

Towards an Ontology for Task and Planning in Autonomous Systems: An Emergency Scenario

Julita Bermejo-Alonso, Jorge Salvador, and Ricardo Sanz

Abstract. Emergency scenarios where chemical explosions or fire out-breaks take place become dangerous environments to send a human team to. UAVs and UGVs working together with the human rescue team can prevent endangering the human team. To perform the emergency tasks, it is necessary to define what the UAV/UGV robotic system will do in the emergency area, detailing what each of the robots will specifically carry out. Next, the different actions should be planned to perform such tasks. An ontology-based approach is described in this paper, where the cohesive element to specify the planning process is a task ontology. We describe the initial contents of our ontology for task and planning in autonomous robots, with an example of its use within an emergency scenario where a combined UAV/UGV robotic system is supposed to act.

Keywords: Ontologies · Autonomous systems · UGV/UAV · Emergency scenario

1 Introduction

Search and rescue operations in emergency scenarios such as earthquakes, gas leaks, chemical explosions and fire outbreaks face some major challenges. The most important is the risk of a dangerous environment where a human rescue team needs to perform. It may be inconvenient, dangerous, or impossible to have a human operator or rescue team present in the emergency area. A set of UAVs and UGVs working together with the human rescue team can provide an alternative approach. An Unmanned Aerial Vehicle (UAV) is an aircraft that does not carry a human operator, and is capable of flying under remote control (teleoperation) or in an autonomous programming way. An Unmanned Ground Vehicle (UGV) is a vehicle that operates in the ground without an onboard human presence.

In some cases, complex tasks cannot be completed by just one type of unmanned robot. While a UAV can scan a wide area, searching for a wide range of targets, it may encounter some difficulty in exact target positioning. By contrast, UGVs have more accurate locations of a target on the ground, but

move slowly and can be easily blocked by obstacles. The UAV and the UGV should work cooperatively. Their combined use have provided greater levels of flexibility both in mobility and response time, as well as exploiting their different capabilities. A multi-robot system such as a combined UAV/UGV has become a common solution where procedure, algorithms and cooperative robotics have been analysed [1, 6, 16, 18, 26], and ontologies have been developed for urban search [24] or for human-robot collaboration mission contexts [7].

To perform the emergency tasks, it is necessary to define what the UAV/UGV robotic system will do in the emergency area, detailing what each of the robots and humans will do in particular. Next, the different actions should be planned to perform such tasks. Generally speaking, the inputs for the planning process consist of some kind of domain knowledge that specifies the original state and the goals. The output provided by the planner will be a plan or set of plans if a solution exists. This is a time-consuming process that requires to define previously all the elements involved in the task and planning process. An ontology that provides the concepts and relationships of the domain can ease the definition of the planning problem. Ontologies can be used to characterise and share the planning domain knowledge, as a knowledge-base to be reused across planning applications. Ontologies can also be the core on which the planning reasoner obtains the plan.

An ontology-based approach is described in this paper, where the cohesive element to specify the planning process is an ontology. In our case, the use of an emergency scenario serves as a mechanism to verify the suitability and definition of the ontological elements. The paper is organised as follows: Sect. 2 explains related research on the topic of ontologies for tasks and planning, Sect. 3 explains the rationale behind the conceptualisation as a set of requirements. Section 4 summarises the contents of the proposed ontology, Sect. 5 introduces the features of the specific emergency scenario where a multi-robot system or UAV/UGV is sent to, and details the use of the proposed ontology. Finally, Sect. 6 provides some concluding remarks, as well as future lines of research to address.

2 Related Research

A computational ontology defines a set of ontological elements as representational primitives that can be used to model a domain of knowledge [14, 15].

There have been different research efforts to characterise the planning domain concepts in natural language such as [3, 13, 21, 27, 29]. These works define ontologies as a vocabulary to define main concepts in the planning domain, such as *plan*, *environment*, *task*, etc. Their underlying idea is to provide a knowledge base for plans that can be reused among applications and users. In general, these ontologies are designed without a particular planning viewpoint in mind.

