Reviewing Task and Planning Ontologies: An Ontology Engineering Process

Julita Bermejo-Alonso
UPM Autonomous Systems Laboratory (ASLab) c/José Gutiérrez Abascal, 2, 28006 Madrid, Spain

Keywords: Task Ontologies, Planning Ontologies, Ontological Engineering, Autonomous Systems.

Abstract: Bermejo-Alonso and colleagues (Bermejo-Alonso et al., 2018) define an ontology for tasks and planning in the autonomous system domain. It focuses on an emergency scenario for Unmanned Ground Vehicles (UGVs) or Unmanned Aerial Vehicles (UAVs). In this context, it is necessary to define how the autonomous system will act, detailing how the actions should be done to achieve system goals. The planning process starts with detailing the *planning domain knowledge*: the initial state, the goals, the actors, the resources, etc. This domain knowledge is then fed into a planner that, if a solution exists, will produce a plan or a set of plans to be used by the robotic system. Ontologies are a useful way to provide this domain knowledge and can be used to characterise the planning domain knowledge. However, there is a number of available ontologies for planning, being unclear which one is best for autonomous systems. This paper presents a *review of existing task and planning* vocabularies, taxonomies and ontologies, as a necessary first step in an *ontology engineering process* that addressed the autonomous system planning needs. This paper describes the analised ontologies, their main features, and how the process to integrate them was carried out.

1 INTRODUCTION

To solve a problem, we would need two kinds of knowledge: the domain knowledge and the problem–solving knowledge (Chandrasekaran et al., 1998). If this problem relates to planning, the *knowledge about the domain* represents the objects, states, causes of change states, tasks, effects, etc. The *problem–solving knowledge* would represent the goal, the domain data, the problem–solving state, the methods and algorithms, etc. The challenge is how to integrate the planning algorithms and methods (i.e. problem–solving knowledge) with a rich representation of planning and task knowledge in terms of e.g. plans, tasks, actions and events (i.e. domain–specific knowledge) (Gil, 2005). The idea would be to provide the planner a consistent representation of the complete domain to work with for any individual planning problem (Hartanto and Hertzberg, 2009).

Defining the different elements involved in a planning problem tends to be a time–consuming task, since it requires to detail all the elements involved in the planning process, each time the planning problem is approached. It would be a key element not only that the knowledge about processes and activities could be shared among the stakeholders, but also that this knowledge could be reused from planning to planning problem.

An ontology that provides the concepts and relationships of the domain can ease the planning process. Firstly, ontologies can be used to characterise and share the planning domain knowledge, as some kind of knowledge base. They represent an approach to enable richer plan representations, as well as knowledge reuse across planning applications (Gil and Blythe, 2000). Secondly, ontologies are the core based on which the planning reasoning mechanisms do produce the plan (Moreno et al., 2000).

However, there is not a unique, agreed view on a planning ontology. This is the problem we faced when attempting to find an existing ontology as backbone for our particular planning problem for an autonomous system in an emergency scenario. This paper describes the review process and not the resulting ontology that has been described elsewhere. The review process analysed and compared existing vocabularies and ontologies, to evaluate their possible re-use or their integrability into a single ontology. The resulting ontology would become a reusable component integrated in the Ontology for Autonomous Systems (OASys) (Bermejo-Alonso et al., 2011), (Bermejo-Alonso et al., 2013).

Bermejo-Alonso, J.
Reviewing Task and Planning Ontologies: An Ontology Engineering Process.
DOI: 10.5220/0006922401830190
In *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2018) - Volume 2: KEOD*, pages 183-190
ISBN: 978-989-758-330-8

The paper is organised as follows: Section 2 reviews the different ontologies found in the literature. Section 3 summarises the performed ontological engineering process. Finally, Section 4 provides some concluding remarks, as well as future lines of research.

2 A SUMMARY OF PLANNNING ONTOLOGIES

This sections summarises the main features of the reviewed ontologies. They are classified based upon the formalisation mechanism or language used in their definition.

