

Ontologies for cloud robotics

EDISON PIGNATON DE FREITAS¹, JULITA BERMEJO-ALONSO² , ALAA KHAMIS³,
HOWARD LI⁴ and JOANNA ISABELLE OLSZEWSKA⁵

¹*University Federal of Rio Grande do Sul, Brazil*
e-mail: epfрейtas@inf.ufrgs.br

²*Universidad Politécnica de Madrid, Spain*
e-mail: julia.bermejo@upm.es

³*General Motors, Canada*
e-mail: alaakhamis@gmail.com

⁴*University of New Brunswick, Canada*
e-mail: howard@umb.ca

⁵*University of West of Scotland, UK*
e-mail: joanna.olszewska@ieee.org

Abstract

Cloud robotics (CR) is currently a growing area in the robotic community. Indeed, the use of cloud computing to share data and resources of distributed robotic systems leads to the design and development of cloud robotic systems (CRS) which constitute useful technologies for a wide range of applications such as smart manufacturing, aid and rescue missions. However, in order to get coherent agent-to-cloud communications and efficient agent-to-agent collaboration within these CRS, there is a need to formalize the knowledge representation in CR. Hence, the use of ontologies provides a mean to define formal concepts and their relations in an interoperable way. This paper presents standard robotic ontologies and their extension in the CR domain as well as their possible implementations in the case of a real-world CR scenario.

1 Introduction

The term ‘cloud robotics’ (CR) was first coined by Kuffner (2010). It refers to any robot or automation system that utilizes the cloud infrastructure for either data or code of its execution, that is, a system where all sensing, computation, and memory are not integrated into a single standalone system (Kehoe *et al.*, 2015).

Cloud robotic systems (CRS) appeared at the beginning of this decade, and their presence is increasing in both industrial utilization (Rahimi *et al.*, 2017) and daily life use (Sharath *et al.*, 2018). Hence, applications of CRS range from smart factory (Rahman *et al.*, 2019) including teleoperation (Salmeron-Garcia *et al.*, 2015) to smart city (Ermacora *et al.*, 2013), encompassing surveillance (Bruckner *et al.*, 2012), urban planning (Bruckner *et al.*, 2012), disaster management (Jangid & Sharma, 2016), ambient-assisted living (Quintas *et al.*, 2011), life support (Kamei *et al.*, 2012), and elderly care (Kamei *et al.*, 2017).

Although CR is still a nascent technology (Jordan *et al.*, 2013), a number of start-ups and spin-offs have been recently created to develop and commercialize CR targeting different application domains. For example, Rapyuta Robotics (Hunziker *et al.*, 2013; Mohanarajah *et al.*, 2015), which is an Eidgenössische Technische Hochschule spin-off, builds cloud-connected, low-cost multi-robot system for security and inspection based on RoboEarth (Waibel *et al.*, 2011), (Riazuelo *et al.*, 2015). CloudMinds is a CR start-up for housekeeping and family services. SoftBank, which is a Tokyo-based phone and Internet service provider, has recently invested \$100 million in the cloud robot Pepper created by Aldebaran Robotics. In

June 2015, SoftBank Robotics announced that the first 1000 units of this robot were sold out in less than 1 minute (Guizzo, 2011).

In particular, CRS are characterized by three levels of implementation (Zhang & Zhang, 2019): (i) the infrastructure as a service (Du *et al.*, 2011) to run, for example, the operating systems on the cloud; (ii) the platform as a service to share, for example, data (Tenorth *et al.*, 2013), system architectures (Miratabzadeh *et al.*, 2016), etc.; and (iii) the software as a service (Arumugam *et al.*, 2010) to allow, for example, the process and the communication of the data. Hence, CRS use the cloud for robotic systems to share their data about the context (Huang *et al.*, 2019), the environment (Wang *et al.*, 2015), the resources (Chen *et al.*, 2018), their formation (Turnbull & Samanta, 2013), their navigation (Hu *et al.*, 2012), or to perform tasks such as the simultaneous localization and mapping (SLAM) (Benavidez *et al.*, 2015), the grasping of objects (Zhang *et al.*, 2019), leading to define path planning (Lam & Lam, 2014) or robot control as services (Vick *et al.*, 2015), and more generally, robotic as a service (Doriya *et al.*, 2012).

Thence, by studying the context of the CR field, its importance, and the need for standards that can support its growth, this paper brings a contribution in identifying and testing ontologies that aim to define and facilitate access to various types of knowledge representation and shared resources for the CRS.

The rest of the paper is structured as follows. Section 2 presents existing standard ontologies for the CR domain, along with a literature overview about relevant efforts in the CRS field. Section 3 describes a real-world case study providing an insight into the use of the presented ontologies to enhance information sharing in a CRS, while Section 4 concludes the work with reflections and future directions.

2 Ontologies for cloud robotic systems

2.1 Overview

In light of the need for standardization of the shared data among the CRS' agents, the use of ontologies plays an important role (Kitamura & Mizoguchi, 2003). Indeed, ontology specifies both formally and semantically the key concepts, properties, relationships, and axioms of a given domain (Olszewska & Allison, 2018). In general, ontologies make the relevant knowledge about a domain explicit in a computer-interpretable format, allowing reasoning about that knowledge to infer new information. Some of the benefits of using ontology are thus the definition of a standard set of domain concepts along with their attributes and inter-relations, and the possibility for knowledge capture and reuse, facilitating systems specification, design and integration as well as accelerating research in the field (Olszewska *et al.*, 2020).

Over the past, there have been several projects focusing on the use of ontologies for CR applications, in order to express the vocabulary and knowledge acquired by robots in specific context such as bioinformatics (Soldatova *et al.*, 2006), rehabilitation (Dogmus *et al.*, 2015), or kit building (Balakirsky *et al.*, 2012). Indeed, through ontologies, robots are able to use a common vocabulary to represent their knowledge in a structured, unambiguous form and perform reasoning over these data (Muhayyuddin & Rosell, 2015). However, the growing complexity of behaviors that robots are expected to present naturally entails, on one hand, the use of increasingly complex knowledge, calling for an ontological approach and, on the other hand, the need for multi-robot coordination and collaboration, requiring a standardized approach.

Thence, the remaining sections (Sections 2.2 and 2.3) are focused on two ontologies, namely the Ontology for Autonomous Systems (OASys) and the Robotic Cloud Ontology (ROCO), which are part of the IEEE standardization effort dedicated to Robot Task Representation (RTR) (P1872.1) and the Standard for Autonomous Robotics (AuR) (P1872.2), respectively. As explained in Section 3, these OASys and ROCO standard ontologies can be extended to the CR domain and applied in CRS.

2.2 Ontology for autonomous systems ontology

The OASys (Bermejo-Alonso & Sanz, 2011) describes the domain of autonomous systems as software and semantic support for the conceptual modeling of autonomous system's description (ASys Ontology)

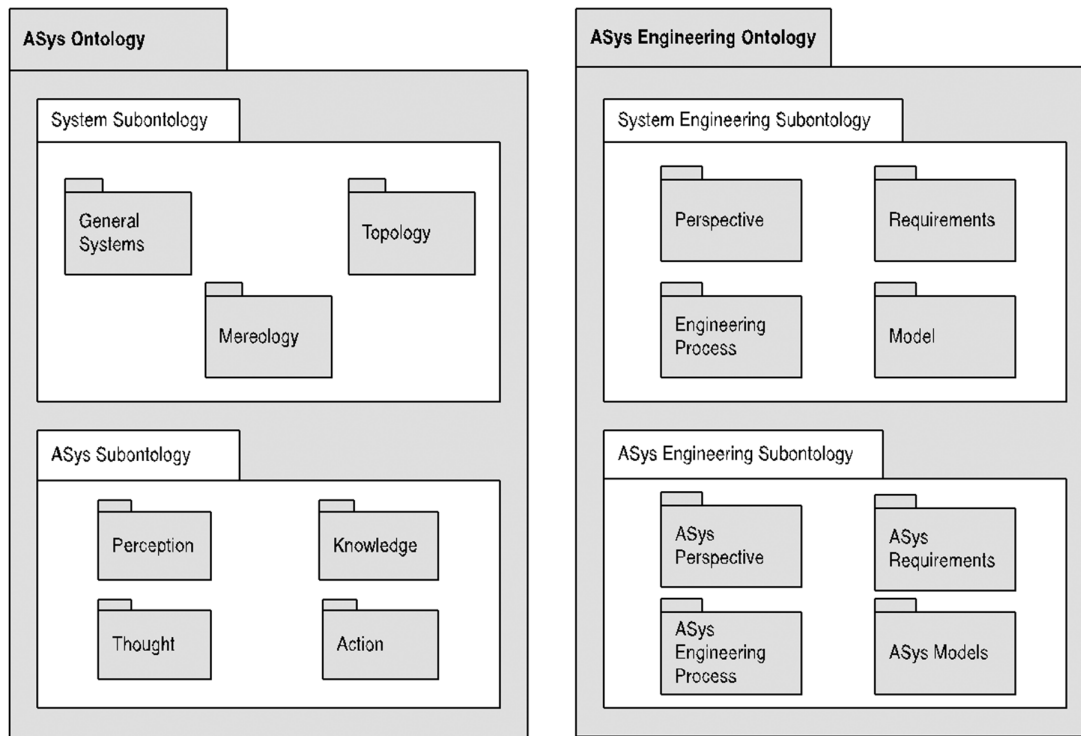


Figure 1 OASys ontologies, subontologies, and packages

and engineering (ASys Engineering Ontology) (Figure 1) and could be used for CR. Indeed, OASys organizes its content in the form of different subontologies and packages. Subontologies address the intended use for autonomous systems' description and engineering at different levels of abstraction, whereas packages within a subontology gather aspect-related ontological elements. Both subontologies and packages have been designed to allow future extensions and updates to OASys. The different subontologies and packages have been formalized using Unified Modeling Language (UML) (Olszewska *et al.*, 2014).