A second approach to planning ontologies is to formalise them using Description Logics (DL) [2], a family of knowledge representation languages used to formalise the knowledge of an application domain in a more structured manner. Some examples are described in [4, 10, 23, 25]. Within the DL approach, a review

of existing taxonomies of objects in the domain and how they have been used by different planning algorithms and reasoners is given in [12]. Using DL allows to organise the class description in a taxonomic hierarchy, useful for Hierarchical Task Network (HTN) approaches in planning, where most of these works are oriented to. Moreover, the ontologies formalised in DL can be used by a reasoner to infer new relationships among the concepts, to recognise instances of those concepts, to obtain new decomposition methods for their tasks, to evaluate pre-conditions to perform a task, or to update the state when the planner simulate the effects of operators.

Another set of ontologies for task and planning have been made using OWL [30]. Bouillet et al. [22] present a domain-independent, general purpose knowledge engineering and planning framework to construct planning domains and problems based on OWL ontologies. An OWL ontology for planning domains based on the HTN (Hierarchical Task Network) paradigm is described in [9]. Using OWL for ontologies is justified by these authors as OWL would allow to reuse available OWL ontologies with common and general concepts for planning, such as time and physical properties.

3 The Ontology Rationale

We are building an ontology that will be integrated in the Ontology for Autonomous Systems (OASys) [5], to describe and engineer the planning processes of autonomous systems. The ontology will be used by different stakeholders, from engineers to end-users. This section summarises a set of requirements to fulfill during the development and use of the ontology.

3.1 The Ontological Engineering Approach

To specify the concepts in the ontology, there are different ways to proceed (bottom-up, top-down or middle-out) [28]. A bottom-up approach usually provides a very high level of detail, which requires time and effort. A top-down one may lead to high-level concepts which would require later rework. A *middle-out approach* keeps the balance of the level of detail, as it defines the most fundamental terms to move later to more general and more specific ones. Following a middle-out approach, a set of initial concepts have been defined. In our ontology, they are those related to setting the ground for the planning problem. If we, as engineers, want to come up with a plan, it is because we would like the robotised system (including human agents) to carry out some tasks in a given environment with some purpose or goal. These tasks will be accomplished by a set of actions. As a result, the environment and the system itself will have changed the state where they were originally in. The planning problem takes place within a particular context that provides the initial assumptions about it.

3.2 The Domain and the Problem-Solving Knowledge

A common approach for task and planning is to consider both the *knowledge about the domain* and the *problem-solving knowledge* [8]. Our challenge is to integrate the planning algorithms and methods (i.e. problem-solving knowledge) with a rich representation of planning and task knowledge (i.e. domain-specific knowledge), making sure they use the *same terminology*. In our case, the knowledge about the domain refers to the concepts about the environment, the objects, and the tasks to be performed. The problem-solving knowledge is conceptualised in the ontology as the actual execution of the tasks by the actors, as actions that require the corresponding skills and resources. Additional concepts support the planning process in terms of limitations and evaluation of the methods and plans to be elected.

3.3 A Generic Task and Planning Ontology

We develop an ontology to conceptualise the tasks and the planning process for autonomous systems, as general as possible following an *domain-independent planning* approach. The idea is to find the commonalities of planning at an abstract level [11]. In the analysed planning ontologies, the general trend is to use the HTN approach. Tasks are decomposed in subtasks in a predefined manner. Trees tend to be searched according to a specific approach (depth first, breadth first, etc.). Our focus differs from this one in the sense that we would like the planning process to be domain and application independent, not constraining it to a particular planning paradigm. As an example, we would use a task representation such as a graph, task tree, but not only as a mechanism to decompose a task into its subtasks. Each node in the graph or tree would provide enough information on how to search in the graph, on possible actors to choose among capable of performing the task, on the type of action to be performed, etc. The idea is to separate between a task description, and its implementation and performance as an action by a particular robot. The task representation specifies what should be done, not how it would be done by a particular robot.

3.4 A Common Ontology with Different Views

As we would use the ontology for engineering the planning process in autonomous systems, we should address different stakeholders needs or concerns. *Viewpoints and views* are used to address stakeholders concerns to describe and engineer a system architecture [17]. A view addresses one or more concerns that system's stakeholders have. A viewpoint establishes the conventions for the construction, interpretation and use of views to frame specific system concerns. This is the same approach we envision as there will be different stakeholders in the planning process, and during the plan execution process. As a result, a common ontology would be use from different viewpoints to cater for the actual needs throughout the planning and decision-making processes.