2.1 Natural Language Task Ontologies, Vocabularies and Taxonomies

Natural language was the first approach to develop vocabularies, taxonomies and ontologies for the planning domain. Sometimes, the conceptualisation focuses on tasks only. Other times, the ontologies also provide additional concepts to characterise the goals, the actions, and plan features.

An initial attempt to provide a planning ontology in natural language is proposed in (Valente, 1995) with a set of concepts or vocabulary at knowledge-level to compare different planning approaches existing at that time. They define the kind of knowledge involved in the planning task, specifying the dynamic and static roles involved. The dynamic role refers to the plan as the goal of the planning task. It will be equivalent to the problem–solving knowledge. The static roles relate to the plan model consisting of the world description and the plan description. Concepts are defined in natural language, with a medium level of formalisation. The same approach is further detailed and extended in (Benjamins et al., 1996) but with an even lower level of formalisation.

To standardise process and plan interchanges, other natural language ontology is proposed in (Tate, 1996), (Tate, 1998). The ontology is structured as different elements: meta-ontology, top level ontology, library of shared ontological elements and detailed ontology section. The concept definitions includes terms such as plan, agent, constraint , environment, and issue. Strictly speaking, it consist more of a dictionary or taxonomy of plan related concepts than a proper ontology. Some relations are verbally expressed between the concepts Agent and Constraint, but no other elements required in an ontology, such as axioms, are provided.

An ontology for a problem solver is described in (Chandrasekaran et al., 1998), considering two different levels of ontologies. The first one relates to the domain knowledge. The second–level one to the methods and algorithms to be used in the planning process. As part of the first–level one, the Domain Factual Knowledge (objects, properties, relations, classes and subclasses, states, processes, events, and parts), the Problem Solving Goal (world description), or the Problem State components (goals, subgoals, requirements as preconditions, etc). Within the second–level ontology, a Control Ontology where tasks are characterised as sequential, conditional, iteration, and recursion. Despite of being called ontologies, the actual descriptions do not sum up. A series of concepts are considered and merely defined.

A comprehensive effort to conceptualise the planning process is the PLANET ontology (Gil and Blythe, 2000), where different planning related concepts and relationships are provided in natural language. The notion of a plan is represented using a series of concepts: initial-planning-context, goals, actions, tasks, and choice-points. The initial-planning-context consists of an initial state, goal description, and external constraints and a resulting plan is represented as a set of commitments to actions taken by an agent to achieve the specified goals. The goals describe what needs to be achieved when developing a plan. A solution plan is validated against completion, consistency, feasibility, and justified criteria. The ontology does not include traditional planning elements related to actors or agents, locations, time and resources. Neither the evaluation of plans according to a criterion.

The viewpoint of planning consisting of dynamic and static roles defined in (Valente, 1995) is reused and extended in (Rajhpathak and Motta, 2004). In this case, the authors provide a task ontology in terms of the initial world state, the goal to be achieved with the planning process, the plan tasks, the actions, the agents, the constraints, the preferences as soft constraints, and the evaluation through a cost function. The ontology is aimed at representing a generic planning task ontology, not related to a particular planning paradigm, specific domain or application. It is formalised at a high level, with a precise and comprehensive description of planning related terms for the planning problem. However, it uses the Operational Conceptual Modelling Language (OCML) (Motta, 1997) which was used during some years but lacks an updated version.

A recent effort to characterise a Task Ontology within the robotics domain is provided by (Balakirsky et al., 2017). The authors propose doing task characterisation in terms of different task dimensions (description, property, implementation, and context) to suitably organise the ontological elements. These definitions are an initial step towards the ontological engineering of a Robot Task Representation ontology to be standardised by IEEE. The dimensions approach is a good mechanisms to approach the different ontology perspectives.

