

# A semantic-centered cloud control framework for autonomous unmanned system

PANG Weijian<sup>1,2</sup>, LI Hui<sup>1</sup>, MA Xinyi<sup>1,3,\*</sup>, and ZHANG Hailin<sup>1</sup>

1. Academy of Military Sciences, Beijing 100091, China; 2. Beijing Aeronautical Engineering Technology Research Center, Beijing 100076, China; 3. Air Forces Command College, Beijing 100097, China

**Abstract:** Rich semantic information in natural language increases team efficiency in human collaboration, reduces dependence on high precision data information, and improves adaptability to dynamic environment. We propose a semantic centered cloud control framework for cooperative multi-unmanned ground vehicle (UGV) system. Firstly, semantic modeling of task and environment is implemented by ontology to build a unified conceptual architecture, and secondly, a scene semantic information extraction method combining deep learning and semantic web rule language (SWRL) rules is used to realize the scene understanding and task-level cloud task cooperation. Finally, simulation results show that the framework is a feasible way to enable autonomous unmanned systems to conduct cooperative tasks.

**Keywords:** scene understanding, cloud control, ontology, autonomous cooperation.

**DOI:** [10.23919/JSEE.2022.000077](https://doi.org/10.23919/JSEE.2022.000077)

## 1. Introduction

Semantic information is much more used than precise information in human collaboration. For example, when someone is asked to “put the package on the first desk to the left of the classroom door”, a semantic-centered inference process is carried out. Relying on semantic information such as <target location,on,desk>, <desk,leftOf,door>, <desk,firstOf,desks>, <classroom,has,desks>, <classroom,has, door>, the target location can be identified easily and the “package delivery task” can be completed without accurate spatial information. Semantic is an important element in intelligence. Knowledge with semantic information enables autonomous unmanned systems to be context-aware, spatial-time aware, and behavior aware. Knowledge graph is considered as one of the keys to intelligent decision-making [1]. Hu et al. [2] considered

that effective utilization of expert knowledge is a key issue for artificial intelligence transferring from the game world to the real battlefield. Defence Advanced Research Projects Agency (DARPA) also proposed a theoretical third-generation artificial intelligence framework that integrates data and knowledge in 2018. And Tang [3] summarized this theory as cognitive graph, which is knowledge graph plus cognitive inference plus logical expression.

Like human beings, autonomous unmanned systems also need the ability to express, efficiently access, and share semantics when they are performing a certain task so that autonomous unmanned systems could collaborate across platforms, systems, or even domains. Semantic information is used to make unmanned systems, such as robots [4], unmanned submarine vehicles [5], self-driving cars [6] and others understand scenes and complete tasks autonomously. However, these researches focus on intelligent control of a single platform. It can be more challengeable when the unmanned vehicles work collaboratively.

Therefore, we propose a semantic-centered cloud control framework for autonomous unmanned systems to realize unified cognition, knowledge sharing, and autonomous cooperation among unmanned vehicles connected to a shared cloud.

The rest of our paper is organized as follows. Section 2 introduces the related works including cloud robotics and knowledge representation of unmanned systems. Section 3 focuses on the design of ontology and the realization of inference engine. Section 4 provides the mechanism of mining deep scene information and triggering autonomous task coordinating. Section 5 simulates and verifies the proposed framework. Finally, Section 6 summarizes the paper.

## 2. Related works and proposed framework

### 2.1 Cloud robotics

Cloud robotics is known as the research field that inte-

Manuscript received February 14, 2022.

This work was supported by the National Defense Science and Technology Innovation Zone of China (193-A13-203-01-01) and the Military Science Postgraduate Project of PLA (JY2020B006).

grates cloud computation and autonomous robotic systems [7]. On one hand, cloud robotics has a centralized cloud that provides massive environment information acquired by robotics connected to the cloud. On the other hand, the cloud provides a shared library of capabilities, behaviors, states of robotics, and even the experiences of other robotics. Cloud robotics is developed from networked robotics and is in line with the trend of multi-agent systems that provide intelligent services. The concept of cloud robotics brings about important applications in areas such as smart city [8], semantic sensor network [9], cloud manufacture [10], and cyber-physic system [11].

However, existing cloud control frameworks focus on data sharing [12], and the control mechanism is tightly coupled with the hardware connected to the cloud, they are more similar to shared databases rather than cloud control systems. In this paper, we focus on building a common and shared cognitive cloud for autonomous unmanned systems. That is the unified representation of knowledge, causal reasoning based on knowledge, and their implementation in task coordination among unmanned vehicles connected to the cloud during task execution.

## 2.2 Knowledge representation based on ontology

Ontology is a formal and normative representation of concepts. The clarity and normative characteristics of ontology make it suitable for knowledge sharing and reuse [13]. Ontology-based way of autonomy has gained attention in recent years. Miguelanez et al. [14] first introduced semantic knowledge information into mission planning, which improved the autonomy level of autonomous underwater unmanned vehicles. Ekvall et al. and Kragic [15,16] proposed a task-level planning system based on experiential activity schema, which realized task planning by configuring task goals and constraints. Mokhtari et al. [17] proposed a conceptual method for autonomous robots that conceptualized task experience into the activity mode and applied it to task planning. Chen [18] proposed a robot task planning method that integrates activity schema with conceptual task experience based on ontology knowledge. Using ontology as a scene modeling method has the advantages of strong expression ability, supporting cognitive inference, high scalability, and loose coupling with hardware. However, most of the existing researches focus on single unmanned vehicle control and their interactivity with human.

In this paper, we focus on ontology's ability in supporting autonomous task coordination of multi-unmanned ground vehicles (UGV) environment. We define unified concepts, relationships, and terminology, provide a unified knowledge representation for the cloud-based unmanned system, and realize the sharing at the knowledge

level. The ontology-based semantic knowledge base (KB) as the carrier of the unmanned system's long-term and short-term episodic memory makes it possible to infer based on task knowledge to support the unmanned system planning autonomously.

## 2.3 Semantic centered cloud control framework for swarm task coordinating

In this paper, we take heterogeneous unmanned system which includes carriers, fork lifters, and wreckers undertaking delivery tasks as an example to verify our framework. In this scenario, carriers need lift coordinating when loading package and may encounter roadblock or maintenance when delivering the package. Therefore, this is a collaborative task scenario in which unmanned vehicles need to understand the scene and undertake collaborative tasks autonomously.

In the framework, a central knowledge engine serves as a cloud, and the unmanned vehicles connected to the knowledge engine serve as nodes that may access the engine at any time. The knowledge engine and all nodes connected to it constitute the cloud control system. All data acquired by sensors is organized under a unified ontology so that all entities in the environment and all details about the collaborative delivery task are described as a semantic network. Then, the rule-based inference engines extract deep scene information to understand the situation they are in and take a suitable collaborative task. Finally, each node undertakes collaborative tasks under the plan domain describe language (PDDL) [19] planner autonomously. The framework is described as Fig. 1. UXV means unmanned X vehicle.