4 The Ontology Contents

This section describes the initial ontological contents of an ontology for task and planning for autonomous system: *Plan*, *State*, *InitialState*, *DesiredGoal*, *GoalSpecification*, *Cost*, *CostFunction*, *Constraint*, *Task*, *TaskRepresentation*, *Action*, *Skill*, *Resource*, *Actor*, *Event* and *Object*. They have been initially defined using natural language and formalised as an UML class diagram (see Fig. 1). As naming convention, we use italics for the terms of the ontology, when they are defined in the text. Attributes appear also in italics, but they are not capitalised (unless a compound word where the second one is written with a capital letter).

On Plan, Initial State and Desired Goal. When planning, the purpose is to obtain a *Plan* as a finite sequence of actions that satisfy the constraints, goals, cost, etc. A key idea is how the environment or world encounters itself prior to the activity of the robot or human. It is a usual aspect considered in planning, as it allows to evaluate how the state of the environment changes after performing the planned actions. This concept is conceptualised in the ontology as *InitialState*, which is the state of the World at the beginning of the planning process. Likewise, we need to establish what the plan will achieve at the end. This concept becomes the *DesiredGoal*, i.e., what is to be accomplished in the process of solving the planning problem. In the case of the desired goal, a *GoalSpecification* can specify as a set of statements the desired goal to be achieved by the plan.

On Objects. The different objects in the environment, and as part of the autonomous system, need to be considered in the ontology. An *Object* is an entity in the system or the environment that will be used or affected during the planning process.

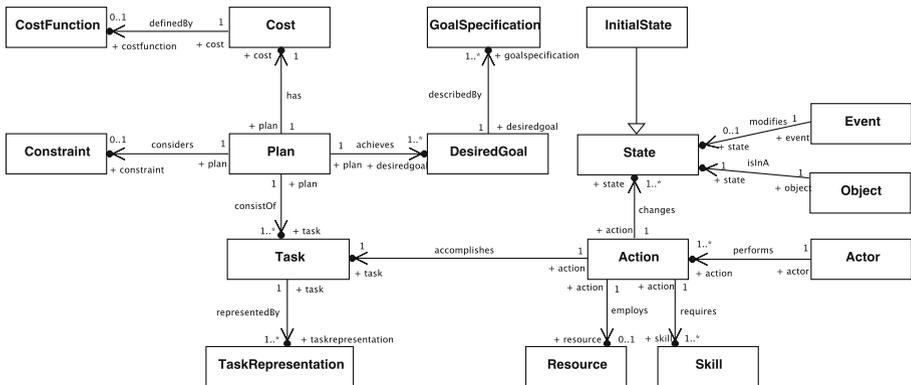


Fig. 1. An UML view of current ontology contents.

On Tasks. The concept of task is the cornerstone of the ontology. When we perform activities, we usually express them as a verb-like expression. We say something like “I need to take a plane tomorrow” or we command a robotic system “to go to some point”. We are just expressing what we would like to do, without paying attention to other factors. It is somehow an abstract viewpoint on what to do. Hence, a *Task* could be defined as an abstract expression of an activity. Nevertheless, if we go one step further, we could get into the detail on how to describe a task.

As humans, we know (most of the time) which activities involve a given one. For example, to take that plane tomorrow, we think that at least we will have to go to the airport, to do the check-in, to pass security, to arrive at the boarding gate, and finally to get into the plane. We have both described or named the activities we would have to do to get that plane, chopping them in chunks that are manageable. Hence, we have got a mental representation of what we want to do, although we do not usually represent it (except for agendas and smartphones calendars). In the case of a robotic system, we need to provide it with this knowledge.

The former analysis leads to different concepts in our ontology. Firstly, how we represent a particular Task. We have conceptualise this idea as a *TaskRepresentation* as a general term to conceptualise the knowledge to represent a Task. This is a traditional mechanism used in planning, but usually limited to a task tree. We would like to include other kind of task flow or diagrams, so a general term as representation seems a sensible idea.

Secondly, the fact that we decompose a task into other subtasks until we reach a point where we could actually perform them. Hence, the concept of *Subtask* is defined as a Task obtained as result of a task decomposition or analysis.