The ASys Ontology gathers the ontological constructs necessary to describe and characterize an autonomous system at different levels of abstraction, in two subontologies, as follows:

1. The System Subontology contains the elements necessary to define the structural and behavioral features of any system.
2. The ASysSubontology specializes the previous concepts for autonomous systems, consisting of several packages to address their structure, behavior, and function.

The ASys Engineering Ontology is dedicated to system's engineering concepts. It comprises two subontologies to address at a different level of abstraction the ontological elements to describe the autonomous system's engineering development process, as follows:

1. The System Engineering Subontology gathers the concepts related to an engineering process as general as possible, based on different metamodels, specifications, and glossaries used for software-based developments.
2. The ASys Engineering Subontology contains the specialization and additional ontological elements to describe an autonomous system's generic engineering process, organized in different packages.

The OASys-driven Engineering Methodology (ODEM) guides the application of the OASys elements for the semantic modeling of an autonomous system considering the phases, the tasks, and the work products (as conceptual models and additional documents). It uses as guideline the ontological elements in the

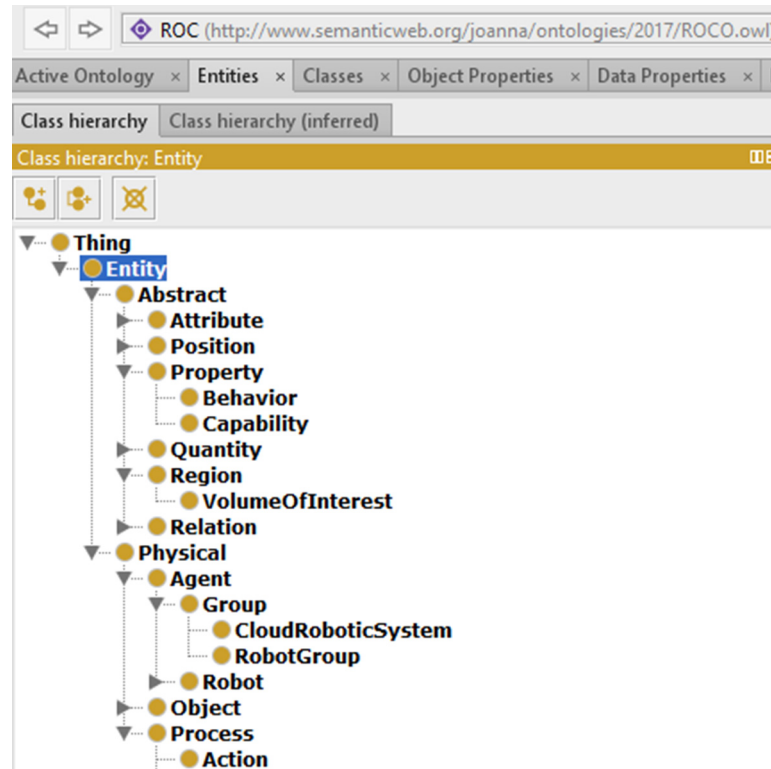


Figure 2 Excerpt of the cloud robotic ontology concepts implemented in OWL language

System Engineering and ASys Engineering Subontologies. ODEM also specifies the OASys packages to be used in each of the phases of the engineering process (Bermejo-Alonso & Sanz, 2011).

2.3 Robotic cloud ontology

The ROCO is based on the robot architecture ontology (ROA) (Olszewska *et al.*, 2017). The ROA ontology, developed by the IEEE Robotics and Automation Society (RAS) AuR Study Group (Fiorini *et al.*, 2017) as part of the IEEE-RAS ontology standard for AuR (Bermejo-Alonso *et al.*, 2018), defines the main concepts and relations regarding robot architecture for autonomous systems and inherits from SUMO/CORA ontologies (Prestes *et al.*, 2013). In particular, essential concepts for effective intra-robot communication (Doriya *et al.*, 2015) such as behavior, function, goal, and task have been defined in ROA (Olszewska *et al.*, 2017). Thence, ROCO ontological framework intends to apply the ROA ontology to CRS by extending ROA concepts to the CR domain (Figure 2), in order to enhance information sharing in CRS.

The ROCO ontology has been implemented in the web ontology language (OWL) (W3C OWL Working Group, 2009), which is well-established for knowledge-based applications (Olszewska *et al.*, 2017; Beetz *et al.*, 2018), using the Protégé tool and following the robotic ontological standard development life cycle (ROSADev) methodological approach (Olszewska *et al.*, 2018).

3 Cloud robotic case study

3.1 Cloud robotic context

The studied system is an interactive, heterogeneous, decentralized multi-robot system (Calzado *et al.*, 2018) that is composed of a set of unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs) that belong to different organizations, such as Regional Police, Federal Police, Firefighters, Civilian Defense, and Metropolitan Guard.

These organizations have their own UAVs and/or UGVs to support them individually in different application scenarios. These robots can share data inside their organizations through private cloud services. The robots used by these different organizations could have different characteristics, depending on the organizations' needs. For instance, multi-rotor UAVs, such as quadcopters or hexacopters, can offer Metropolitan Guard surveillance of specific buildings or public areas, while UGVs from the Federal Police can access places, which are potentially under threat by terrorist attacks, in order to disarm bombs.

The convergence of efforts from different law enforcement organizations can better assist the population against the threats posed by criminals and outlaws. The previously mentioned UAVs and UGVs can form a robotic cloud (Mell & Grance, 2011) to share data about the environment under concern, to consume data, to use resources to perform computation intensive tasks, and to offer their resources to other members of this cloud. As different organizations are sharing the same robotic cloud, it is possible to state that this is a CRS.

A service that the cloud can provide to the UAVs and UGVs composing this CRS is motion planning. As stated above, there are different types of UAVs and UGVs (e.g., fixed-wing, multi-rotors of different types, aerostats, ground robots with or without manipulators, etc.). Depending on the environment data that are uploaded to the cloud, different forms of event assistance services are needed in different areas. Whatever the required actions to be performed, obstacles may hinder the usage of given UAV or UGV to perform a given task or make one more appropriate than another. Upon the decision about which UAV or UGV should perform a given task, there is a need to execute its motion planning, in order the robot can move to the area in which it is needed. Moreover, it is also important to specify the tasks which have to be performed (Bruckner *et al.*, 2012).

For this heterogeneous system composed of different types of UAVs and UGVs with different capabilities, different sensors, and different actuators, semantic coherent definitions of the involved concepts have to be provided in order to ensure efficient communication in between the agents (Olszewska, 2017). Indeed, the semantic coherence of these definitions is important so that the required tasks can be assigned to any UAV or UGV that may be able to perform them, regardless its specific type. In particular, the CR-based system, which owns the need for clear semantic processing of tasks allocation in context of this surveillance application (as described in Section 3.2), can benefit from a standard ontology (as illustrated in Sections 3.3–3.4).

3.2 Cloud robotic scenario

As a real-world example of this overall CRC system architecture, we consider that two UGVs (i.e., a wheeled robot with a surveillance camera and a gripper (UGV-1) and a wheeled robot with only a surveillance camera (UGV-2)) are used for surveillance by Neighborhood Patrol Division of the Metropolitan Guard, and that two UAVs (i.e., a conventional helicopter (UAV-1) and a quadrotor (UAV-2)) are used for patrolling operations by the Highway Patrol Division of the Regional Police. All these vehicles use cloud computing facilities to process data and/or computation intensive algorithms such as SLAM/cooperative SLAM, sampling-based planning, or planning under uncertainty, since it is much faster than relying on their own onboard computers.

On the other hand, all these UGVs and UAVs evolve in an environment, as illustrated in Figure 3, that is, with nine volumes of interest (VOIs). These VOIs consist of six areas of interest (AOIs), two highways (HWs), and a base station (BS). In particular, AOI-1 is a rural area, AOI-2 is an industrial zone, AOI-3 is a sport complex, AOI-4 is a residential area, AOI-5 is a service area, and AOI-6 represents a gas station and its surroundings.