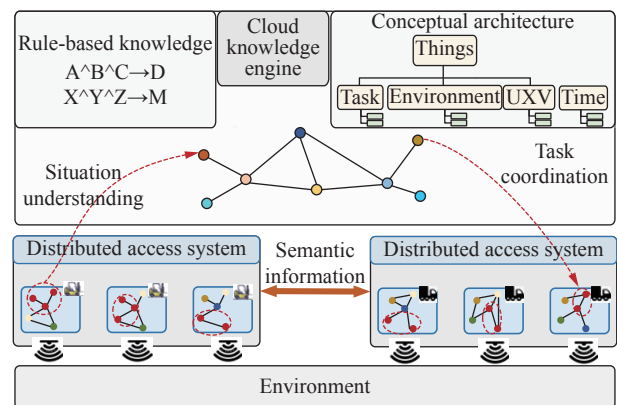


Fig. 1 Semantic-centered cloud control framework

## 3. Knowledge engine for cloud control framework

To implement the proposed framework, the following work must be done: firstly, an ontology must be defined to store information acquired by sensor under a unified conceptual architecture as the common sense of swarmed

unmanned system; secondly, a mechanism of mining deep information of sensor data is needed so that the unmanned vehicles may be aware of the situation they are in and perform tasks autonomously. Ontology is used in this framework to model domain knowledge such as expert knowledge and environmental knowledge acquired by the unmanned system itself.

### 3.1 Building ontology for task scenario

The formal definition of ontology is

$$O = (C, P, A, I) \quad (1)$$

where  $C$  represents a collection of classes and  $P$  represents a collection of predicates that describe object property or data property of a certain instance.  $A$  represents a collection of axioms, describing the relationships between classes, properties, and instances related relationships, such as inclusion, equivalence, reciprocity, transitivity, and symmetry.  $I$  represent a collection of instances, which is an

instantiation of classes. Environment, mission, and UXV are three core ontologies of cloud control ontology. As shown in Fig. 2, ontology mainly expresses knowledge from the perspective of task planning. Among them, environment ontology defines environment-related knowledge, such as artificial objects, natural objects, terrain, and weather; mission ontology defines task-related domain knowledge, such as tasks, behaviors, goals, states, and plans; UXV ontology defines unmanned vehicle related software and hardware knowledge, such as different types of unmanned vehicles, loads, capabilities, physical and geometric parameters, as well as software modules, algorithms. Besides, the cloud control ontology also includes other important ontologies such as universal instruction ontology which defines cooperative task instructions between unmanned systems, communication and network ontology which defines communication knowledge, space-time concept ontology, and concept and common sense ontology.

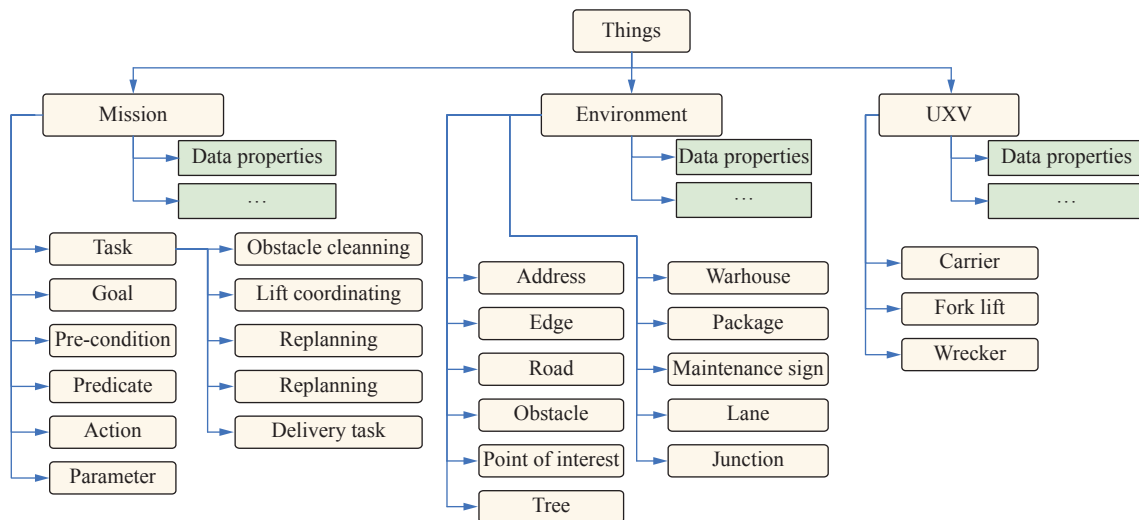


Fig. 2 Task ontology

In addition, the ontology includes the rule that expresses knowledge from a logical perspective. The rule is a kind of causal knowledge expression method. For example, the capability attributes of a certain vehicle in UXV ontology correspond to an atom action in the task ontology, these two ontologies are connected through causal knowledge in the form of rules.

Protege [20] is used as an ontology modeling tool. It provides a graphical modeling interface, and the ontology can be formalized and stored using the web ontology language (OWL) based on the resource description framework (RDF). Good adaptability to the web environment of OWL makes it much more suitable for building an ontology for the cloud control system.

### 3.2 Semantic representation of task domain knowledge

Ontology defines a framework that describes the entities involved in the task scenario. However, these pieces of information are static and conceptual, thus two kinds of predicates are used to describe the semantic relationship needed for scene understanding and autonomous task coordinating. One kind of predicates describes the relationship between different entities, such as “located-at” which describe the positional relationship like “vehicle located at a certain point”. Some predicates that describe entity relationships are shown in Table 1. The other kind of predicates describes the relationship between entities and their data properties, such as “hasState” which describe the state of a certain entity like “road segment #39 has

state ‘LOCKED’”. Some predicates that describe such data properties are shown in Table 2.

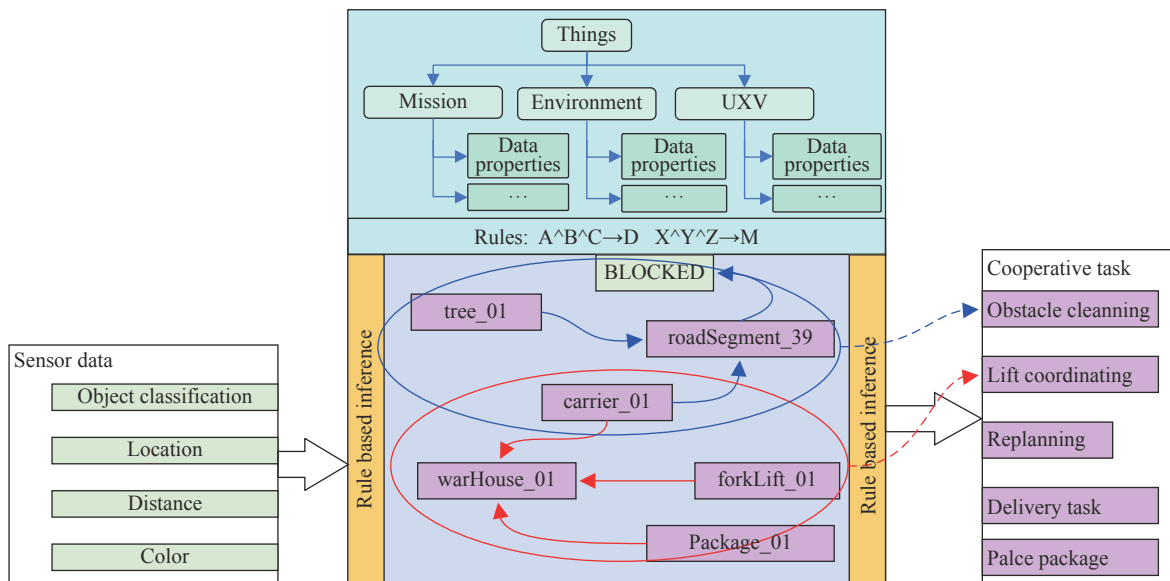
**Table 1 Entity property**

Predicate	Domain	Range
arriveAddress	UXV	Address
atWarehouse	UXV	Warehouse
onRoad	UXV	Road
hasObstacle	Obstacle	Road
hasSign	Road	Sign
hasCandidateTask	UXV	Task

**Table 2 Data property**

Predicate	Domain	Data type
hasCurrentSpeed	UXV	Float
hasState	Road	Enum
hasWidth	Obstacle	Float
hasDistance	UXV	Float
inScene	UXV	Self define scene type

In the semantic centered cloud control framework, the system implements task-level cooperative control by the scene that vehicles are facing. The connection between the scene and the vehicle’s behavior is a semantic relationship. This kind of semantic relationship is represented by semantic web rule language (SWRL). Beyond this, SWRL is also used to represent relationships such as explicit data acquired by the sensor and their deep meanings. The process of discovering these two kinds of semantic relationships may be defined as situation awareness and scene understanding. Scene understanding is the process of mining deep hidden pieces of information behind the representation. Situation awareness is the process of triggering an unmanned vehicle’s behaviors or tasks when they are in a certain situation. The semantic representation of task domain knowledge and its principle of driving scene understanding and situation awareness is shown in Fig. 3.