2.2 Description Logics based Ontologies

Other ontologies have been formalised using Description Logics (DL) (Baader et al., 2010), a family of knowledge representation languages used to formalise the knowledge of an application domain in a structured manner. There are several efforts to characterise or analyse the planning knowledge from a DL perspective, as reviewed in (Gil, 2005). Generally speaking, these ontologies focused on the domain representation only, as taxonomies of objects in the domain, action, plan, and goals. Procedural or problem–solving knowledge is rarely considered in these ontologies. Using DL allows to organise the class description in a taxonomic hierarchy, which is useful when a Hierarchical Task Network approach is used to solve the planning problem. Moreover, DL reasoners provide two interesting capabilities: class subsumption and instance recognition. Additional DL benefits include inference, modularity and tolerance for inconsistent descriptions. This paper also lists use of these taxonomies by different planning algorithms and reasoners.

The DOLCE+DnS Plan Ontology (DPPO) (Gangemi et al., 2005), where DnS stands for the Ontology of Descriptions and Situations, is an ontology for planning based in DOLCE (Borgo and Masolo, 2010). The intended use of DPPO is to specify plans at an abstract level, without considering existing resources. The underlying idea is to provide an ontology to specify social or cognitive plans, not so much computationally executable plans. This example focuses more on conceptualising the domain object representation, not so much on procedural knowledge about how to tackle the planning process. It is worth mentioning that the different kinds of tasks initially mentioned in (Chandrasekaran et al., 1998) are detailed and extended in this ontology. This way a task is not only defined as an activity to perform, but also on how it should be performed as part of a plan.

Additional research attempted to integrate the domain knowledge with the planning process. A DL ontology for the Hierarchical Task Network (HTN) planning paradigm is described in (Sánchez-Ruiz et al., 2007). The authors consider that to resolve a planning problem, two different kinds of knowledge need to be formalised. On the one hand, the *domain description* as the rules of the world where the planner will act in terms of domain constraints and operators (this term being understood from a HTN perspective, not a human-being within the planning problem). On the other hand, the *description of the problem* in terms of the initial state and the goal, where a plan becomes a sequence of grounded operators to evolve the world from this initial state to a final one that fulfils or satisfies the expressed goal.

A combination of OWL-DL and HTN planning is proposed in (Hartanto and Hertzberg, 2009), (Hartanto, 2009), where DL is used to express the planning domain by an ontology. The HTN Planning Ontology includes a DL formalisation of concepts such as planning domain, planning problem, HTN method, operator, and task network. Once the ontology is defined, it is used by a DL reasoner, to obtain a concrete or filtered DL model. The overall process was applied to a mobile robotic navigation domain. However, the research focuses more on the reasoning process phases as a combination of DL reasoning and a HTN planner, where the problem–solving knowledge is formalised as an ontology. No mention is made to the domain knowledge.

Behnke et al. (Behnke et al., 2015a), (Behnke et al., 2015b) provide an approach to integrate ontologies within the planning problem. Their objective is twofold. Firstly, to define an ontology as the central element of domain knowledge. This ontology consists of two ontologies, one for the planning domain concepts (O1) and a second one for the domain knowledge (O2). Within O1, different planning-related concepts in terms of abstract tasks (compound tasks), subtasks, task hierarchy, refinement or decomposition method, relations and axioms are defined following the HTN planning formalism. O2 contains the knowledge about the objects and elements in the domain. The overall ontology is used as the knowledge element. The level of formalisation is not high, but too oriented to the HTN planning paradigm.

2.3 OWL Related Ontologies

Another set of ontologies for task and planning have made use of OWL (W3C, 2012). In (Sirin et al., 2004), an OWL reasoner was integrated with an artificial intelligence planner. The reasoner was used to store the world state, answer the planner's queries about the evaluation of preconditions, and update the state when the planner simulated the effects of operators. An HTN planning system was integrated with an OWL DL reasoner to explore the use of semantic reasoning over the ontology. Some general notions on concepts supporting the planning process, such as action, effects, or states are given. No detailed description of the ontology is provided, with a focus on the planning mechanisms from a HTN perspective.