To perform daily routine surveillance and patrolling operations, it is required in the scenario to deploy these vehicles starting from the BS of the Regional Police and then to bring them back to this BS. It is assumed that the VOI of each vehicle is pre-assigned as follows: AOI-1, AOI-6, and HW-1 are assigned to UAV-1; AOI-4, AOI-5, and HW-2 are assigned to UAV-2; AOI-2 is assigned to UGV-1; and AOI-3 is assigned to UGV-2.

Part of this deployment mission is to plan the motion of each vehicle from the BS to a certain vantage point in the pre-assigned VOI and vice versa. In this planning problem, the goal is to determine a future

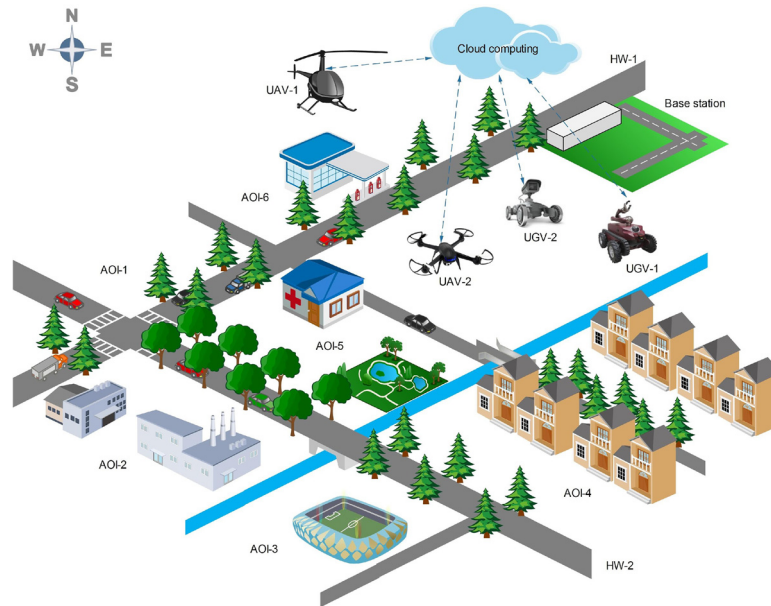


Figure 3 Cloud robotic scenario environment with unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs)

course of actions/activities for an executing entity (UAV/UGV) to drive it from an initial state (BS) to a specified goal state (VOI). It is supposed that at the beginning of the scenario, all UAVs and UGVs are in the BS, and all AOIs and HWs are not under surveillance/patrolling. Besides, it is assumed that the CRS system must be able to provide ground robots with a safe manner to reach their destination AOIs, as they are not able to have the same overview as the UAVs. Indeed, UGV-1 has to go through AOI-5 and HW-2 to reach AOI-2, while UGV-2 has to go through AOI-4 and HW-2 to reach AOI-3.

Considering the cooperative manner a CRS operates, UGV-1 and UGV-2 can request to the cloud a service for clearance about these places that they have to go through, in order they are able to make a safe move towards their destination AOIs. Thence, this request for clearance triggers a query to all robots connected to the cloud, and thus, a request capability match takes place. This matching has to consider both the necessary capabilities related to surveillance areas and the situation or context. In this case, it consists in matches between what the robots are able to do and about where the information is required (i.e., AOI-4, AOI-5, and HW-2).

According to what is stated in the presented scenario, UAV-2 has the necessary capabilities to provide the requested information, and for this, it must behave in a way to perform the actions of moving to AOI-4, collecting an image of AOI-4, then moving to AOI-5, capturing an image of this area and next, moving to HW-2, and collecting an image of this place. Afterwards, it has to share within the cloud structure the requested data which will be used by the robots that demanded the clearance over the areas.

3.3 *Ontology for autonomous systems ontology implementation*

The case study presented in Section 3.2 is analyzed in this subsection focusing on the application of the proposed OASys ontological framework (Section 2.2) in a cloud-based robotic system that has the goal to provide law enforcement services (Section 3.1).

It is worth noting that ontological terms are written in this section as usual to ease the reading process, whereas they are named as a unique word capitalizing each word within it as made in the naming convention in OASys. Italics are used for the case study scenario components specified in Section 3.2.

Given the fact that the proposed case study is a multi-agent system scenario, it is possible to use multi-agent planning Language or MA-PDDL (multi-agent planning domain definition language) as extensions of the PDDL (McDermott *et al.*, 1998) in multi-agent scenarios. MA-PDDL (Kovacs, 2012) provides

the ability to represent different actions of different agents or different capabilities, different goals of the agents and actions with interacting effects between the agents such as cooperation, joint actions, and constructive synergy.

For the sake of simplicity, it is assumed that there are no cooperative behaviors/joint actions between the vehicles in the proposed scenario, and then, PDDL can be used. Objects, predicates, initial state, goal specification, and actions/operators are the main components of a PDDL planning task. Indeed, objects represent things in the world that interest us. Predicates are properties of objects that are of interest to be studied; they can be true or false. Initial state is the state of the world that the system starts in. Goal specification represents things that is desirable to be true, and finally actions/operators are ways of changing the state of the world or capabilities.

In the proposed scenario, it is possible to extract the following examples of these five components:

1. **Objects:** The two UAVs, the two UGVs, the nine VOIs. (At lower level of abstraction for each vehicle, it is possible to consider planning entity, planned/execution entity, monitoring entity, motion planning solver, and planning environments as objects.)
2. **Predicates:** Is VOI fully observable? Is VOI structured? Is VOI adversarial? Is VOI under surveillance by UAV/UGV? Is HW patrolled by UAV? Is UGV-2 within AOI-5?
3. **Initial state:** All UAVs and UGVs are in the BS. All AOI and HWs are not under surveillance/patrolling.
4. **Goal specification:** UAV-1 is in AOI-1/AOI-6/HW-1; UAV-2 is in AOI-2/AOI-3/HW-2; AGV-1 is within AOI-4, and AGV-2 is within AOI-5.
5. **Actions and operators:** Each vehicle can execute same/different motion plan shared via the cloud to move from the BS to the AOI/HWs and vice versa. It can avoid stationary and/or moving obstacles while executing the plan, or it can continuously monitor the execution of the plan and it can adapt/repair the plan to resolve conflicts, if any.

For example, we assume the plan execution case as follows:

1. **Objects:** VOI: AOI-1, AOI-2, AOI-3, AOI-4, AOI-5, AOI-6, HW-1, HW-2, and Base-Station
Vehicles: UGV-1, UGV-2, UAV-1, UAV-2

In PDDL:

AOI-6,

1 (:objects AOI-1, AOI-2, AOI-3, AOI-4, AOI-5, AOI-6,

2) UGV-1, UGV-2, UAV-1, UAV-2)

2. **Predicates:** VOI(x) true iff x is a AOI/HW

Vehicle(x) true iff x is a UGV/UAV

at-Base-Veh(x) true iff x is a Base-Station and a Vehicle is in x

at-VOI-Vehicle(x,y) true iff x is a VOI and VOI Base-Station and a Vehicle y is in x

In PDDL:

(:predicates (VOI ?x) (Vehicle ?x)

(at-Base-Veh ?x) (at-AOI- UGV ?x)

(at-AOI- UAV ?x) (at-HW- UAV ?x))

3. **Initial state:** VOI(Base-Station) is true. Vehicle(UGV-1), Vehicle(UGV-2), Vehicle(UAV-1), Vehicle(UAV-2) are true. at-Base-Veh(UGV-1), at-Base-Veh(UGV-2), at-Base- Veh(UAV-1), at-Base-Veh(UAV-1) are true. Everything else is false.

In PDDL:

(:init (VOI Base-Station) (VOI AOI-1) (VOI AOI-2) (VOI AOI-3) (VOI AOI-4) (VOI AOI-5)

(VOI AOI-6) (VOI HW-1) (VOI HW-2) (Vehicle UGV-1), (Vehicle UGV-2), (Vehicle UAV-1),

(Vehicle UAV-2) (at-Base-Veh UGV-1), (at-Base-Veh UGV-2), (at-Base-Veh UAV-1), (at-Base-Veh

UAV-2))

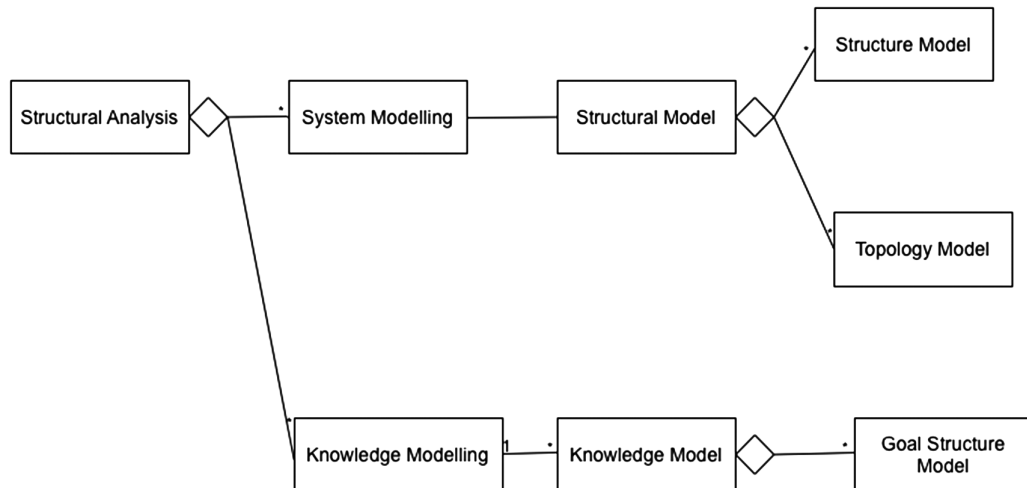


Figure 4 Structural analysis task defined in ASys engineering subontology

4. **Goal specification:** at-VOI-Vehicle(AOI-1, UAV-1), at-VOI-Vehicle(AOI-6, UAV-1), at-VOI-Vehicle(HW-1, UAV-1), at-VOI-Vehicle(AOI-2, UAV-2), at-VOI-Vehicle(AOI-3, UAV-2), at-VOI-Vehicle(HW-2, UAV-2), at-VOI-Vehicle(AOI-4, UGV-1), at-VOI-Vehicle(AOI-5, UGV-2). Everything else we do not care about.