**Fig. 3 Semantic representation of task domain knowledge**

### 3.3 Implement of knowledge engine

A semantic knowledge engine is an interactive access system for semantic KB based on Jena Fuseki [21]. Jena Fuseki is deployed in the cloud, and Fig. 4 shows its structure. Two main interfaces for human-machine interaction are implemented: one is the reasoner interface, which is mainly used for importing rules; the other is the ontology interface, which is mainly responsible for adding or modifying descriptive pieces of knowledge. Vehicles as nodes connected to the cloud may access the semantic knowledge engine based on hyper text transfer protocol (HTTP) protocol, query or update information

by SPARQL protocol and RDF query language (SPARQL) [22]. The inference is implemented in the form of query for the KB. These queries fall into two categories: simple query and complex query. Simple query mainly queries for explicitly defined facts, that is data properties of a single entity (such as orientation, connectivity, and state). Complex query has two more circumstances. One is querying for inference results based on axioms and predefined SWRL rules. These rules must be strictly validated to ensure correctness. The other is querying for inference results based on temporary and local rules that may not be suitable in all situations. These queries are implemented in the form of SPARQL.

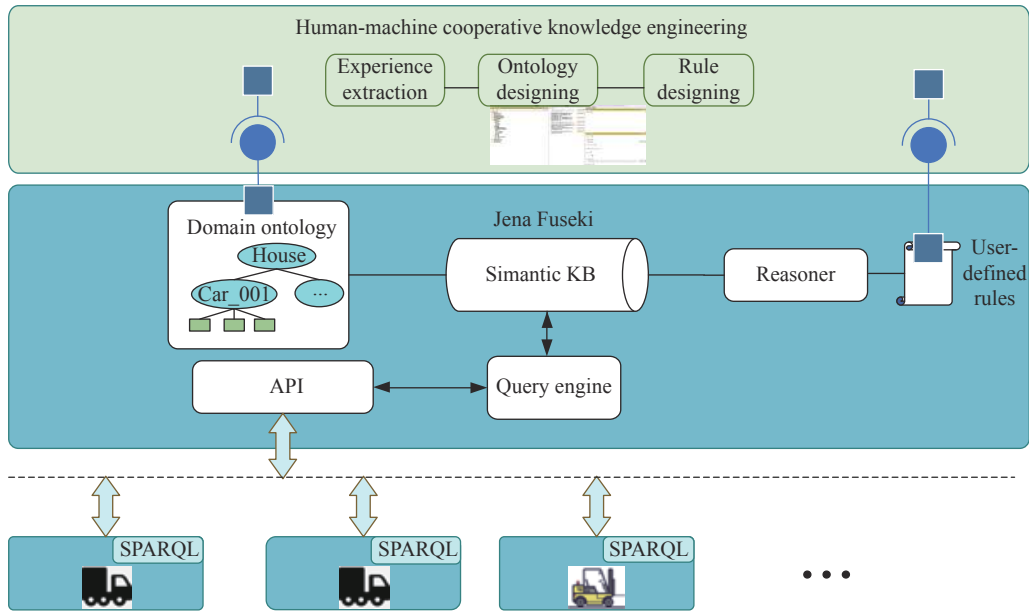


Fig. 4 Diagram of the knowledge engine

When the cloud control system runs, all nodes keep a connection to the cloud by Algorithm 1. This algorithm registers the vehicles when they first connect to the cloud and keeps updating the state of the vehicles and data acquired by the sensor. These data are the raw material for scene understanding. When any node makes a certain query, the knowledge engine conducts inference based on these facts and the rules predefined.

---

#### Algorithm 1 Semantic synchronization algorithm

---

##### Input:

Interface of KB;  
Topic and service of UXV component;

##### Output:

State update result;

---

```

swarmInit {
  subscribeComponentTopic ();
  connectToKB (KB);
  if uxv not instantiated in KB then
    creat_uxv (uxv);
  end
}
excuse with fixedperiod {
  if uxv state changed then
    update individual state in KB;
    update inferred knowledge;
  end
}

```

---

## 4. Inference based scene understanding and cloud task coordinating

The process of scene understanding is generally divided

into two main stages [23–25]. In the first stage, standard target detection networks such as region-convolution neural network (R-CNN) [26], faster-R-CNN [27], and You Only Look Once (YOLO) [28]. are used to identify the object and obtain the bounding box of the object according to the input image. Then, in the next stage, the object is instantiated as instances, and the object’s bounding box features, object labels, and spatial coordinates are instantiated as instance features, and use long short-term memory (LSTM) [29], gated recurrent unit (GRU) [30], TreeLSTM [31], etc. to predict the relationship between objects. Predicates are usually from specific vectors such as Word2Vec [32], and ConceptNet [33]. In this study, the predicate vector is obtained from the task ontology rather than a certain data set. For the problem of relationship prediction, a hybrid scene understanding method that combines the advantages of deep learning and rule-based inference is proposed. Cloud control framework based on this method has the advantages of end-to-end learning, concise, efficient and quick adaption to different task scenarios as required.

### 4.1 Object detection based on deep learning

YOLO series network has the advantages of fast and relatively high accuracy. In recent years, researchers have improved its performance by adding an additional pooling layer between the backbone network and the head. YOLO3-SPP adds the spatial pyramid pooling (SPP) [34] module to realize the multi-scale overlap of images, which improves the accuracy of the network in predict-

ing images of different resolutions and objects of different sizes. This paper chooses YOLO3-SPP as the target detection network.

## 4.2 Estimation of spatial parameters

Extraction of semantic information needed by unmanned system tasks varies from that in the field of computer vision which pursues rich picture information. Spatial parameters and spatial semantic relations are crucial to the execution of tasks. Therefore, after completing basic target detection and image recognition, it is necessary to estimate and infer the spatial parameters and spatial semantic relations of entities in the scene. In the field of robotics, spatial parameter estimation relies heavily on stereo vision, depth information, or point cloud information. However, in the real mission scenario, the vehicles usually observe objects from a distance, and it is very difficult to obtain depth information and point cloud data. Therefore, this paper is based on the use of a monocular camera image to achieve spatial parameter estimation.

The monocular camera acquires a two-dimensional image, and it is difficult to estimate the distance using Euclidean distance directly. Inverse perspective mapping (IPM) is an effective method for realizing the conversion from two-dimensional pixel points to the three-dimensional actual position [35]. In the conversion process, it only needs to know the camera's position, height, angle of view, and the inherent parameters of the camera, which are available in advance. The relationship between the position in the two-dimensional image of the object and its position in the real world can be expressed as

$$\begin{bmatrix} u & v & 1 \end{bmatrix}^T = \mathbf{KRT} \begin{bmatrix} X_w & Y_w & Z_w & 1 \end{bmatrix}^T \quad (2)$$

where  $u$  and  $v$  represent the number of pixels of the target recognition frame in the horizontal and vertical directions.  $X_w, Y_w, Z_w$  represent the position of the object in the world coordinate system.  $\mathbf{K}$  is the inherent parameter of the camera.  $\mathbf{R}$  and  $\mathbf{T}$  respectively represent the rotation and translation matrices from the world coordinate system to the two-dimensional image coordinate system.