Bouillet et al. (Bouillet et al., 2007) propose a domain-independent, general purpose knowledge engineering and planning framework to construct planning domains and problems based on OWL ontologies. The planning model is described as a set of OWL facts, represented as RDF graphs. Actions are described as RDF graph transitions. In turn, preconditions, effects as well as planning goals are also specified as RDF graph pattern. The framework allows for the different stakeholders to participate as domain experts, action specifiers or end-users to specify the ontologies, actions and planning goals respectively. The justification to use OWL ontologies is to be able to reuse available OWL ontologies with common and general concepts for planning, such as time and so.

An OWL ontology is developed for planning domains using Protégé, based on the HTN (Hierarchical Task Network) paradigm in (Freitas et al., 2014). The ontology provides a set of concepts such as: DomainDefinition as the definition of the planning domain, Operator to represent the primitive tasks, Method to decompose compound tasks, Axioms to infer preconditions for methods and operators, Predicates to represent the preconditions and postconditions of actions and world state, Goal as method invocations, etc. To exemplify its use, they instantiated the ontology for a multi–agent scenario known as gold miners, with one instance for the operator and the methods. They also proposed algorithms to convert the OWL planning ontology to other formalisms such AgentSpeak language or to SHOP (an HTN planning system). This way, they solved both the knowledge representation (as an ontology) with the problem solving (automated planning).

Balakirsky (Balakirsky, 2015) proposes an OWL ontology within agile manufacturing. The ontology is divided in three parts. The first part contains knowledge on elements involved in the application: basic elements such as point to conceptualise 3-D positions or more complex elements such as parts to define how a part is composed of, where is stored, and its name. The second part contains all the action-related concepts, with a focus on robotic and vision aspects. The third part of the ontology contains the specific instances for this particular domain. The actual contents of the three ontologies are provided in a graphical way, with not much level of detail on actual definitions, properties or axioms.

3 ONTOLOGICAL ENGINEERING PROCESS

This section describes the different stages followed in the ontological engineering of a planning ontology for our robotic application, by analysing in detail the reviewed ontologies described in Section 2.

3.1 Graphical Representations

As described in Section 2, different formalisms have been used in planning ontologies. To establish a comparison of them, the first step was to represent them on a common ground. To prevent committing to a particular implementation language (natural, DL, OWL, etc) or planning mechanism (HTN, etc), the representation of the ontologies was made as a diagram where most important concepts and relationships were detailed. This process was carried out for all the reviewed ontologies. For the sake of space in this paper, partial examples are provided in Fig. 1, Fig. 2, and Fig. 3.

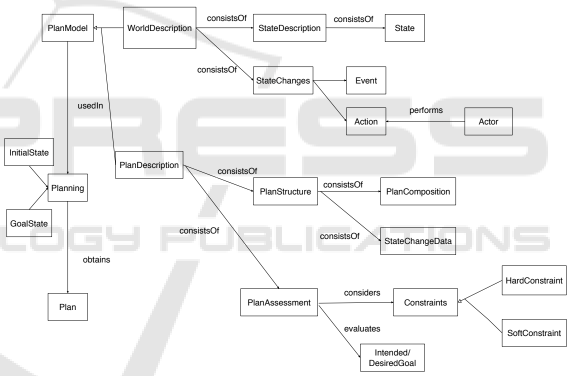

Figure 1: Graphical representation of (Valente, 1995).

The goal was to represent the different planning vocabularies, taxonomies and ontologies in a common and straightforward representation. This way, we could analyse the main concepts and relationships, to determine if any of them was suitable to be applied to our particular robotic system application. The analysis of the graphical representations provided different outcomes.

Firstly, the different vocabularies and ontologies focused on the main concepts for the planning domain knowledge, such as plan, task, action, actor, constraint, etc, however with different level of detail. The two main problems were that i) the same term was used with different purposes and ii) different terms were used to represent the same concept. Aproaching the analysis with a graphical representation helped providing some insights on the main concepts to be considered as the core of the integrated ontology.