In PDDL:

```

(:goal (and (at-VOI-Vehicle AOI-1 UAV-1)
(at-VOI-Vehicle AOI-1 UAV-1)
(at-VOI-Vehicle AOI-6 UAV-1)
(at-VOI-Vehicle HW-1 UAV-1)
(at-VOI-Vehicle AOI-2 UAV-2)
(at-VOI-Vehicle AOI-3 UAV-2)
(at-VOI-Vehicle HW-1 UAV-2)
(at-VOI-Vehicle AOI-4 UGV-1)
(at-VOI-Vehicle AOI-5 UGV-2)))
  
```

5. **Action and operators:**

Description: The vehicle can execute motion plan available on the cloud to move from the BS and between the AOI/HWs, can avoid stationary and/or moving obstacles, can monitor the execution of the plan, and can adapt/repair the plan to resolve conflicts. For the sake of simplicity, assume that a vehicle y moves from a to b .

Precondition: VOI(a), VOI(b), and at-VOI-Vehicle(a,y) are true.

Effect: at-VOI-Vehicle(b,y) becomes true. at-VOI-Vehicle(a,y) becomes false. Everything else does not change.

In PDDL:

```

(:action move :parameters (?a ?b)
:precondition (and (VOI ?a) (VOI ?b)
(at-VOI-Vehicle ?a))
:effect (and (at-VOI-Vehicle ?b)
(not (at-VOI-Vehicle ?a))))
  
```

Considering the OASys-driven ODEM, its analysis phase involves different tasks performed for the structural, behavioral, and functional analysis of the system. As such, the structural analysis task models the system structure and knowledge (Figure 4). The system modeling subtask models the structural elements in the system and their relationships, obtaining a structural model, based on the ontological elements in the system subontology. OASys allows to model the abstract knowledge to be used in the

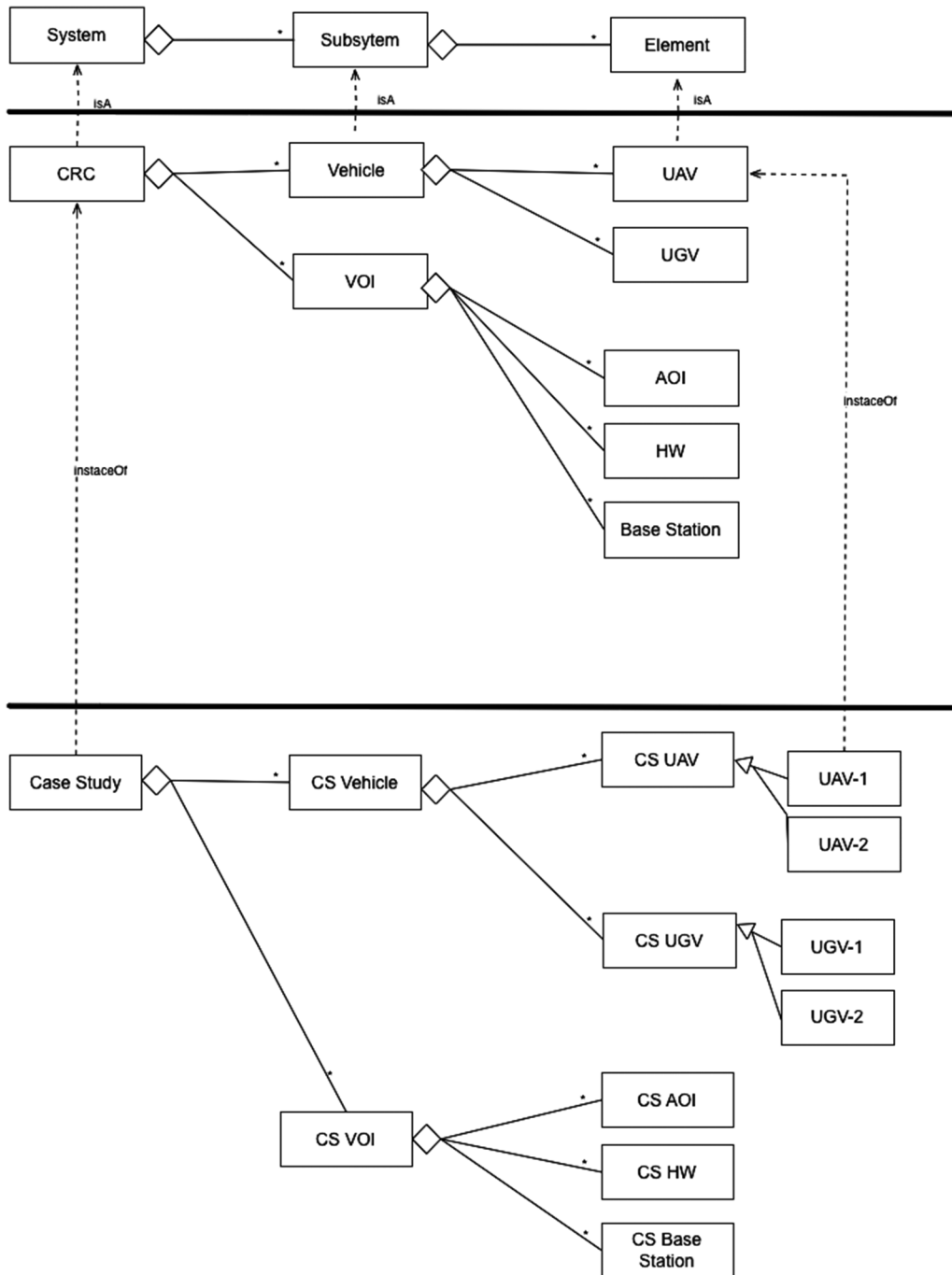


Figure 5 Structure model specializing concepts for the case study

system. Later on, the motion planning process uses the instantiated knowledge for the particular system under specification.

In this sense, Figure 5 shows the structure model specializing OASys system subontology abstract knowledge classes (system, subsystem, and element) and their composition relationships using a UML class diagram notation. The bold horizontal lines distinguish the different level of abstraction, that is, any system concepts at the upper level, the CRC system concepts at the intermediate level as specialization through isA relationships, and the actual Objects in the Case Study as instances of the previous classes through instanceOf relationships (not all shown to avoid cluttering the diagram). This structure model is complemented with the topology models to show the topological and mereological relationships. These