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & -\frac{h}{\sin\theta} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4)$$

$$\mathbf{K} = \begin{bmatrix} fk_u & s & c_x & 0 \\ 0 & fk_v & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad (5)$$

where  $h$  is the height of the camera,  $f$  is the focal length,  $k_u$  and  $k_v$  are the numbers of horizontal and vertical pixels of the camera,  $s$  is the image zoom ratio, and  $c_x$  and  $c_y$  are used to correct the main optical axis of the image plane.

Assume that the  $Z$ -axis value of the target in world coordinates is 0, then the relationship between the world coordinate system and the image coordinate system is

$$\begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}. \quad (6)$$

Relative coordinates can be calculated by

$$(X_w, Y_w) = \left( \frac{m_{11} \times u + m_{12} \times v + m_{13}}{m_{31} \times u + m_{32} \times v + m_{33}}, \frac{m_{21} \times u + m_{22} \times v + m_{23}}{m_{31} \times u + m_{32} \times v + m_{33}} \right). \quad (7)$$

## 4.3 Scene understanding and situation awareness

Target detection and spatial parameter estimation obtain the basic physical dimensions and spatial parameters of environment entities, but these parameters cannot support the autonomous behavior of the cloud control system. Thus, as shown in Subsection 2.2, SWRL rules are used to mine deep semantic relationships behind the data, figure out the situation a single node is in, and take proper actions that are beneficial to achieving the swarm task goal.

SWRL is described as antecedent  $\rightarrow$  consequent.

Both antecedent and consequent consist of zero or more atoms. SWRL as production rules may infer new information based on the information already known, so it is an efficient way to infer information that is hidden behind the data but is important to the cloud system's autonomy behavior. For example, when a UGV encounters a fallen tree on road, the cloud control system may infer whether the road is blocked or not by the physical size of the obstacle and width of the road, and the road's state then is the decision basis for cloud task coordinating, as shown in the following:

---

```
Tree(?tree)^ on(?tree, ?road)^ Road(?road)^ hasWidth(?tree, ?wk)^ hasWidth(?road, ?wd)^ swrlb:divide(?wd, 2)^ swrlb:greaterThan(?wk, ?wd)  $\rightarrow$  hasState(?road, BLOKED)
```

---

Furthermore, based on the information inferenced, the cloud control system may recognize the situation it confronts, and perform further inference about the actions the system can take. For example, when the road is blocked, the cloud control system should dispatch another wrecker for obstacle cleaning task. The rule for triggering the obstacle cleaning task is shown as follows:

$$\text{Carrier(?u) \wedge Wrecker(?w) \wedge Road(?r) \wedge Tree(?tree) \wedge at(?u, ?r) \wedge hasState(?r, BLOCKED) \wedge on(?tree, ?road) \wedge hasMass(?t, ?m) \wedge hasMaxLift(?w, ?l) \wedge swrlb:lessThan(?m, ?l) \rightarrow hasCandidateTask(?w, OBSTACLE\_CLEANNING)}$$

SWRL rules used for scene semantic information extraction are shown in [Table 3](#).

**Table 3** SWRL rules in scene understanding and situation awareness

Number	Rule body	Description
Rule#1	$Uxv(?x) \wedge isIdle(?x, ?y) \wedge swrlb: booleanNot(?y, false) \rightarrow UxvAvailable(?x)$	Infer the availability of vehicles
Rule#2	$UxvAvailable(?x) \wedge hasEndurance(?x, ?y) \wedge swrlb: greaterThan(?y, 80) \rightarrow UxvEndurance(?x, HIGH)$	Duration time inference
Rule#3	$BaseStation(?bs) \wedge hasPosition(?bs, ?pos1) \wedge Uxv(?uxv) \wedge hasPosition(?uxv, ?pos2) \wedge hasRemainEndurance(?uxv, ?re) \wedge swrlb: divide(?pos1, ?pos2, ?dis) \wedge swrlb: lessThan(?dis, ?re) \rightarrow Alert(?alert) \wedge hasUxv(?alert, ?uxv) \wedge hasAlertType(?alert, INSUFFICIENT\_ENDURANCE)$	Infer low endurance alert event
Rule#4	$Uxv(?uxv) \wedge Entity(?entity) \wedge inFrontOf(?entity, ?uxv) \wedge \rightarrow Obstacle(?entity)$	Infer obstacle type
Rule#5	$Tree(?tree) \wedge hasHight(?tree, ?hight) \wedge hasWidth(?tree, ?width) \wedge swrlb: divide(?rate, ?width, ?hight) \wedge swrlb: greatThan(?rate, 4) \rightarrow hasState(?tree, FALLEN)$	Infer entity state
Rule#6	$Tree(?tree) \wedge hasHight(?tree, ?hight) \wedge hasWidth(?tree, ?width) \wedge hasDensity(?tree, ?density) \wedge swrlb: multiply(?mass, ?width, ?width, ?hight, ?density) \rightarrow hasMass(?tree, ?mass)$	Estimate the mass of the tree
Rule#7	$Tree(?tree) \wedge on(?tree, ?road) \wedge Road(?road) \wedge hasWidth(?tree, ?wk) \wedge hasWidth(?road, ?wd) \wedge swrlb: divide(?wd, 2) \wedge swrlb: greaterThan(?wk, ?wd) \rightarrow hasState(?road, BLOCKED)$	Infer the road is blocked or not
Rule#8	$Carrier(?u) \wedge Wrecker(?w) \wedge Road(?r) \wedge Tree(?tree) \wedge at(?u, ?r) \wedge hasState(?r, BLOCKED) \wedge on(?tree, ?road) \wedge hasMass(?t, ?m) \wedge hasMaxLift(?w, ?l) \wedge swrlb: lessThan(?m, ?l) \rightarrow hasCandidateTask(?w, OBSTACLE\_CLEANNING)$	Infer obstacle cleaning task
Rule#9	$Tree(?tree) \wedge hasMass(?tree, ?mass) \wedge Uxv(?uxv) \wedge hasMaxGrip(?uxv, ?grip) \wedge swrlb: greaterThan(?grip, ?mass) \rightarrow hasCandidateUxv(?task, ?uxv)$	Infer candidate task executor
Rule#10	$Carrier(?c) \wedge ForkLift(?fl) \wedge Package(?p) \wedge WarHouse(?r) \wedge at(?c, ?wh) \wedge hasTargetPakage(?c, ?p) \wedge hasMass(?p, ?m) \wedge hasMaxLift(?fl, ?l) \wedge swrlb: lessThan(?m, ?l) \rightarrow hasCandidateTask(?w, LIFT\_COOPERATION)$	Infer lift cooperative task
Rule#11	$Carrier(?c) \wedge Road(?r) \wedge MaintenanceSign(?ms) \wedge at(?c, ?r) \wedge hasState(?r, BLOCKED) \wedge hasSign(?r, ?ms) \rightarrow hasCandidateTask(?w, ROUTE\_REPLANNING)$	Replanning path when the road is blocked since maintenance
Rule#12	$Carrier(?u) \wedge Wrecker(?w) \wedge Road(?r) \wedge Tree(?tree) \wedge at(?u, ?r) \wedge hasState(?r, BLOCKED) \wedge on(?tree, ?road) \wedge hasMass(?t, ?m) \wedge hasMaxLift(?w, ?l) \wedge swrlb: greaterThan(?m, ?l) \rightarrow hasCandidateTask(?w, TOUTE\_REPLANNING)$	Replanning path when the road is blocked by an obstacle
Rule#13	$Carrier(?c) \wedge Customer(?p) \wedge Address(?a) \wedge at(?c, ?a) \wedge hasAddress(?p, ?a) \rightarrow hasCandidateTask(?c, PLACING\_PACKAGE)$	Infer placing package action
Rule#14	$Carrier(?c) \wedge ForkLift(?fl) \wedge Package(?p) \wedge WarHouse(?r) \wedge at(?c, ?wh) \wedge hasTargetPakage(?c, ?p) \wedge hasMass(?p, ?m) \wedge hasMaxLift(?fl, ?l) \wedge swrlb: greaterThan(?m, ?l) \rightarrow Alert(?alert) \wedge hasUxv(?alert, ?c) \wedge hasAlertType(?alert, LIFT\_COOPERATION\_FAILED)$	Infer alert of forklift cannot lift a package

On the whole, SWRL has the following functions, which are helpful for situation awareness:

(i) Link various attributes of environmental entities according to their inherent logic.