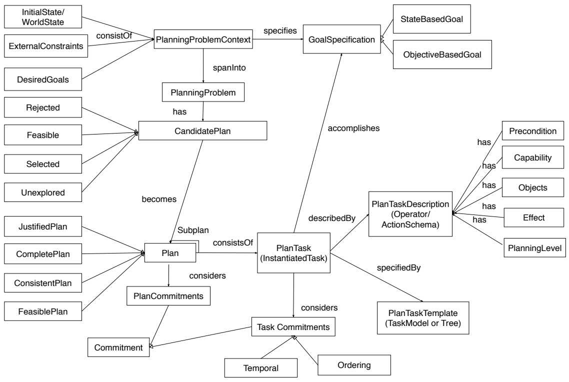

Figure 2: Graphical representation of (Gil and Blythe, 2000).

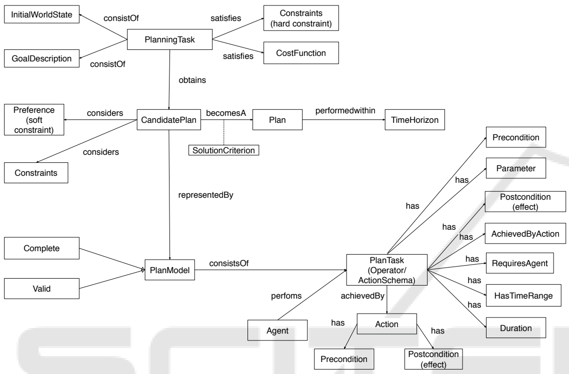

Figure 3: Graphical representation of (Rajhpathak and Motta, 2004).

However, it was clear that further work was needed to establish commonalities among the different ontologies. Not so much on the terms themselves, but the underlying ontological conceptualisations.

3.2 Tabular Representation

As a first step to reuse and integrate the ontological contents, the different terms from the vocabularies, taxonomies and ontologies were represented in an intermediate tabular form. Some partial views of the tabular representation for the same graphical representation in Section 3.1 are shown in Fig. 5, Fig. 6, and Fig. 7.

These tables collected the definitions of the different terms, either explicitly provided in the original paper or extracted from its text. The terms were further classified as concepts (C), attributes (A), or relationships (R).

The ontological contents of these tables were later divided and classified according to different dimensions as described in Section 3.3.

TERM	SYNONYMS	DEFINITION	TYPE
PlanModel		What a plan is and what is made of. It consists of the WorldDescription and the PlanDescription	C
WorldDescription		Knowledge about the World in which the Planning will take place. It comprises StateDescription and StateChange	C
WorldState		The State of the World	C
StateDescription		Part of WorldDescription. Knowledge to represent the WorldState	C
StateChanges		Part of WorldDescription. Knowledge to represent the changes in the WorldState. Specification of the elements a Plan is composed of.	C
Plan			C
Subplan		Routines that are composed of commands which are either Actions or other Subplans.	C
PlanDescription		Knowledge about the Plan to be generated. It consists of PlanStructure and PlanAssessmentKnowledge.	C
PlanStructure		It specifies how the parts of a Plan such as Actions or Subplans are assembled together. It consists of PlanComposition and StateChangeData	C
PlanComposition		Description of the plan with respect to how the StateChanges are arranged to make up a plan.	C
StateChangeData		Knowledge about the Resources (Agent, time, etc) assigned to StateChanges.	C
PlanAssessmentKnowledge		Factors that make a certain Plan (or Subplan) better that another	C

Figure 4: Tabular representation of (Valente, 1995).

