiment to demonstrate that the volume of the integrated ontology based on the SKOS was appropriate. Finally, in the third experiment, we compared the results of SA inferred using the integrated ontology with the results of domain experts to determine whether the integrated ontology was accurately generated.

This research can be extended in several directions in the future. First, the full-scale implementation suggested in this paper should be carried out. In addition, we can seek to develop domain ontologies that can be integrated into SA ontologies. Finally, various scenarios, including military operations, C2, and business domains, can be tested to evaluate the performance of the layered SA ontology and the integration framework. This paper is designed for C2 systems but can be applied to other fields. For example, there are similar aspects and problems to C2 systems in other fields such as the Cyber Physical Systems and the Internet of Things. The proposed method can be applied to those fields to improve the quality of services in their environments.

Acknowledgements This research was supported by C2 integrating and interfacing technologies laboratory of Agency for Defense Development (UD180014ED).

References

1. Chmielewski M, Kukiełka M, Frąszczak D, and Bugajewski D (2017) Military and crisis management decision support tools for situation awareness development using sensor data fusion. In: International Conference on Information Systems Architecture and Technology. Springer: Cham, pp 189–199
2. Fenza G, Furno D, Loia V, & Veniero M (2010) Agent-based cognitive approach to airport security situation awareness. In: 2010 International Conference on Complex, Intelligent and Software Intensive Systems. IEEE, pp 1057–1062
3. Maran V, Machado A, Machado GM, Augustin I, de Oliveira JPM (2018) Domain content querying using ontology-based context-awareness in information systems. *Data Knowl Eng* 115:152–173
4. Subramaniaswamy V, Manogaran G, Logesh R, Vijayakumar V, Chilamkurti N, Malathi D, Senthilselvan N (2019) An ontology-driven personalized food recommendation in IoT-based healthcare system. *J Supercomput* 75(6):3184–3216
5. Kokar MM, Matheus CJ, Baclawski K (2009) Ontology-based situation awareness. *Inf Fusion* 10(1):83–98
6. Euzenat J, Shvaiko P (2007) *Ontology matching*, vol 1. Springer: Berlin

7. Albagli S, Ben-Eliyahu-Zohary R, Shimony SE (2012) Markov network based ontology matching. *J Comput Syst Sci* 78(1):105–118
8. Doerr M, Hunter J, and Lagoze C (2003) Towards a core ontology for information integration. *J Digital Inf* 4(1)
9. Doran P, Tamma V, and Iannone L (2007) Ontology module extraction for ontology reuse: an ontology engineering perspective. In: *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*, ACM, pp 61–70
10. Gao M, Chen F, and Wang R (2018) Improving medical ontology based on word embedding. In: *Proceedings of the 2018 6th International Conference on Bioinformatics and Computational Biology*. ACM, pp 121–127
11. Wang K (2015) Research on the theory and methods for similarity calculation of rough formal concept in missing-value context. In: *Proceedings of the 2015 joint international mechanical, electronic and information technology conference*. <https://doi.org/10.2991/jimnet-15.2015.49>
12. Baumgartner N et al. (2010) BeAware!—situation awareness, the ontology-driven way. *Data Knowl Eng* 69(11):1181–1193
13. Kong J, Kim K, Park G, and Sohn M (2018) design of ontology framework for knowledge representation in command and control. In: *6th International Conference on Big Data Applications and Services*, pp 157–164
14. Kabilan V (2007) *Ontology for information systems (04IS) design methodology: conceptualizing, designing and representing domain ontologies* (Doctoral dissertation, KTH)
15. Ra M, Yoo D, No S, Shin J, and Han C (2012) The mixed ontology building methodology using database information. In: *Proceedings of the International Multiconference of Engineers and Computer Scientists (Vol. 1)*
16. Smith B, Miettinen K, and Mandrick W (2009) *The ontology of command and control (C2)*. State Univ Of New York At Buffalo National Center For Ontological Research
17. Smith B, Vizenor L, and Schoening J (2009) Universal core semantic layer. In: *Ontology for the Intelligence Community, Proceedings of the Third OIC Conference*, George Mason University, Fairfax, VA
18. Arp R, Smith B, Spear AD (2015) *Building ontologies with basic formal ontology*. Mit Press: Cambridge
19. Gangemi A, Guarino N, Masolo C, Oltramari A, Schneider L (2002) Sweetening ontologies with DOLCE. In: *International Conference on Knowledge Engineering and Knowledge Management*. Springer, Berlin, Heidelberg, pp 166–181
20. Niles I, and Pease A (2001) Towards a standard upper ontology. In: *Proceedings of the International Conference on Formal Ontology in Information Systems-volume 2001*. ACM, pp 2–9
21. Deitz PH, Michaelis JR, Bray BE, and Kolodny MA (2016) The missions & means framework (MMF) ontology: matching military assets to mission objectives. In: *Proceedings of the 2016 International C2 Research and Technology Symposium (ICCRTS 2016)*, London, UK
22. Matheus CJ, Kokar MM, Baclawski K (2003) A core ontology for situation awareness. In: *Proceedings of the Sixth International Conference on Information Fusion*, vol 1, pp 545–552
23. Morosoff P, Rudnicki R, Bryant J, Farrell R, and Smith B (2015) Joint doctrine ontology: a benchmark for military information systems interoperability
24. Li J, Tang J, Li Y, Luo Q (2009) RiMOM: a dynamic multistrategy ontology alignment framework. *IEEE Trans Knowl Data Eng* 21(8):1218–1232. <https://doi.org/10.1109/tkde.2008.202>
25. Kontchakov R, Wolter F, Zakharyashev M (2010) Logic-based ontology comparison and module extraction, with an application to DL-Lite. *Artif Intell* 174(15):1093–1141
26. Bhatt M, Flahive A, Wouters C, Rahayu W, Taniar D (2006) Move: a distributed framework for materialized ontology view extraction. *Algorithmica* 45(3):457–481
27. Kontchakov R, Pulina L, Sattler U, Schneider T, Selmer P, Wolter F, Zakharyashev M (2009) Minimal module extraction from DL-Lite ontologies using QBF solvers. *IJCAI* 9:836–841
28. Wouters C, Rajagopalapillai R, Dillon TS, and Rahayu W (2006) Ontology extraction using views for semantic web. In: *Web Semantics and Ontology*. IGI Global, pp 1–40
29. Lozano J, Carbonera J, Abel M, and Pimenta M (2014) Ontology view extraction: an approach based on ontological meta-properties. In: *2014 IEEE 26th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, pp 122–129
30. Doran P, Palmisano I, and Tamma VA (2008) SOMET: algorithm and tool for SPARQL based ontology module extraction. *WoMO*, 348
31. Nagy M, Vargas-Vera M, and Motta E (2007) Dssim-managing uncertainty on the semantic web