(ii) Link the physical attributes of environmental entities to their semantic descriptions.

(iii) Link the properties of environmental entities with the behaviors that can be imposed on them.

#### 4.4 Individual task planning and controlling

According to the assumption of the cloud control system [12], all nodes connected to the cloud should be intelli-

gent enough to undertake the coordinative task. In this paper, we use a PDDL planner as the individual planner. The detailed cloud control system is shown in Fig. 5.

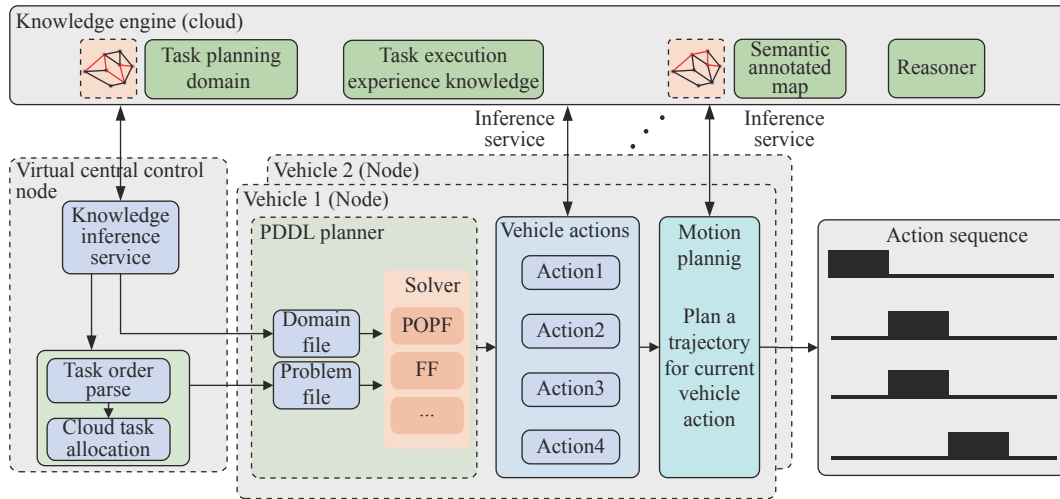


Fig. 5 Diagram of cloud control system

Knowledge of task is defined according to the PDDL language specifications and integrated into task ontology, so that knowledge inference may be integrated with task planning algorithms, enabling high-level logic inference and planning algorithms such as fast down (FD), and forward-chaining partial-order planning (POPF).

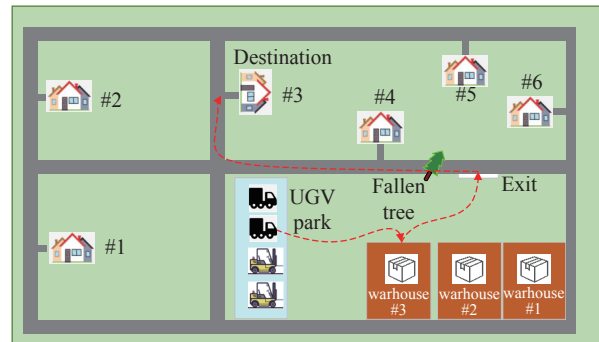
After inference, the problem file is transmitted to the individual PDDL planner of relevant vehicles. The planner completes plan generation, plan distribution, and action schedule. In the process of task execution, when the environment changes, for example, the task cannot be completed, or new task requirements are generated. The problem file would be regenerated and the planner is called to replan the task.

## 5. Experimental verification and analysis

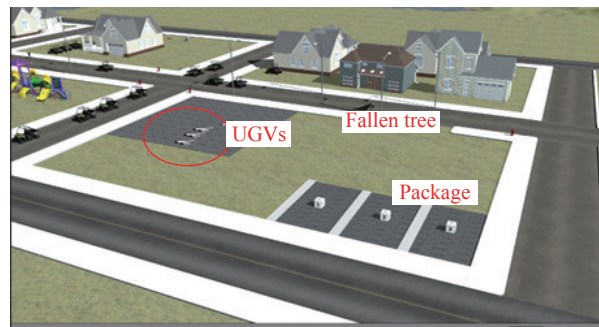
### 5.1 Design of experiment

The test scenario is shown in Fig. 6. Two delivery UGVs, a forklift, and a wrecker are in the simulation. The UGVs are equipped with a camera to detect targets. The UGVs use the A\* based route navigation method for path planning. The simulation platform is Ubuntu 18.04, ROS Melodic, Gazebo 7. The semantic ontology model of the task, primary entities of environment, and SWRL rules have been built in advance and integrated into the knowledge engine based on Jena Fuseki. The Jena Fuseki server serves as the cloud control center, and both UGVs connect to the server are nodes of the cloud control sys-

tem. The cloud control system assigns delivery UGV to conduct delivery tasks when a package arrives, and dispatches the wrecker for obstacle cleaning task or forklift for loading tasks if needed. All the coordinating tasks are dispatched autonomously.



(a) 2D scenario diagram



(b) 3D scenario diagram

Fig. 6 Simulated experiment scenario



## 5.2 Scenario

To make the experiment easier, this scenario is designed as a coordinated delivery task in a closed and limited environment. The control center generates a delivery task when a certain package arrives. The delivery UGV that can perform the task goes to the warehouse to obtain the package and perform the delivery task. During the delivery, the UGV may encounter a fallen tree on the road

which blocks the road, so the delivery task is aborted. The UGV acquires the sensor data, and synchronizes it to the knowledge engine. Cloud knowledge engine perceives the “blocked road” situation and generates the “Obstacle cleaning” task, and assigns it to a wrecker UGV. All the UGVs work coordinatively for faster delivery.

Summary of the plot of the experimental scene is shown in Table 4.