TERM	SYNONYMS	DEFINITION	TYPE
PlanModel		What a plan is and what is made of. It consists of the WorldDescription and the PlanDescription	C
WorldDescription		Knowledge about the World in which the Planning will take place. It comprises StateDescription and StateChange	C
WorldState		The State of the World	C
StateDescription		Part of WorldDescription. Knowledge to represent the WorldState	C
StateChanges		Part of WorldDescription. Knowledge to represent the changes in the WorldState. Specification of the elements a Plan is composed of.	C
Plan			C
Subplan		Routines that are composed of commands which are either Actions or other Subplans.	C
PlanDescription		Knowledge about the Plan to be generated. It consists of PlanStructure and PlanAssessmentKnowledge.	C
PlanStructure		It specifies how the parts of a Plan such as Actions or Subplans are assembled together. It consists of PlanComposition and StateChangeData	C
PlanComposition		Description of the plan with respect to how the StateChanges are arranged to make up a plan.	C
StateChangeData		Knowledge about the Resources (Agent, time, etc) assigned to StateChanges.	C
PlanAssessmentKnowledge		Factors that make a certain Plan (or Subplan) better that another	C

Figure 5: Tabular representation of (Valente, 1995).

TERM	SYNONYMS	DEFINITION	TYPE
PlanninProblemContext		Initial given assumptions about the PlanningProblem	C
PlanningProblem		Specific Goals, Constraints and assumptions about the (World?)InitialState	C
WorldState		Model of the Environment for which the Plan is intended	C
World	Environment		
WorldInitialState	InitialWorldState	A certain WorldState description	C
DesiredGoal		What is to be accomplished in the process of solving the PlanningProblem	C
ExternalConstraint		Desirable or undesirable properties or Effects of Solutions to the PlanningProblem, such as user advice and preferences	C
Plan		A set of Commitment of Actions taken by an Agent to achieve some specified Goals.	C
CandidatePlan		Potential solution to a PlanningProblem	C
Untried		A CandidatePlan that is yet to be explored or tested	C
Rejected		For some reason, CandidatePlan that has been rejected as the preferred one	C
Feasible		A CandidatePlan that has been tried and not rejected	C
Selected		A CandidatePlan that has been	C
GoalSpecification		Anything that gets accomplished by a Plan, Subplan or Task	C
StateBasedGoalSpecification		GoalSpecification to represent Goals that refer to some predicate used to describe the WorldState	C
ObjectiveBasedGoalSpecificaton		GoalSpecification to represent verb- or action-based expressions	C
HumanReadableDescription		GoalSpecification to provide a description of a Goal to an EndUser	C
PlanTaskDescription	TaskDecomposition TaskTree	Actions or operations that can be taken in the WorldState	C
PlanTask	Task	An instantiation of a Task as it appears in a Plan	C
PlanTaskTemplate	TaskTemplate	An Action that can be performed in the WorldState	C
Subtask		A TaskDescription (aka Task)	C
PrimitivePlanTaskDescription		A primitive TaskTemplate	C
Capability		A Goal for which the Task can be used	C

Figure 6: Tabular representation of (Gil and Blythe, 2000).

3.3 Concept Integration

To establish a criterion on how to organise the ontological conceptualisations, we considered that the main component in any planning process is the task. As such, a task theory enables addressing tasks at class level through the definition of tasks, environment, and their characteristics from a formal viewpoint (Thorisson et al., 2016). Being impossible to represent or characterise all possible tasks, such a theory should focus on characterising the main features of different