On Action and Event. The ontology includes the concepts of action and event, as the mechanisms that affect the state of the environment or the system itself. An *Action* is defined as a grounded task, in the sense that satisfies all the preconditions, constraints and alike with a valid effect. We want to reflect the viewpoint of an abstract task in the mind of the human or the system being actually implemented to obtain a result. We call this process grounding, as it would require to instantiate the general task description into a specific performing action in terms of related actors, skills and resources, whilst considering constraints and costs. Moreover, to define a particular action, we can add some attributes such as *actionName* to identify a particular Action, *precondition* to consider previous actions or elements necessary to perform it, additional information as *argument* that could have an *argumentName* and *argumentValue*. An *Event* can be defined as the occurrence of a change in the state (world or system state) because of external factors beyond the actor’s control. An Event is characterised by an *eventName*, a *condition* e.g. an obstacle is detected, a possible *action* to handle the event and a *terminationMode* to specify if the mission continues or needs to be aborted.

On Skill and Resource. To perform an action, we would need a set of elements to do so. These concepts have been included in the ontology at a initial stage as skill and resource. In general terms, a *Skill* is a particular ability of the system (either human or robotic). A *Resource* is an element used or consumed to perform a skill, such as time, memory space, or battery. Actions would require a particular skill when performing a task, and this skill would consume a particular resource. These can be modelled as relationships among the concepts.

On Actor. If we pay attention to who performs the action, we need to introduce the concept of actor. An *Actor* represents a person, group, an entity or system who is responsible for achieving different tasks by performing the corresponding actions. The latter can be conceptualised as a relationship between actors and actions. It remains to specialise the actors in the planning process in a possible taxonomy. It is worth mentioning that we have not used another common term, agent, for this concept as it conveys a software oriented approach. Actor is a rather general term that covers a wide range of possible entities participating in the process.

On Constraints. When planning, not all possible actions can be performed by the actors as there are some limitations in terms of timing, ordering, resources, required skills, available actors, etc. To represent this idea, the terms constraint and preference are included in the ontology. A *Constraint* can be defined as a property that must not be violated by a candidate or final plan. Constraints are usually classified as hard and soft ones.

On Cost. To choose among possible plans to solve the same problem, we need some kind of criterion to establish a ranking among plans. This idea could be conceptualised as the *Cost* associated to performing a plan. This cost can be evaluated or expressed as a *CostFunction*. A combination of constraints and cost, by means of proper axioms to be defined, will allow the planner to classify the possible plans to choose the most suitable one among them.

On State and Effect. Actions are performed with a goal in mind. They change the state of the environment, the objects in it, and the actors that perform the action. Hence, the concept of *State* is included in the ontology to define the (state) variables whose value changes over time or as a result of performing an action. For future planning needs, it will be necessary to detail between a predicted state (when the actor would reason about what might happen to decide the suitability of the action to be performed) as different from an observed state (when the actor would reason about what has happened to update its knowledge). However, this distinction is not fully formalised in the current ontology. As a result of performing an action, there will be an effect on the environment, the system itself or the actor.

5 A Case Study

5.1 The Emergency Scenario Description

This section analyses the use of the proposed set of concepts in an emergency scenario: due to a chemical explosion, there is risk for human operators to work in the damaged area. Hence a robotic system is sent to inspect and evaluate the risks after the explosion. The robotic system consists of a UGV/UAV combination. The UGV is equipped with a recharging platform, where the UAV can recharge its batteries. Both the UGV and the UAV have a camera (thermal and optical) as main sensorial inputs.

The mission is set to proceed as follows. The robotic system is sent to the emergency area from a base station. At this time, the UAV lies on the UGV with no activity. The UGV moves towards the emergency area from the base station. This could be either in an autonomous way or with help from an operator. When the UGV cannot continue its trajectory, the UAV takes off. The UAV will help as guidance for the path planning to be followed by the UGV. Once both the UGV and the UAV have reached the emergency area, they work collaboratively to inspect it, performing different tasks (taking pictures, making a map of the area, etc.). Once the mission is finished, the robotic duo returns to the base station.