**Table 4 Plot summary of the experimental scenario**

Item	Description	Action
Task name	Delivery task with exception event	—
Precondition	#1, #2 delivery UGV, #3 wrecker UGV, #4 forklift UGV located at the depot, new package arrives	—
	UGV completes navigate autonomously	—
	UGV report to the cloud control center	—
		Query for related information: candidate vehicle, package location, destination location
	#1 or #2 UGV conduct delivery task (Task1)	Get package (Action 11)
		Navigate to destination (Action 12)
		Complete delivery (Action 13)
Event flow		Query for related information: Obstacle type, location
	#3 wrecker conduct Obstacle_cleaning task (Task 2)	Navigate to the obstacle (Action 21)
		Clean up obstacle (Action 22)
		Navigate to the depot (Action 23)
		Query for related information: Package location
	#4 forklift conduct lift cooperative task when needed (Task 3)	Navigate to package (Action 31)
		Load package (Action 32)
		Navigate to the depot (Action 33)
Exception handling	#1 or #2 UGV delivery task failed since the road blocked	Generate scene graph and report to control center
		According to the inference result, generate a new “obstacle cleaning” task (Task 2)
	#1 or #2 UGV delivery task failed since road maintenance	Acquire sensor data and update cloud KB information
		According to the inference result, replanning the path
	#1 or #2 UGV arrive at a warehouse	Acquire sensor data and update cloud KB information
		According to the inference result, generate a new “Lift_Cooperation” task (Task 3)
	#1 or #2 UGV arrives at customer address	According to the inference result, unload the package
Effect	Each vehicle completes a designed task and reports to the cloud control center	—

## 5.3 Deep learning data set preparation and network training

To verify the target detection, a database is constructed based on the experimentally scenes. The categories of ob-

jects to be recognized are houses, roads, road signs, trees, and cars. In the constructed Gazebo 3D environment, the pictures of each kind of object under different angles are collected and the images are labeled with the Labelme [36]. The network training results are shown in Fig. 7.

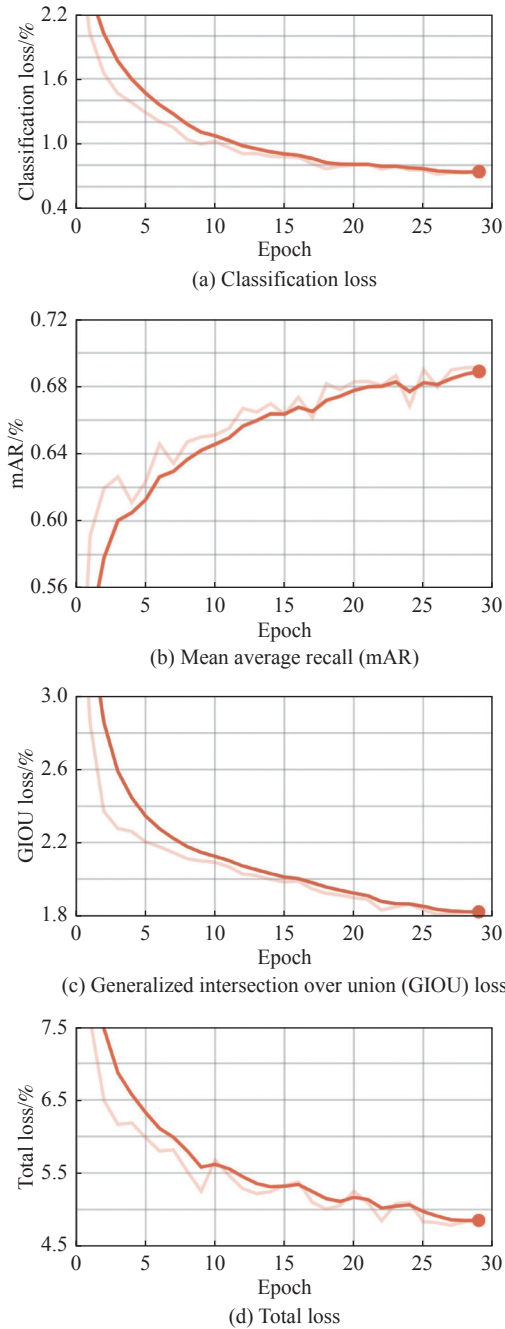


Fig. 7 Training results of YOLO network

### 5.4 Ontology model and inference function test

The ontology model and inference function test is used to find whether the ontology model and rules properly meet the requirements of scene modeling. The test was carried out using the Drools plug-in in protege. After the construction of the ontology model, some virtual facts are added to the ontology for the test.

Four scenarios are tested: (i) Infer entity state based on sensor data; (ii) Infer the lift cooperative task when delivery UGV arrives at the warehouse; (iii) Infer road block-

ing state and obstacle type according to sensor data; (iv) Infer the cooperative task of cleaning obstacles when the delivery UGV encounters road closure. The test results are shown in Fig. 8.

Raw data:  
Tree's size, density, etc.

Inference result:  
EquivalentProperties: hasName  
SameIndividual: tree\_001  
tree\_001 hasState "FALLEN"<sup>^^xsd:string</sup>  
Wrecker SubClassOf Wrecker  
owl:Nothing SubClassOf Probability

Infer entity state, time consumed: 38 ms

(a) Reasoning for entity state

Raw data:  
UGV arrives at the warehouse, package at the warehouse, etc.

Inference result:  
UGV SubClassOf UGV  
LIFT\_COOPERATION liftPackage package\_01  
StraightRoad SubClassOf StraightRoad  
owl:Nothing SubClassOf Obstacle  
carrier\_01 hasCandidateTask LIFT\_COOPERATION  
Predicates SubClassOf owl:Thing

Infer lift cooperative task, time consumed: 60 ms

(b) Reasoning for transit task

Raw data:  
Road's width, tree's size, etc.

Inference result:  
Tree SubClassOf Tree  
roadSegment\_32 hasState "BLOCKED"<sup>^^xsd:string</sup>  
owl:topObjectProperty SubPropertyOf owl:topObjectProperty  
EquivalentProperties: owl:topDataProperty  
tree\_001 Type Obstacle  
Probability SubClassOf owl:Thing

Infer road state, time consumed: 63 ms

(c) Reasoning for road state

Raw data:  
Road's state, tree's mass, Maximum lifting capacity of wrecker, etc.

Inference result:  
EquivalentClasses: PlaceCargo  
carrier\_02 hasCandidateTask OBSTACLE\_CLEANNING  
RePlanning SubClassOf RePlanning  
carrier\_02 owl:topObjectProperty OBSTACLE\_CLEANNING  
OBSTACLE\_CLEANNING willClean tree\_001  
warHouse\_01 Type Environment

Infer obstacle cleaning cooperative task, time consumed: 77 ms

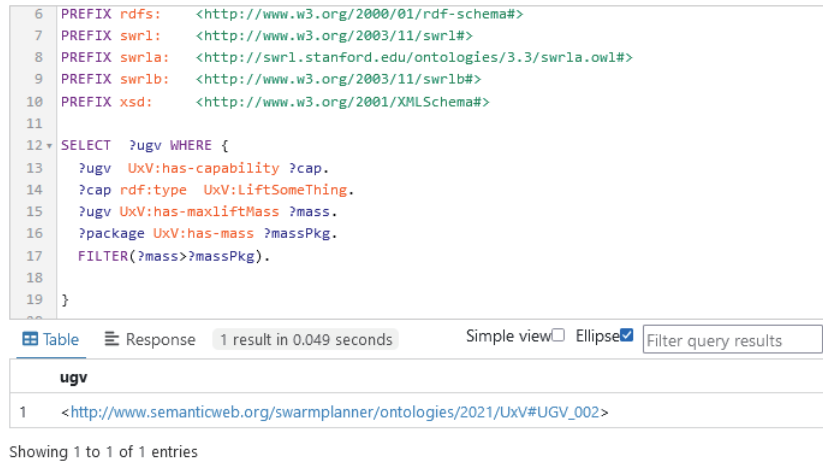
(d) Reasoning for the obstacle cleaning task

Fig. 8 Logic test result of knowledge engine

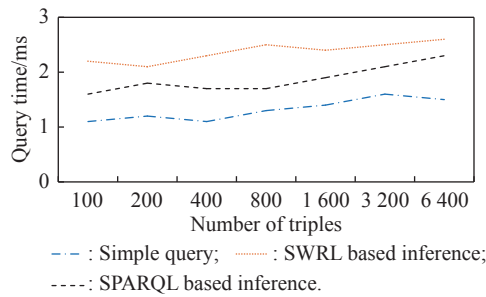
### 5.5 KB's responsiveness to different types of inference

According Subsection to 3.3, knowledge queries fall into three categories: simple query, SWRL based inference, and SPARQL based inference [37]. An example of SPARQL based inference statement and result is shown in

Fig. 9(a). Response time of different queries is shown in Fig. 9(b). It can be seen that the query type is the main factor affecting the time consuming, and the overall query time is relatively low, which can meet the real-time requirements of task planning.