TERM	SYNONYMS	DEFINITION	TYPE
InitialWorldState		The WorldState at the beginning of the planning process	C
Goal		Desired WorldState to achieve through the planning process.	C
PlanTask		A set of plantask which specify intermediate Goals which need to be accomplished to achieve the overall Goal of the Planning Task	C
Action		A finite set of Actions to be executed to accomplish a PlanTask	C
Agent		A set of Agents who are responsible for achieving PlanTasks through the execution of Actions. A person, group or entity which holds a purpose and is responsible for achieving different PlanTasks through the execution of Actions.	C
Parameter		Pointer to domain entities relevant to the Planning Process	C
TimeHorizon		A time window within which the Plan is required to take place	C
Constraint	Hard Constraint	A set of constraints not to be violated by the Plan. A Property that must not be violated by a valid Solution.	C
Preferences	Soft constraint	A set of criteria for partially ranking competing plans	C
CostFunction		A function which provides a global mechanism for comparing the Cost of AlternativePlan	C
SolutionCriterion		A mapping from a Plan to (true, false) which determines whether a CandidatePlan is a Solution	C
PlanOptimization Criterion		A Criterion to choose an OptimalPlan with the least Cost	C
OptimalPlan		A Plan with the least Cost according to some evaluation function	C
PlanModel	Candidate Plan	A sequence of (PlanTask,Agent) pairs where the Agent is able to execute the Actions associated with the PlanTask	C
hasParameter		Parameter associated with a PlanTask	A
hasPrecondition		Condition to be true before a PlanTask is executed	A
hasPostcondition		Condition to be true after a PlanTask is executed	A
achievedByAction		Action to be executed to achieve the PlanTask	A
requiresAgent		A PlanTask is associated with one or more Agents to achieve it	A
hasTimeRange		The TimeWindow within which a PlanTask must take place.	A

Figure 7: Tabular representation of (Rajhpathak and Motta, 2004).

task-related concepts: the task itself, the environment where it takes place, the goals, the effects, and the agents that perform the task. With this approach, the tabular representations made for each one of the vocabularies, taxonomies and ontologies considered in Section 2 were further classified to characterise different dimensions of the concept of task:

- Task Context: this set of concepts were focused on the world (aka the environment) and its state. Considered concepts such *WorldInitialState*, *WorldDescription*, etc.
- Task Description: the concept of task is the cornerstone of the ontology. How the concept of task has been named and conceptualised was considered within this dimension. A partial view of the result is shown in Table 1.

Table 1: Some concepts for the Task Description dimension.

NAME	SYNONYM	DEFINITION	TYPE	NOTES	SOURCE
Task		An Action with proper input and output information that moves the application from one State to another.	Concept		Balakirsky2017
Task	Activity	An activity as part of a Plan	Concept		Behnke2015b
PlanTask	Task	An instantiation of a Task as it appears in a Plan	Concept	Plan just as a prefix to imply it is a Task part of a Plan, which leads to ordering or temporal aspects.	Gil2000
PlanTask		A set of Plan Tasks which specify intermediate Goals which need to be accomplished to achieve the overall Goal of the Plan Task	Concept		Rajtaphak2004

- Task Decomposition: different types of tasks were included, such as *CompoundTask* to characterise tasks to be further decomposed, as well as *PrimitiveTask* to be used for single tasks. Synonyms were established from all the analysed ontologies. A partial view on different task decomposition concepts are shown in Table 2.

Table 2: Some concepts for the Task Decomposition dimension.

NAME	SYNONYM	DEFINITION	TYPE	NOTES	SOURCE
PrimitiveTask	AtomicTask SimpleTask ElementalTask EndingTask TerminalTask	A single Task that can be performed by a single Agent (Actor). A Task that cannot be decomposable. A Task that can be directly executed by the User (Agent/Actor?).	Concept Subclass of Task		Various authors Various authors Behnke2015b
CompoundTask	Complex Task	A Task that can be decomposed in other SubTasks Abstract Task or high-level Task that has to be refined into more tangible Tasks, being PrimitiveTask or other CompoundTask.	Concept Subclass of Task		Various authors Behnke2015b
Subtask		A Task one is decomposed in.	Concept		Balakirsky2017

- Task Implementation: from our viewpoint, there is a subtle difference between the concept of *Task* and *Action*. Task is regarded from a conceptual viewpoint, whereas action will be used in the actual planning process to assign actors and resources. In this sense, the implementation concepts included *Precondition* or *Effect* to represent the conditions to and the result of performing a particular action. Among this set of concepts, it is also needed to characterise the *Actor* that performs the action, to make possible to further distinguish among the human or the artificial actors in the system. Also, preconditions and constraints are part of this dimension, as they will be used to determine whether an action can or cannot be performed to fulfil a required task. Some concepts are shown in Table 3, and Table 4.