As in any mission, unexpected events might happen. If there is a failure in the UGV that prevents it from moving, the action will be to try the start a finite number of times, and if not, to abort the mission. When the UGV encounters a blocked way to follow, the UGV stops, and the UAV takes off to guide the ground vehicle (providing points for its path planning). If the UAV cannot take off due to a failure, the UGV should not move, and the take off can be retried a number of times, and if not, the mission is aborted. When UAV battery level is low, the UAV has to land on the UGV while the UGV continues its movement. When the UGV is moving and its battery is low, the only possibility will be to abort or finish the mission, with the UAV landing on it to return to base station. Other event relates to finding obstacles, both by the UGV and the UAV, being necessary to change the path of the vehicle that encounters it.

5.2 Using the Ontology

To investigate the feasibility of the proposed concepts, we exemplify their instantiation for the emergency scenario. As a convention, we use *italics* for the ontology terms, *courier* for instances in the emergency scenario, () for detailing instance values, and = to provide attributes values.

Objects in the scenario are *zone* (base station, emergency area), *point*, *obstacle*, *robots* (multi-robot or UAV/UGV, UAV, UGV, robot part (camera, sensor), human operator).

The concept of *InitialState* can be used for different purposes. For example, to express that the UAV/UGV *location* at *zone* base station, with UAV *location* on UGV. The *DesiredGoal* will be for the multi-robot system, e.g.,

UAV/UGV to be at zone base station. During planning, it might be necessary to identify some kind of intermediate goals to specify the goals for each of the robots, e.g. UAV inspecting zone emergency area. When the mission is finished the goal will be for the multi-robot system to return to the zone base station.

Tasks can be defined at different level of detail. If we refer to the multi-robot system, tasks are leave base station, go to emergency area, explore emergency area, return to base station. UGV tasks can be leave zone (base station), go to zone (emergency area), explore zone (emergency area), take picture, return to zone (base station). UAV ones are take off, survey zone (emergency area), take pictures, make map, recharge, land on UGV. All are compound tasks to be detailed with *Sub-tasks* in a *TaskDescription* until reaching atomic tasks not possible to decompose (not shown here for sake of space).

An *Event* such as the failure of the UGV, i.e., UGV failure can be detailed with the following attributes values *eventName* = UGV start failure, *condition* = action failure, and *terminationMode* = repeat task. For *Event* such as UAV battery level is low, its attributes could be detailed as *eventName* = UAV Battery Low, *condition* = percentage battery charge less than a value, *action* = recharge, and *terminationMode* = next task. Other foreseen events can be defined in a similar way using the ontological contents.

According to the proposed ontology, actors perform actions. In the emergency scenario, *Actors* will be the multi-robot system or UAV/UGV, UAV, UGV, cameraUAV, sensorUAV, cameraUGV, sensorUGV, operator. These are the highest level actors. If needed, it would be necessary to detail specific type of cameras in each robot, and provided sensors.

Actions ground abstract tasks in terms of resources, and skills. An example of *Action* for the emergency scenario would be UAV scan zone (emergency area) as implementation of the defined survey zone emergency area task. For this particular action, it is necessary to provide further details in terms of other concepts proposed in the ontology. We can provide *actionName* = scan zone (emergency area) and specify the arguments to perform the action, such as the *actor* = UAV, where the action is performed *argumentName* = zone (emergency area), *argumentValue* = dimensions, the *precondition* as arguments such as speed and initial location, *argumentName* = speed, *argumentValue* = 5, *at init-Point* = (x0, y0, z0), time related aspects such as *startingTime* = t0, *finishingTime* = t1, where $t1 \geq t0$. Possible *Constraints* as ordering constraints of this action with respect to others e.g. this scan action can only take place if UAV has already taken off. *Skill* required for this action will be e.g. Avoid obstacle, and a *Resource* will be UAV camera, UAV battery. A possible evaluation of *Cost* for this action will be the battery consumption.

6 Concluding Remarks

This paper describes a proposal of an ontology for task description and planning in autonomous systems, such as robots, UAVs or UGVs. Our purpose is not to develop yet another ontology for emergencies and rescue where robots are involved. The emergency scenario described in Sect. 5 is just a mechanism to test the suitability of the approach. Our purpose is to provide a common terminology and relationships to define the planning problem in a high-level and abstract way, independently of the planning paradigm as presented in Sect. 3. To do so, a set of initial concepts have been defined as result of the analysis of existing ontologies for task and planning reviewed in Sect. 2. It remains to detail concepts, to specify attributes, and most importantly to detail relationships among concepts as well as to define axioms to ensure consistency during the planning process such as establishing which actors can actually perform a task, if resources will be available, constraint verification, etc. Our approach shares and integrates some contents of existing planning ontologies, however the novel approach is how the ontology will be used for autonomous systems planning.