(a) Query for candidate vehicles



(b) Response time of different queries

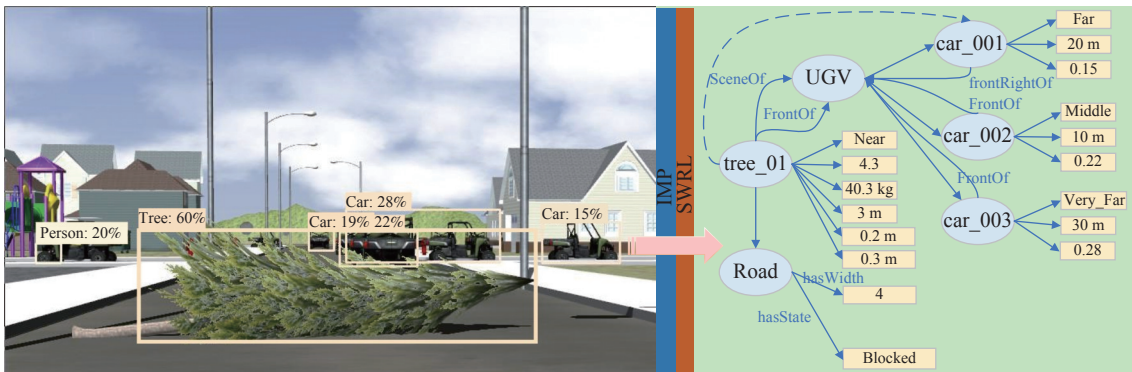
Fig. 9 Responsiveness of KB

### 5.6 Semantic centered scene understanding and intelligent task replanning

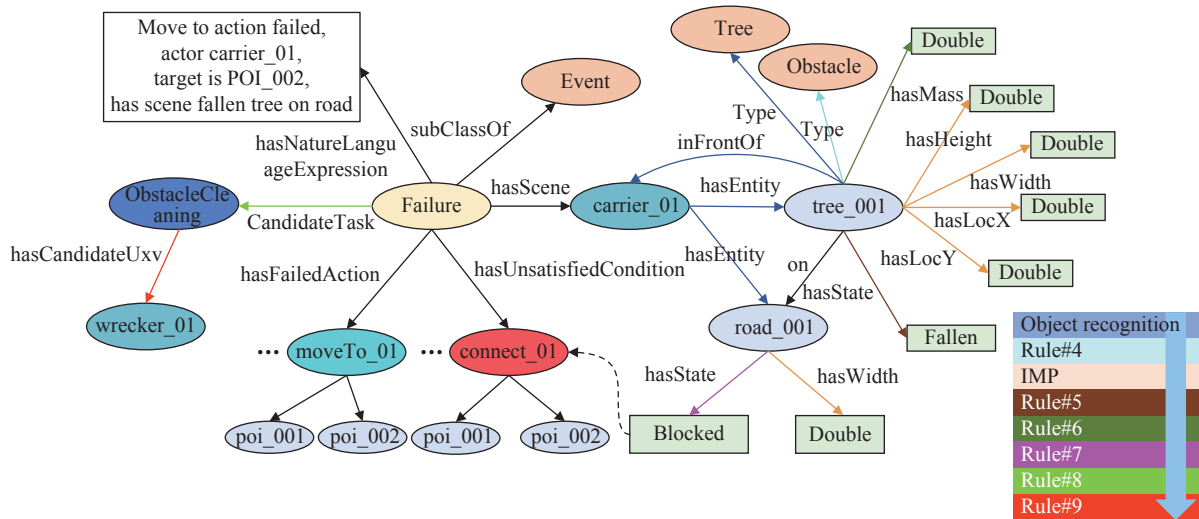
When the delivery UGV encounters a fallen tree obstacle, the unmanned vehicle classifies the tree obstacle, estimates its size, location, etc. These data are updated to the

cloud; then, when the UGV queries for the candidate task, the cloud knowledge engine inferses road state, tree mass. When the road is blocked, classifies tree is an instance of Obstacle class.

This process is scene understanding and its procedure is shown in Fig. 10.



(a) UGV encountering obstacle on the road



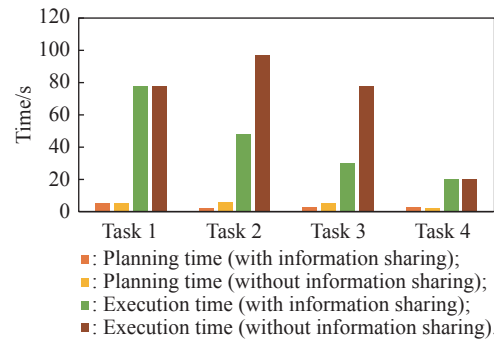
(b) Process of scene understanding and triggering autonomous cooperation behaviors

**Fig. 10** Extracting semantic knowledge of the scene

The cloud KB first infers the deep scene information according to sensor data such as entity category, size, and location. Then, if the scene meets the situation predefined as SWRL rules, a cooperative task order may be obtained by querying for “hasCandidateTask” predicate, as shown in Fig. 10(b).

The cloud KB provides an information-sharing mechanism that inspires swarm intelligence. When delivery UGV finds the road is blocked, this information is synchronized to the cloud, the follow-up delivery task would consider this condition when planning. Thus, four delivery tasks are designed to verify the advantage of the cloud control framework, task1–task4 corresponding to the delivery package for address #1–#4 in Fig. 6(a). Task planning time and task execution time with and without cloud information sharing are shown in Fig. 11. The scenario is that packages for address #1–#4 arrive sequentially, when performing task 1, the delivery UGV finds the road is blocked and the cloud control system dispatches a wrecker to clean the obstacle and the delivery UGV replanning its path, the blocking situation persists before task 4 starts. It can be seen that during the execution of task 2 and task 3, due to the realization of information sharing within the cloud system, the follow-up UGVs can avoid the blocked road. However, in the absence of information sharing, the UGVs could not be aware of the situation of the road segment being blocked until they arrive at the blocked road segment, which cost extra task planning time and execution time. The information-sharing mechanism provided by the knowledge engine optimizes the task performance of all vehicles as a

whole and shows the feature of preliminary swarm intelligence.

**Fig. 11** Comparison of performance of task planning with and without knowledge sharing

## 6. Conclusions

In this paper, we propose a semantic-centered cloud control framework for heterogeneous cooperative autonomous unmanned systems. The semantic knowledge is represented by ontology, and a semantic knowledge engine is designed to express the domain knowledge which is applied to the cloud task coordination. Then a hybrid scene understanding method combining deep learning with rule-based knowledge reasoning is proposed, which extracts the knowledge required for task execution, and triggers autonomous behavior of the cooperative cloud system. In addition, natural language processing attributes are added for each scene to lay the foundation for the next step of realizing natural language-based task collaboration and human-machine interaction. Finally, simulation experiments verify the feasibility of the framework.

The semantic centered cloud control system is realized according to the knowledge-based method, and works as a knowledge parse system. These make it easier to be applied into other fields such as agriculture, industrial manufacture, or supplies delivery and medical evacuation in the military. This research shows a feasible way to realize the cognitive ability of autonomous unmanned systems at task-level. However, there is still much to do to deal with large-scale graph problems if the scenario gets more complicated.