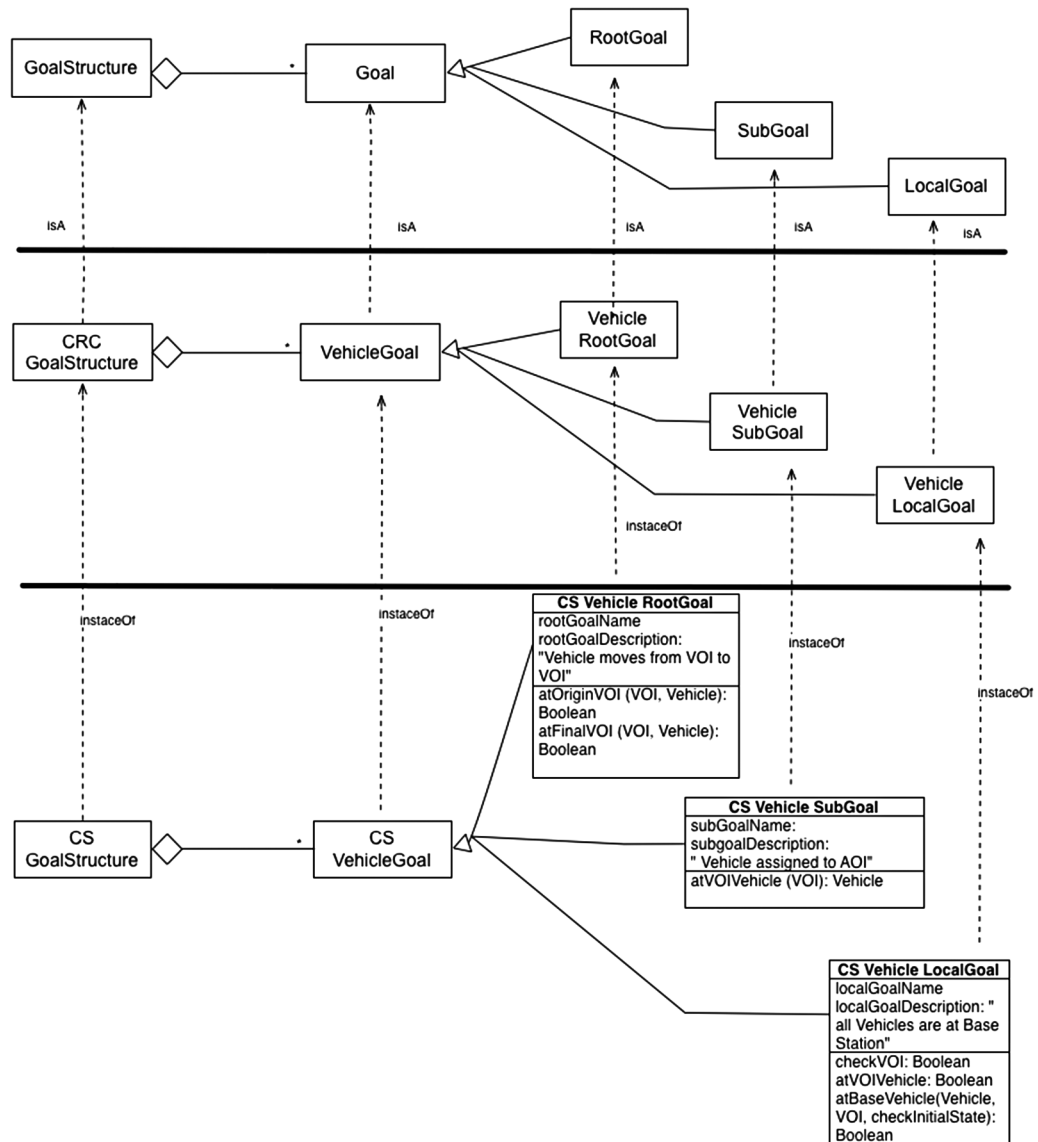


Figure 6 Goal structure model as result of knowledge modeling task

models allow representing the predicates specifying which specific vehicle surveys, patrols, or is within the limits of a concrete VOI.

The role played by this ontological-based structural model is twofold. Firstly, it allows the different elements in the CRC system to know about the different, involved vehicles and VOIs and the topological as well as mereological relationships among them for a further reasoning on the motion planning tasks. Secondly, the models can also be used as a semantic system specification for a Model-based Systems Engineering process.

In a similar way, the knowledge modeling subtask obtains a knowledge model to be used by the system. In this case, the main knowledge to model is the planning-related concepts about initial state and the goal specification, in a goal structure model. Figure 6 shows how the abstract knowledge is conceptualized in OASys. As illustrated in this figure, concepts such as Goal, RootGoal, SubGoal, and LocalGoal are specialized for the CRC system at the intermediate level, while UML classes are used to specify generic vehicle goals. Later on, the actual objects for the case study become instances of the former classes, as shown in the UML object diagram, with attributes and operations to define the case study's specific goals to be fulfilled by the vehicle (i.e., AGVs and UAVs) in relation to the VOI (i.e., AOI, HW, and BS). This semantic model constitutes the first kind of knowledge for the CRC system to

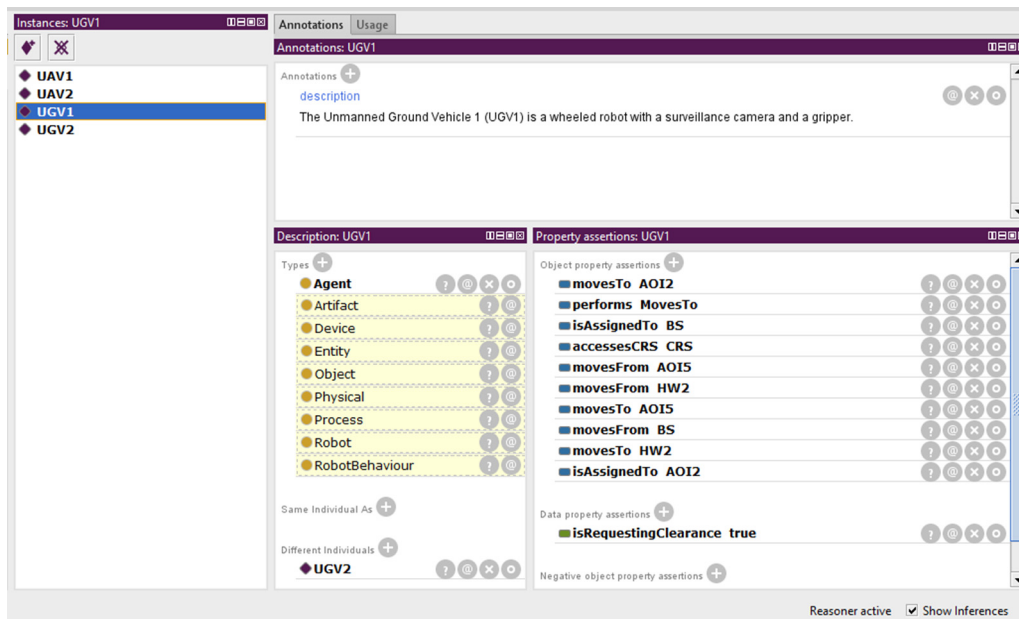


Figure 7 Snapshot of the UGV1 instance of *Agent*. Yellow rows are automatically inferred by the reasoner. Best viewed in color

make decisions during the motion planning reasoning. Other conceptual models using ODEM (Bermejo-Alonso *et al.*, 2016) can be obtained in a similar way in order to specify the system behavior in terms of the motion planning actions (e.g., avoid obstacles, resolve conflicts, etc.), as shown in different UML activity and sequence diagrams by specializing OASys ontological concepts and relationships.

Hence, the first qualitative results of OASys have been presented in Figures 4–6 in context of this CRS surveillance application. As a first attempt to characterize the CRS domain, it provides enough support to conceptualize this case study. Currently, OASys is under revision to be implemented using OWL and reasoning mechanism to enhance both the characterization of the robotic systems (Balakirsky *et al.*, 2017) and the metacontrol features (Hernandez *et al.*, 2018). In particular, OASys is considered for the standardization efforts under development by the IEEE P1872.1 working group focused on RTR.

3.4 Robotic cloud ontology implementation

The ROCO ontological framework introduced in Section 2.3 represents robots' actions, functions, behaviors, and capabilities using the standard OWL language as demonstrated in context of the scenario described in Section 3.2.

The actual objects present in the case study are instances of the proposed CRS ontological concepts and are related by the defined axioms.

Particularly, the agents UGV1 (Figure 7) and UGV2 (Figure 8), which are both requiring clearance from UAV2 (Figure 9), are considered, since they navigate through the same VOIs; all these agents are being part of the same CRS, as implemented in OWL (Figure 10) and displayed in Figure 11. Therefore, UGV1 and UGV2 could use the CRS as a resource to request clearance. This has been implemented in OWL by means of the *isRequestingClearance* property, as illustrated in Figure 12.

Hence, the first qualitative results of ROCO have been presented in Figures 7–15 in context of this CRS surveillance application. To sum up, the role played by this ontological-based CR model is twofold. Firstly, it allows the different elements in the CRS to know which agents (i.e., vehicles) are involved in and what are their positions (i.e., VOIs) as well as the topological and mereological relationships among them, leading to further reasoning on their motion planning (Figures 13–15). On the other hand, this approach is useful to establish a common shared vocabulary to enhance CRS agents' communication. Thereupon, ROCO is considered for the standardization efforts under development by the IEEE P1872.2 working group focused on AuR.