Firstly, the development of *a methodology to select the required concepts according to a particular view* is a forthcoming step to use the ontology. The ontology represents the knowledge shared by different stakeholders about a domain, with a viewpoint as a subset of the complete representation that focuses on the relevant knowledge to a particular stakeholder.

We would also explore the *use of UML* [20] and *SysML* [19] to represent both the ontological concepts (domain knowledge) and the planning methods and algorithms (problem-solving knowledge) based on the common terminology provided by the ontology. Domain knowledge could be represented as class diagrams, object diagrams, SysML BDD or IBD. Problem-solving knowledge can be formalised as activity, sequence and state machine diagrams, allocation diagrams, etc. The underlying idea is to integrate metamodeling, modelling and ontologies for the planning domain: how we can actually obtain the possible and candidate plans from the ontology, how the actions are allocated, how the constraints are considered, etc.

An additional aspect to develop is the representation or description of tasks not only as hierarchical ones, but as more *flexible task representations*, where a tree leaf will include additional information on the task for planning purposes, e.g., behaviour trees. Nevertheless, further developments of the ontology will not only focus on a robot task representation ontology, but mainly on the planning aspects for autonomous robots. The emergency scenario described in this paper is just a particular application to test the suitability of the approach.

Related to it, it remains to use the ontological contents in the *deliberation and decision-making processes* that will provide the systems with autonomy. Planning is to choose and organise the actions to achieve a given objective or goal. In many cases, the actions are known beforehand. However, it is necessary some kind of decision-making, deliberation to decide on how to perform each action, considering the current context, changes in the environment, and effect.

Aspects of how humans intervene in the planning process from a mixed-initiative autonomy approach are also to be explored.

Acknowledgments. We acknowledge the support of the Spanish Government through grant RTC-2016-5191-8 Sistema Autónomo de Vigilancia y Seguridad basado en multirrotores (ADVISE), and grant RTC-2016-5059-8 Sistema Autónomo para La Intervención en Emergencias (SALINE).

References

1. Aghaeeyan, A., Abdollahi, F., Talebi, H.A.: UAV-UGVs cooperation: with a moving center based trajectory. In: *Robotics and Autonomous Systems* (2015)
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): *The Description Logic Handbook: Theory, Implementation and Applications*, 2nd edn. Cambridge University Press, New York (2010)
3. Balakirsky, S., Schlenoff, C., Fiorini, S., Redfield, S., Barreto, M., Nakawala, H., Carbonera, J., Soldatova, L., Bermejo-Alonso, J., Maikore, F., Goncalves, P., de Mori, E., Ragavan, S.V., Haidegger, T.: Towards a robots task ontology standard. In: *Proceedings of the ASME 2017 International Manufacturing Science and Engineering Conference*, Los Angeles, CA, USA (2017)
4. Behnke, G., Bercher, P., Biundo, S., Glimm, B., Ponomaryov, D., Schiller, M.: Integrating ontologies and planning for cognitive systems. In: *Proceedings of the 28th International Workshop on Description Logics (DL 2015)*. CEUR Workshop Proceedings (2015)
5. Bermejo-Alonso, J., Sanz, R., Rodríguez, M., Hernández, C.: An ontological framework for autonomous systems modelling. *Int. J. Adv. Intell. Syst.* **3**(3), 211–225 (2011)
6. Cantelli, L., Mangiameli, M., Melita, C.D., Muscato, G.: UAV/UGV cooperation for surveying operations in humanitarian demining. In: *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics*, pp. 1–6 (2013)
7. Chandra, R., Rocha, R.: Knowledge-based framework for human-robots collaborative context awareness in USAR missions. In: *16th IEEE International Conference on Autonomous Systems and Competitions* (2016)
8. Chandrasekaran, B., Josephson, J., Benjamins, V.R.: Ontology of tasks and methods. In: *11th Workshop on Knowledge Acquisition, Modeling and Management (KAW98)*. Banff, Alberta, Canada (1998)
9. Freitas, A., Schmidt, D., Panisson, A., Meneguzzi, F., Vieira, R., Bordini, R.H.: Semantic representations of agent plans and planning problem domains. In: Dalpiaz, F., Diz, J., van Riemsdijk, M. (eds.) *Engineering Multi-Agent Systems: Second International Workshop, EMAS 2014, Paris, France, 5–6 May 2014, Revised Selected Papers*. Lecture Notes in Artificial Intelligence (LNAI), vol. 8758, pp. 351–366. Springer International Publishing (2014)
10. Gangemi, A., Borgo, S., Catenacci, C., Lehmann, J.: Task taxonomies for knowledge content. Technical report D07, Laboratory for Applied Ontology (LOA) (2005)
11. Ghallab, M., Nau, D., Traverso, P.: *Automated Planning and Acting*. Cambridge University Press, Cambridge (2016)
12. Gil, Y.: Description logics and planning. *AI Mag.* **26**(2), 73–84 (2005)
13. Gil, Y., Blythe, J.: PLANET: a shareable and reusable ontology for representing plans. In: *Proceedings of the AAAI Workshop on Representational Issues for Real-world Planning Systems*, pp. 28–33 (2000)