## References

- [1] GU Y M. Overview of technology development of the USV group. *Ship Science and Technology*, 2019, 41(12): 35–38. (in Chinese)
- [2] HU X F, QI D W. On problems of intelligent decision-making—how far is it from game-playing to operational command. *Journal of Command and Control*, 2020, 6(4): 356–363. (in Chinese)
- [3] TANG J. Professor Tang Jie deep report: the next decade of artificial intelligence. <https://zhuanlan.zhihu.com/p/133517890>.
- [4] TENORTH M. Knowledge processing for autonomous robots. Munich: Technical University of Munich, 2011.
- [5] LI X, SONIA B, TAMARA M W, et al. SWARMS ontology: a common information model for the cooperation of underwater robots. *Sensors*, 2017, 17(3): 569–588.
- [6] HUANG L, LIANG H W, YU B, et al. Ontology-based driving scene modeling, situation assessment and decision making for autonomous vehicles. *Proc. of the 4th Asia-Pacific Conference on Intelligent Robot Systems*, 2019: 57–62.
- [7] PIGNATON E, OLSZEWSKA J I, CARBONERA J L, et al. Ontological concepts for information sharing in cloud robotics. *Journal of Ambient Intelligence and Humanized Computing*, 2020. DOI:10.1007/s12652-020-02150-4.
- [8] MARQUES P, MANFROI D, DEITOS E, et al. An IoT-based smart cities infrastructure architecture applied to a waste management scenario. *Ad Hoc Networks*, 2019, 87: 200–208.
- [9] BRUCKNER D, PICUS C, VELIK R, et al. Hierarchical semantic processing architecture for smart sensors in surveillance networks. *IEEE Trans. on Industrial Informatics*, 2012, 8(2): 291–301.
- [10] MENG Z Z, WU Z P, GRAY J. A collaboration-oriented M2M messaging mechanism for the collaborative automation between machines in future industrial networks. *Sensors*, 2017, 17(11): 2694–2709.
- [11] LU Y Q, ASGHAR M R. Semantic communications between distributed cyber-physical systems towards collaborative automation for smart manufacturing. *Journal of Manufacturing Systems*, 2020, 55: 348–359.
- [12] XIA Y Q. Cloud control systems. *IEEE/CAA Journal of Automatica Sinica*, 2015, 2(2): 134–142.
- [13] YIN L L. Research on uncertain environment cognition for unmanned maritime systems. Harbin: Harbin Engineering University, 2015. (in Chinese)
- [14] MIGUELANEZ E, PATRON P, BROWN K E, et al. Semantic knowledge-based framework to improve the situation awareness of autonomous underwater vehicles. *IEEE Trans. on Knowledge and Data Engineering*, 2011, 23(5): 759–773.
- [15] EKVALL S, KRAGIC D. Robot learning from demonstration: a task-level planning approach. *International Journal of Advanced Robotic Systems*, 2008, 5(3): 223–234.
- [16] KIRSCH A. Integration of programming and learning in a control language for autonomous robots performing everyday activities. Munich: Technical University of Munich, 2008.
- [17] MOKHTARI V, LIM G H, LOPES L S. Gathering and conceptualizing plan-based robot activity experiences. *Advances in Intelligent Systems and Computing*, 2016, 302: 993–1005.
- [18] CHEN Z X. Study on approaches of automated planning of service robot for complex environment. Shenzhen: University of Chinese Academy of Sciences, 2019. (in Chinese)
- [19] HASLUM P, LIPOVETZKY N, MAGAZZENI D, et al. An introduction to the planning domain definition language. San Mateo: Morgan Kaufmann Publishers, 2019.
- [20] MUSEN M A. The protégé project: a look back and a look forward. *AI Matters*, 2015, 1(4): 4–12.
- [21] APACHE J. Apache Jena Fuseki. <https://jena.apache.org/documentation/fuseki2/>.
- [22] JOVANOVIK M, HOMBURG T, SPASIC M. Software for the GeoSPARQL compliance benchmark. *Software Impacts*, 2021, 8: 100071–100073.
- [23] TANG K H, NIU Y L, HUANG J Q, et al. Unbiased scene graph generation from biased training. *Proc. of the Conference on Computer Vision and Pattern Recognition*, 2020: 3713–3722.
- [24] REZAEI M, AZARMI M. DeepSOCIAL: social distancing monitoring and infection risk assessment in COVID-19 pandemic. *Applied Sciences*, 2020, 10(21): 1–30.
- [25] SUHAIL M, MITTAL A, SIDDIQUIE B, et al. Energy-based learning for scene graph generation. *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021: 13936–13945.
- [26] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014: 580–587.
- [27] REN S Q, HE K M, GIRSHICK R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. on Pattern Analysis & Machine Intelligence*, 2017, 39(6): 1137–1149.
- [28] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016: 779–788.
- [29] SHAO T, WU D O. Graph-LSTM with global attribute for scene graph generation. *Journal of Physics: Conference Series*, 2003, 2003: 012001–012007.
- [30] CHUNG J, GULCEHRE C, CHO K, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. <https://doi.org/10.48550/arXiv.1412.3555>.
- [31] TANG K H, ZHANG H W, WU B Y, et al. Learning to compose dynamic tree structures for visual contexts. *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019: 6612–6621.
- [32] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space. *Proc. of the International Conference on Learning Representations*, 2013: 1–13.

- [33] SPEER R, CHIN J, HAVASI C. ConceptNet 5.5: an open multilingual graph of general knowledge. <https://doi.org/10.48550/arXiv.1612.03975>.
- [34] HE K M, ZHANG X Y, REN S Q, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. on Pattern Analysis & Machine Intelligence*, 2014, 37(9): 1904–1916.
- [35] REZAEI M, KLETTE R. *Computer vision for driver assistance*. Cham: Springer International Publishing, 2017.
- [36] BRYAN R, ANTONIO T, KEVIN M, et al. LabelMe: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 2008, 77(1/3): 157–173.
- [37] ZHAI Z Y, MARTINEZ J F, LUCAS N, et al. A rule-based reasoner for underwater robots using OWL and SWRL. *Sensors*, 2018, 18(10): 3481.

## Biographies



**PANG Weijian** was born in 1986. He received his B.S. degree in radar engineering and M.S. degree in electronic science and technology from Air Force Engineering University, Xi'an, China, in 2009 and 2014 respectively. He is currently a doctoral candidate of the Academy of Military Sciences. His research interests are military operational research, and military application of intel-

ligent unmanned systems theory and technology.

E-mail: pangweijian2013@163.com



**LI Hui** was born in 1964. He received his M.S. degree in military thoughts from the Academy of Military Sciences, Beijing, China, in 1993. He is currently a researcher of the Academy of Military Sciences. His research interests are military operational research and military assessment.

E-mail: 13391883525@189.sina.com



**MA Xinyi** was born in 1984. She received her B.S. degree in computer science and technology and M.S. degree in software engineering from Beihang University, Beijing, China, in 2006 and 2010 respectively. She is currently a doctoral candidate of the Academy of Military Sciences. Her research interests are military operational research and causal inference in military applications.

E-mail: cynthiacatmm@163.com



**ZHANG Hailin** was born in 1985. He received his B.S. degree in radar engineering, and M.S. and Ph.D. degrees in science of military equipment from Air Force Engineering University, Xi'an, China, in 2007, 2009 and 2015 respectively. He is currently a researcher of the Academy of Military Sciences. His research interests are military operational research and strategic assessment of mili-

tary issues.

E-mail: zhanghailin0350@163.com