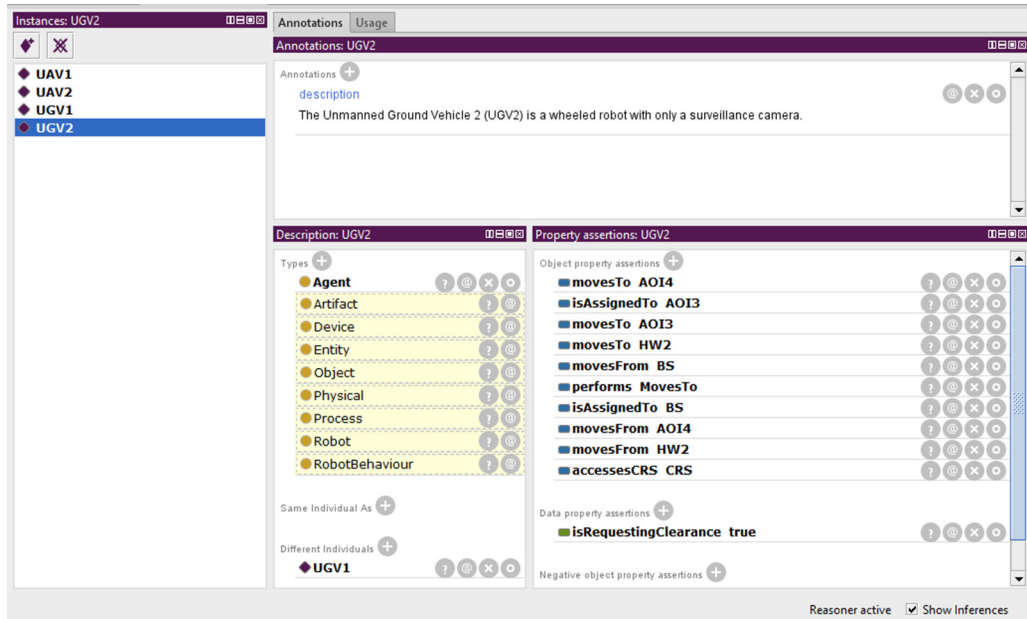


Figure 8 Snapshot of the UGV2 instance of Agent. Yellow rows are automatically inferred by the reasoner. Best viewed in color

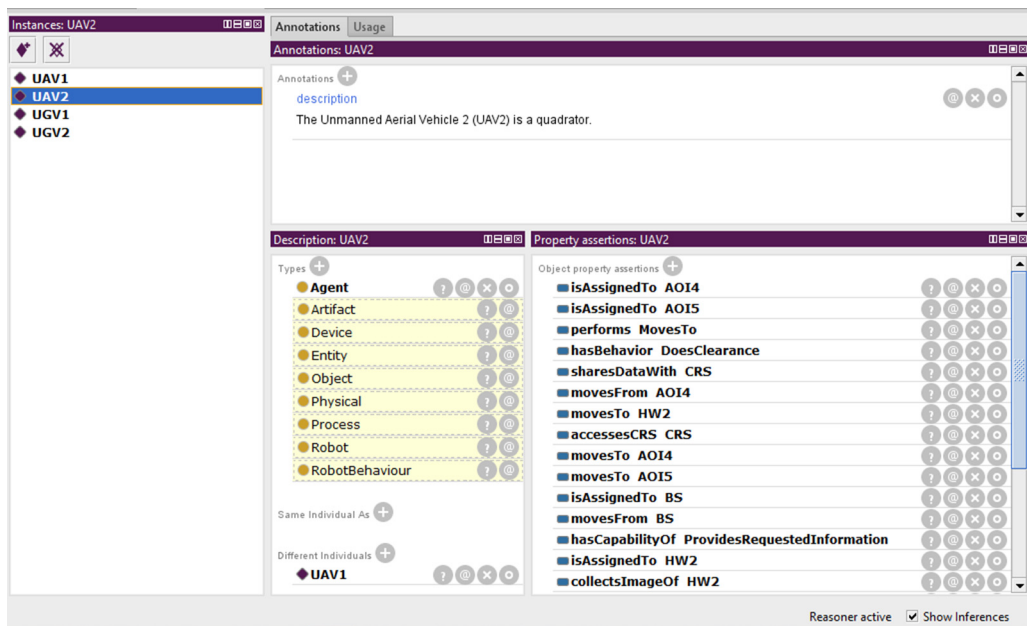


Figure 9 Snapshot of the UAV2 instance of Agent. Yellow rows are automatically inferred by the reasoner. Best viewed in color

```

OWL functional syntax rendering:
AnnotationAssertion(<http://purl.org/dc/elements/1.1/description> :AOI1 "Area Of Interest 1 (AOI1) is a rural area.")
ClassAssertion(:VolumeOfInterest :AOI1)
AnnotationAssertion(<http://purl.org/dc/elements/1.1/description> :AOI2 "Area Of Interest 2 (AOI2) is an industrial zone.")
ClassAssertion(:VolumeOfInterest :AOI2)
AnnotationAssertion(<http://purl.org/dc/elements/1.1/description> :AOI3 "Area Of Interest 3 (AOI3) is a sport complex.")
ClassAssertion(:VolumeOfInterest :AOI3)
AnnotationAssertion(<http://purl.org/dc/elements/1.1/description> :AOI4 "Area Of Interest 4 (AOI4) is a residential area.")
ClassAssertion(:VolumeOfInterest :AOI4)
AnnotationAssertion(<http://purl.org/dc/elements/1.1/description> :AOI5 "Area Of Interest 5 (AOI5) is a service area.")
ClassAssertion(:VolumeOfInterest :AOI5)
AnnotationAssertion(<http://purl.org/dc/elements/1.1/description> :AOI6 "Area Of Interest (AOI6) represents a gas station and its surroundings.")
ClassAssertion(:VolumeOfInterest :AOI6)
AnnotationAssertion(rdfs:comment :BS "All the agents starts initially from this VOL.")
AnnotationAssertion(<http://purl.org/dc/elements/1.1/description> :BS "Base Station of the Regional Police (BS)")
ClassAssertion(:VolumeOfInterest :BS)
    
```

Figure 10 Snapshot of the CRS instances implemented in OWL

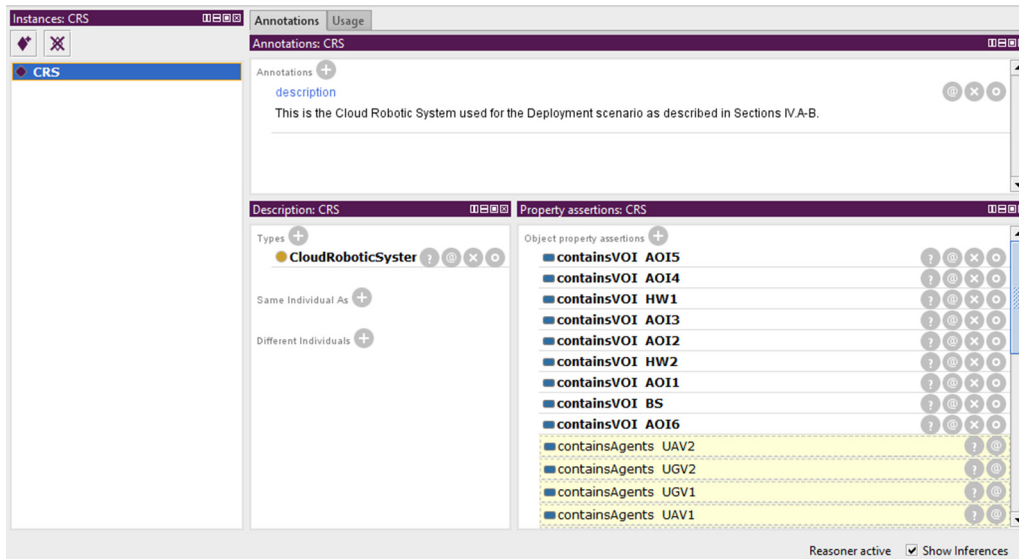


Figure 11 Snapshot of the CRS instance of *CloudRobotic System*. Yellow rows are automatically inferred by the reasoner. Best viewed in color

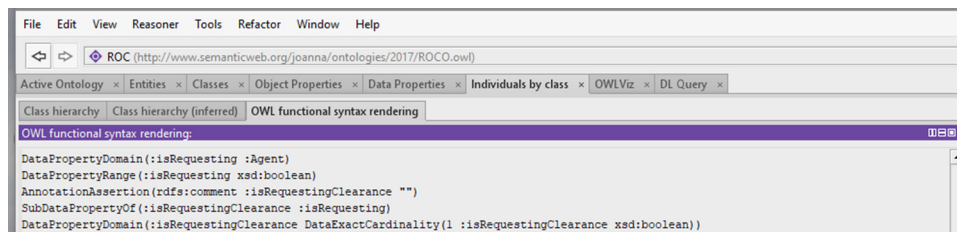


Figure 12 Snapshot of the *isRequestingClearance* property implemented in OWL

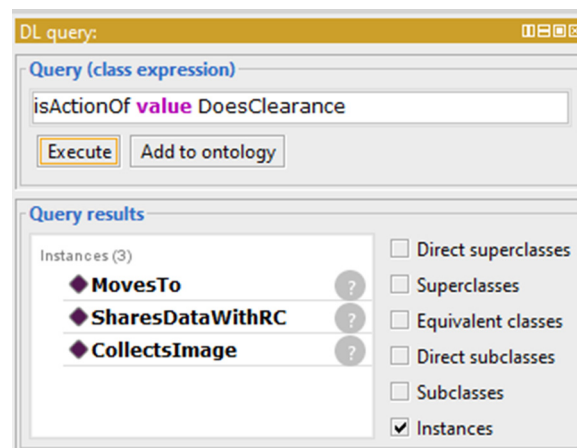


Figure 13 Snapshot of the reasoning on the *Action* concept in context of the CRS case study

3.5 Discussion

OASys is an ontology that comprises both autonomous systems description and its engineering process as mentioned in Section 2.2. The upper level regarding a general system is well founded, based on former standard and proved theories. Thence, the autonomous system level, both from a description and an engineering view, would benefit of the current standardization efforts by IEEE, for example, in its P1872.1 Task Representation Ontology project. Indeed, the action package in the ASys Subontology of OASys is