14. Gruber, T.: Ontology. In: Liu, L., Ozsu, M.T. (eds.) *Encyclopedia of Database Systems*, pp. 1963–1965. Springer, New York (2009)
15. Guarino, N., Oberle, D., Staab, S.: What is an ontology? In: Staab, S., Studer, R. (eds.) *Handbook on Ontologies*, pp. 1–17. Springer, Heidelberg (2009)
16. Habib, M., Baudoin, Y., Nagata, F.: Robotics for rescue and risky intervention. In: *37th Annual Conference on IEEE Industrial Electronics Society (IECON)*, pp. 3305–3310 (2011)
17. ISO/IEC/IEEE: 42010:2011: Systems and software engineering - architecture description. Technical report, ISO/IEC/IEEE (2011)
18. Karma, S., Zorba, E., Pallis, G., Statheropoulos, G., Balta, I., Mikedi, K., Vamvakari, J., Pappa, A., Chalaris, M., Xanthopoulos, G., Statheropoulos, M.: Use of unmanned vehicles in search and rescue operations in forest fires: advantages and limitations observed in a field trial. *Int. J. Disaster Risk Reduct.* **13**, 307–312 (2015)
19. Object Management Group (OMG): *OMG Systems Modeling Language (SysML) Version 1.4* (2015)
20. Object Management Group (OMG): *Unified Modeling Language (UML) Version 2.5* (2015)
21. Rajhpathak, D., Motta, E.: An ontological formalization of the planning task. In: *International Conference on Formal Ontology in Information Systems (FOIS)*, Torino, Italy (2004)
22. Bouillet, E., Feblowitz, M., Liu, Z., Ranganathan, A., Riabov, A.: A knowledge engineering and planning framework based on OWL ontologies. In: *Proceedings of the Second International Competition on Knowledge Engineering* (2007)
23. Sánchez-Ruiz, A., González-Castro, P.A., Díaz-Aguado, B.: Planning with description logics and syntactic updates. In: *Planning, Scheduling and Constraint Satisfaction CAEPIA 2007 Workshop*, pp. 140–150 (2007)
24. Schlenoff, C., Messina, E.: A robot ontology for urban search and rescue. In: *2005 ACM Workshop on Research in Knowledge Representation for Autonomous Systems* (2005)
25. Sirin, E., Parsia, B., Wu, D., Hendler, J., Nau, D.: HTN planning for web service composition using SHOP2. *J. Web Seman.* **1**(4), 377–396 (2004)
26. Tanner, H.G.: Switched UAV–UGV cooperation scheme for target detection. In: *IEEE International Conference on Robotics and Automation*, pp. 3457–3462 (2007)
27. Tate, A.: Towards a plan ontology. *AI*IA Notizie (J. Ital. Assoc. AI)* **9**(1), 19–26 (1996)
28. Uschold, M., King, M.: Towards a methodology for building ontologies. In: *Workshop on Basic Ontological Issues in Knowledge Sharing (IJCAI95)* (1995)
29. Valente, A.: Knowledge-level analysis of planning system. *SIGART Bull.* **6**(1), 33–41 (1995)
30. W3C: *OWL 2 Web Ontology Language, Second edn.* (2012)