32. Noy NF, and Musen MA (2004) Specifying ontology views by traversal. In: International Semantic Web Conference. Springer: Berlin, Heidelberg, pp 713–725
33. d'Aquin M, Doran P, Motta E, and Tamma VA (2007) Towards a parametric ontology modularization framework based on graph transformation. *WoMO*, 315
34. d'Aquin M, Schlicht A, Stuckenschmidt H, and Sabou M (2007) Ontology modularization for knowledge selection: experiments and evaluations. In: International Conference on Database and Expert Systems Applications. Springer: Berlin, Heidelberg, pp 874–883
35. Seddiqui MH, Aono M (2009) An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. *Web Semant Sci Serv Agents World Wide Web* 7(4):344–356
36. Rashevsky N (1955) Life, information theory, and topology. *Bull Math Biophys* 17(3):229–235
37. Flahive A, Taniar D, Rahayu W (2013) Ontology as a Service (OaaS): a case for sub-ontology merging on the cloud. *J Supercomput* 65(1):185–216
38. Ardjani F, Bouchiha D, Malki M (2015) Ontology-alignment techniques: survey and analysis. *Int J Mod Edu Comput Sci* 7(11):67
39. Cruz IF, Antonelli FP, Stroe C (2009) Agreement maker: efficient matching for large real-world schemas and ontologies. *Proceed VLDB Endow* 2(2):1586–1589
40. Jean-Mary YR, Shironoshita EP, Kabuka MR (2009) Ontology matching with semantic verification. *Web Semant Sci Serv Agents World Wide Web* 7(3):235–251
41. Corrales JC, Grigori D, Bouzeghoub M, and Burbano JE (2008, March). Bematch: a platform for matchmaking service behavior models. In: Proceedings of the 11th International Conference on Extending Database Technology: Advances in Database Technology. ACM, pp 695–699
42. Su W, Wang J, and Lochoovsky F (2006). Holistic schema matching for web query interfaces. In: International Conference on Extending Database Technology. Springer, Berlin, Heidelberg, pp 77–94
43. Spiliopoulos V, Vouros G, Karkaletsis V (2010) On the discovery of subsumption relations for the alignment of ontologies. *SSRN Electron J*. <https://doi.org/10.2139/ssrn.3199467>
44. Shao C, Hu LM, Li JZ, Wang ZC, Chung T, Xia JB (2016) RiMOM-IM: a novel iterative framework for instance matching. *J Comput Sci Technol* 31(1):185–197
45. Wimmer M, Seidl M, Brosch P, Kargl H, and Kappel G (2009) On realizing a framework for self-tuning mappings. In: International Conference on Objects, Components, Models and Patterns. Springer, Berlin, Heidelberg, pp 1–16
46. Khiat A, and Benaissa M (2014) AOT/AOTL results for OAEL 2014. In: OM, pp 113–119
47. Khiat A, and Benaissa M (2014) InsMT/InsMTL results for OAEL 2014 instance matching. In: OM, pp 120–125
48. Miles A, Matthews B, Wilson M, and Brickley D (2005) SKOS core: simple knowledge organisation for the web. In: International Conference on Dublin Core and Metadata Applications, pp 3–10
49. de Souza LCC, and Pinheiro WA (2015) An approach to data correlation using JC3IEDM model. In: MILCOM 2015–2015 IEEE Military Communications Conference. IEEE, pp 1099–1102

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.