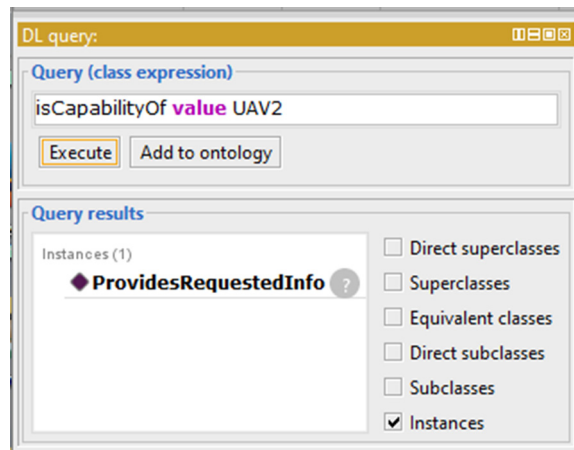


Figure 14 Snapshot of the reasoning on the *Capability* concept in context of the CRS case study

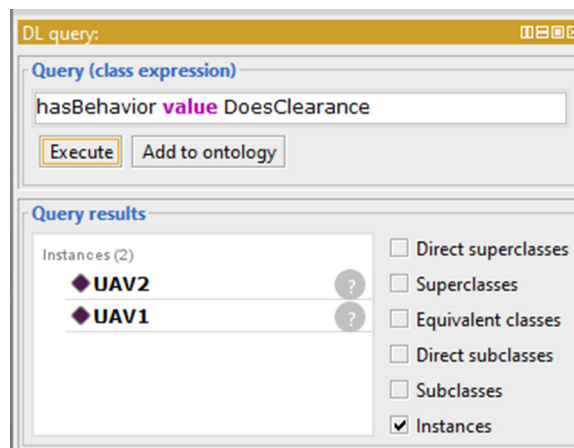


Figure 15 Snapshot of the reasoning on the *Behavior* concept in context of the CRS case study

an initial attempt to characterize goals, tasks, and functions in autonomous systems, which are aligned with the focus of the IEEE P1872.1 under-progress standard and which are useful for CRS.

The ROCO ontology in turn follows the METHONTOLOGY ontological methodology approach to characterize autonomous systems, and in particular to contribute to the IEEE P1872.2 ontology under development by the Autonomous Systems Working Group, as pointed out in Section 2.3. Thus, ROCO inherits from the autonomous ROA which was developed in the context of the IEEE P1782.2 Autonomous Robotic effort, while it has a greater focus on CR. Indeed, ROCO considers concepts related to behavior, function, task, etc. in order to specifically formalize the CR domain.

Hence, both ontologies contribute towards the current standardization efforts by the IEEE, regarding robotics and autonomous systems, and in particular CRS specific concepts and relations. It is worth noting that OASys was developed using a systems engineering language called UML, which is considered to be a lightweight language without an embedded reasoning mechanism. Consequently, the reasoning on OASys has to be processed externally, for example, on a PDDL platform as explained in Section 3.3. Unlike OASys, the ROCO ontology has been directly developed in OWL, which is a computational logic-based language such that knowledge expressed in OWL can be exploited by computer programs and automated reasoners, for example, to verify the consistency of that knowledge or to make implicit knowledge explicit, as described in Section 3.4.

Therefore, OASys ontological engineering has served as a general systems approach and a common taxonomic basis for the IEEE P1872 ontologies in development, while the ROCO ontology has allowed

to specialize the standardized ontologies to include CR-specific concepts and relations which automated reasoning can be directly performed on.

4 Conclusions

CRS alloy cloud computing and autonomous multi-robot systems. These technologies require an adequate understanding and a coherent sharing of the resources and data among the CRS agents in order to make possible an interoperable communication and efficient collaboration between the CRS stakeholders.

For this purpose, ontologies have been identified as a possible solution. Thence, this paper presents the current state of standard ontologies for the CR domain. Moreover, a case study showcases the implementation of the described ontologies, namely the OASys and the ROCO, into a real-world motion planning CRS scenario, highlighting the main features of these ontologies in the CR context and reporting the results obtained when using the defined ontologies to enhance the interoperability of CRS.

Future works aim to enrich the vocabulary developed so far in the existing standard ontologies and create further semantic extensions that will contribute to the ontological standard which is being developed by the IEEE Ontologies for Robotics and Automation (ORA) Working Group and its RTR and AuR subgroups.

References

- Arumugam, R., Enti, V. R., Bingbing, L., Xiaojun, W., Baskaran, K., Kong, F. F., Kumar, A. S., Meng, K. D. & Kit, G. W. 2010. DAVinCi: a cloud computing framework for service robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 3084–3089.
- Balakirsky, S., Kootbally, Z., Schlenoff, C., Kramer, T. & Gupta, S. 2012. An industrial robotic knowledge representation for kit building applications. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1365–1370.
- Balakirsky, S., Schlenoff, C., Fiorini, S., Redfield, S., Barreto, M., Nakawala, H., Carbonera, J., Soldatova, L., Bermejo-Alonso, J., Maikore, F., Goncalves, P., De Momi, E., Sampath Kumar, V. & Haidegger, T. 2017. Towards a robot task ontology standard. In *Proceedings of the ASME International Manufacturing Science and Engineering Conference*, 1–10.
- Beetz, M., Bessler, D., Haidu, A., Pomarlan, M., Bozcuoglu, A. K. & Bartels, G. 2018. KnowRob 2.0: a 2nd generation knowledge processing framework for cognition-enabled robotic agents. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 512–519.
- Benavidez, P., Muppidi, M., Rad, P., Prevost, J. J., Jamshidi, M. & Brown, L. 2015. Cloud-based realtime robotic visual SLAM. In *Proceedings of the IEEE Annual IEEE Systems Conference*, 773–777.
- Bermejo-Alonso, J., Chibani, A., Goncalves, P., Li, H., Jordan, S., Olivares, A., Olszewska, J. I., Prestes, E., Fiorini, S. R. & Sanz, R. 2018. Collaboratively working towards ontology-based standards for robotics and automation. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*.
- Bermejo-Alonso, J., Hernandez, C. & Sanz, R. 2016. Model-based engineering of autonomous systems using ontologies and metamodels. In *Proceedings of the IEEE International Symposium on Systems Engineering*, 1–8.
- Bermejo-Alonso, J. & Sanz, R. 2011. An ontological framework for autonomous systems modelling. *International Journal On Advances in Intelligent Systems* **30**(3–4), 211–225.
- Bruckner, D., Picus, C., Velik, R., Herzner, W. & Zucker, G. 2012. Hierarchical semantic processing architecture for smart sensors in surveillance networks. *IEEE Transactions on Industrial Informatics* **80**(2), 291–301.
- Calzado, J., Lindsay, A., Chen, C., Samuels, G. & Olszewska, J. I. 2018. SAMI: interactive, multi-sense robot architecture. In *Proceedings of the IEEE International Conference on Intelligent Engineering Systems*, 317–322.
- Chen, C.-L., Li, Y.-T., Deng, Y.-Y. and Li, C.-T. 2018. Robot identification and authentication in a robot cloud service system. *IEEE Access* **6**, 0 56488–56503.
- Dogmus, Z., Erdem, E. & Patoglu, V. 2015. RehabRobo-Onto: design, development and maintenance of a rehabilitation robotics ontology on the cloud. *Robotics and Computer-Integrated Manufacturing* **33**, 100–109.
- Doriya, R., Chakraborty, P. & Nandi, G. C. 2012. Robotic services in cloud computing paradigm. In *Proceedings of the IEEE International Symposium on Cloud and Services Computing*, 80–83.
- Doriya, R., Mishra, S. & Gupta, S. 2015. A brief survey and analysis of multi-robot communication and coordination. In *Proceedings of the IEEE International Conference on Computing, Communication and Automation*, 1014–1021.
- Du, Z., Yang, W., Chen, Y., Sun, X., Wang, X. & Xu, C. 2011. Design of a robot cloud center. In *Proceedings of the IEEE International Symposium on Autonomous Decentralized Systems*, 269–275.

- Ermacora, G., Toma, A., Bona, B., Chiaberge, M., Silvagni, M., Gaspardone, M. & Antonini, R. 2013. A cloud robotics architecture for an emergency management and monitoring service in a smart city environment. In *IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS)*.
- Fiorini, S. R., Bermejo-Alonso, J., Goncalves, P., Pignaton de Freitas, E., Olivares Alarcos, A., Olszewska, J. I., Prestes, E., Schlenoff, C., Ragavan, S. V., Redfield, S., Spencer, B. & Li, H. 2017. A suite of ontologies for robotics and automation. *IEEE Robotics and Automation Magazine* **24**(1), 8–11.
- Guizzo, E. 2011. Robots with their heads in the clouds. *IEEE Spectrum* **48**(3), 16–18.
- Hernandez, C., Bermejo-Alonso, J. & Sanz, R. 2018. A self-adaptation framework based on functional knowledge for augmented autonomy in robots. *Integrated Computer-Aided Engineering* **25**(2), 157–172.
- Hu, G., Tay, W. P. & Wen, Y. 2012. Cloud robotics: architecture, challenges and applications. *IEEE Network* **26**(3), 21–28.
- Huang, J.-Y., Lee, W.-P. & Lin, T.-A. 2019. Developing context-aware dialoguing services for a cloud-based robotic system. *IEEE Access* **7**, 44293–44306.
- Hunziker, D., Gajamohan, M., Waibel, M. & D’Andrea, R. 2013. Rapyuta: the RoboEarth cloud engine. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 438–444.
- Jangid, N. & Sharma, B. 2016. Cloud computing and robotics for disaster management. In *Proceedings of the IEEE International Conference on Intelligent Systems, Modelling and Simulation*, 20–24.
- Jordan, S., Haidegger, T., Kovacs, L., Felde, I. & Rudas, I. 2013. The rising prospects of cloud robotic applications. In *Proceedings of the IEEE International Conference on Computational Cybernetics*, 327–332.
- Kamei, K., Nishio, S., Hagita, N. & Sato, M. 2012. Cloud networked robotics. *IEEE Network* **26**(3), 28–34.
- Kamei, K., Zanlungo, F., Kanda, T., Horikawa, Y., Miyashita, T. & Hagita, N. 2017. Cloud networked robotics for social robotic services extending robotic functional service standards to support autonomous mobility system in social environments. In *Proceedings of the IEEE International Conference on Ubiquitous Robots and Ambient Intelligence*, 897–902.
- Kehoe, B., Patil, S., Abbeel, P. & Goldberg, K. 2015. A survey of research on cloud robotics and automation. *IEEE Transactions on Automation Science and Engineering* **12**(2), 398–409.
- Kitamura, Y. & Mizoguchi, R. 2003. Ontology-based description of functional design knowledge and its use in a functional way server. *Expert Systems with Applications* **24**(2), 153–166.
- Kovacs, D. 2012. A multi-agent extension of PDDL3. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, 1–9.
- Kuffner, J. J. 2010. Cloud-enabled humanoid robots. In *IEEE-RAS International Conference on Humanoid Robots*.
- Lam, M.-L. & Lam, K.-Y. 2014. Path planning as a service (PPaaS): cloud-based robotic path planning. In *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 1839–1844.
- McDermott, D., Ghallab, M., Howe, A., Knoblock, C., Ram, A., Veloso, M., Weld, D. & Wilkins, D. 1998. *PDDL: The Planning Domain Definition Language*.
- Mell, P. & Grance, T. 2011. The NIST definition of cloud computing. NIST Special Publication 800–145. September 2011.
- Miratabzadeh, S. A., Gallardo, N., Gamez, N., Haradi, K., Puthussery, A. R., Rad, P. & Jamshidi, M. 2016. Cloud robotics: a software architecture for heterogeneous large-scale autonomous robots. In *Proceedings of the IEEE World Automation Congress*, 1–6.
- Mohanarajah, G., Hunziker, D., D’Andrea, R. & Waibel, M. 2015. Rapyuta: a cloud robotics platform. *IEEE Transactions on Automation Science and Engineering* **12**(2), 481–493.
- Muhayyuddin, A. A. & Rosell, J. 2015. Ontological physics-based motion planning for manipulation. In *Proceedings of the IEEE Conference on Emerging Technologies Factory Automation (ETFA)*, 1–7.
- Olszewska, J. I. 2017. Clock-model-assisted agent’s spatial navigation. In *Proceedings of the International Conference on Agents and Artificial Intelligence*, 687–692.
- Olszewska, J. I. & Allison, A. K. 2018. ODYSSEY: software development life cycle ontology. In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, 303–311.
- Olszewska, J. I., Barreto, M., Bermejo-Alonso, J., Carbonera, J., Chibani, A., Fiorini, S., Goncalves, P., Habib, M., Khamis, A., Olivares, A., Freitas, E. P., Prestes, E., Ragavan, S. V., Redfield, S., Sanz, R., Spencer, B. & Li, H. 2017. Ontology for autonomous robotics. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 189–194.
- Olszewska, J. I., Houghtaling, M., Goncalves, P., Haidegger, T., Fabiano, N., Carbonera, J. L., Fiorini, S. R. & Prestes, E. 2018. Robotic ontological standard development life cycle. In *IEEE International Conference on Robotics and Automation (ICRA)*.
- Olszewska, J. I., Houghtaling, M., Goncalves, P. J. S., Fabiano, N., Haidegger, T., Carbonera, J. L., Patterson, W. R., Ragavan, S. V., Fiorini, S. R. & Prestes, E. 2020. Robotic ontological standard development life cycle in action. *Journal of Intelligent and Robotic Systems* **98**(1), 119–131.
- Olszewska, J. I., Simpson, R. & McCluskey, T. L. 2014. Dynamic OWL ontology design using UML and BPMN. In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, 436–444.

- Prestes, E., Carbonera, J., Fiorini, S., Jorge, V., Abel, M., Madhavan, R., Locoro, A., Goncalves, P., Barreto, M., Habib, M., Chibani, A., Gerard, S., Amirat, Y. & Schlenoff, C. 2013. Towards a core ontology for robotics and automation. *Robotics and Autonomous Systems* **61**(11), 1193–1204.
- Quintas, J. M., Menezes, P. J. & Dias, J. M. 2011. Cloud robotics: toward context aware robotic networks. In *Proceedings of the IASTED International Conference on Robotics*, 420–427.
- Rahimi, R., Shao, C., Veeraraghavan, M., Fumagalli, A., Nicho, J., Meyer, J., Edwards, S., Flannigan, C. & Evans, P. 2017. An industrial robotics application with cloud computing and high-speed networking. In *Proceedings of the IEEE International Conference on Robotic Computing*, 44–51.
- Rahman, A., Jin, J., Cricenti, A. L., Rahman, A. & Kulkarni, A. 2019. Communication-aware cloud robotic task offloading with on-demand mobility for smart factory maintenance. *IEEE Transactions on Industrial Informatics* **15**(5), 2500–2511.
- Riazuelo, L., Tenorth, M., Di Marco, D., Salas, M., Galvez-Lopez, D., Moesenlechner, L., Kunze, L., Beetz, M., Tardos, J. D., Montano, L. & Martinez-Montiel, J. M. 2015. RoboEarth semantic mapping: a cloud enabled knowledge-based approach. *IEEE Transactions on Automation Science and Engineering* **12**(2), 432–443.
- Salmeron-Garcia, J., Inigo-Blasco, P., Diaz-del Rio, F. & Cagigas-Muniz, D. 2015. A tradeoff analysis of a cloud-based robot navigation assistant using stereo image processing. *IEEE Transactions on Automation Science and Engineering* **12**(2), 444–454.
- Sharath, B. S., Srisha, R., Shashidhar, K. V. & Bharadwaj, S. S. 2018. Intelligent and smart cloud based autonomous robotic kitchen system. In *Proceedings of the IEEE International Conference on Intelligent Computing and Control Systems*, 601–606.
- Soldatova, L. N., Clare, A., Sparkes, A. & King, R. D. 2006. An ontology for a robot scientist. *Bioinformatics* **22**(14), e464–e471.
- Tenorth, M., Kamei, K., Satake, S., Miyashita, T. & Hagita, N. 2013. Building knowledge-enabled cloud robotics applications using the ubiquitous network robot platform. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5716–5721.
- Turnbull, L. & Samanta, B. 2013. Cloud robotics: formation control of a multi-robot system utilizing cloud infrastructure. In *Proceedings of IEEE Southeastcon*, 1–4.
- Vick, A., Vonasek, V., Penicka, R. & Kruger, J. 2015. Robot control as a service - towards cloud-based motion planning and control for industrial robots. In *Proceedings of the IEEE International Workshop on Robot Motion and Control*, 33–39.
- W3C OWL Working Group. 2009. OWL2: Web Ontology Language Document Overview.
- Waibel, M., Beetz, M., Civera, J., D'Andrea, R., Elfring, J., Galvez-Lopez, D., Haussermann, K., Janssen, R., Montiel, J. M. M., Perzylo, A., Schiessle, B., Tenorth, M., Zweigle, O. & van de Molengraft, R. 2011. Roboearth. *IEEE Robotics Automation Magazine* **18**(2), 69–82.
- Wang, S., Krishnamachari, B. & Ayanian, N. 2015. The optimism principle: a unified framework for optimal robotic network deployment in an unknown obstructed environment. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2578–2584.
- Zhang, H. & Zhang, L. 2019. Cloud robotics architecture: trends and challenges. In *Proceedings of the IEEE International Conference on Service-Oriented System Engineering*, 362–367.
- Zhang, Y., Li, L., Nicho, J., Ripperger, M., Fumagalli, A. & Veeraraghavan, M. 2019. Gilbreth 2.0: an industrial cloud robotics pick-and-sort application. In *Proceedings of the IEEE International Conference on Robotic Computing*, 38–45.