Table 3: Precondition concepts for the Task Implementation dimension.

NAME	SYNONYM	DEFINITION	TYPE	NOTES	SOURCE
Precondition		A necessary condition for a Task	C	As a class or as attribute of Task?	Gil2000
Precondition		Condition under which a PrimitiveTask can be applicable in a State		Part of PrimitiveTask. Defined as literal	Behnke2015b
Axiom		Preconditions not explicitly asserted in the current State	C	DISCARDED: axiom is a reserved word	Freitas2014, 2014b
Predicate		Expression used to represent Precondition, Postcontion	C		Freitas2014, 2014b
Parameter		Pointer to domain entities relevant to the planning process	C		Rajtaphak2004

- Task Evaluation: an important part of the planning process is to define the goal and some evaluation mechanism to choose among feasible plans. The latter is usually made using some kind of cost evaluation in terms of resources or time. With this purpose, concepts such as *Goal* or *CostFunction* were included in the integrated ontology.

Table 4: Constraint concepts for the Task Implementation dimension.

NAME	SYNONYM	DEFINITION	TYPE	NOTES	SOURCE
TaskConstraint	Constraint	A limitation defining the desired properties of the Task.	Concept		Balakirsky2017
Constraint		An assertion that can be evaluated with respect to a given Plan			Tate1996
Constraint	HardConstraint	A set of constraints not to be violated by the Plan. A Property that must not be violated by a valid Solution.	Concept	A constraint can be theoretical, technical, engineering, social or cognitive control actions	Rajtaphak2004
ExternalConstraint		Desirable or undesirable properties or Effects of Solutions to the PlanningProblem, such as user advice and preferences	Concept		Gil2000
Commitment		Some kind of choice to be taken into account; A Constraint the Agent elects to add as a partial specification of a Plan	Concept	RELATIONS: An Agent may Intend, Desire, Enforce or Synthesise a Commitment	Gil2000
Preference	Soft constraint	A set of criteria for partially ranking competing plans	Concept		Rajtaphak2004

For each aspect, related concepts from the analysed vocabularies and ontologies were grouped, to identify similarities, becoming synonyms, e.g. simple tasks are named *PrimitiveTask*, *EndingTask*, *SingleTask*, *TerminalTask* are used. The underlying conceptualisation was the same, so a single concept *SimpleTask* was chosen. Names were analysed to use a meaningful one, e.g. between *Agent* and *Actor*, the second one was chosen to remove the software related meaning of agent. Around 175 concepts regarding task and planning were identified, defined, grouped, and analysed for the final ontology.

4 CONCLUSIONS

As a result of the review and the ontological engineering process described in former sections, an integrated ontology to provide a common terminology and relationships to define the planning problem in a high–level and abstract way was obtained. The ontology rationale, the initial set of concepts included, and its application in an emergency scenario for a combined UAV/UGV robotic system is detailed in (Bermejo-Alonso et al., 2018). The ontology is under extension, as additional information needs to be added.

To provide the flexibility and the autonomy required in a robotic unmanned system, it will be important to detail additional aspects for tasks. For example, concurrency among tasks being performed, or deliberation tasks for deliberation states in the autonomous system. It is also important to extend the HTN approach of compound tasks to be decomposed. With this purpose, a general concept *TaskRepresentation* was chosen to expando to other representations, such as behaviour trees (Colledanchise and Ogren, 2018) that address reactive behaviour depending on the set of action decided by the player in a game. Another example would be the need for deliberation tasks as a task to represent deliberation states in the autonomous system.

The ontology obtained as result of the ontological engineering process described in this paper focused mainly in the domain knowledge, that is, the concepts needed to conceptualise a task theory from different viewpoints (context, implementation, evaluation, etc) as described in Section 3.3. Nevertheless, as stated in Section 1, for the planning process is necessary to consider both this domain knowledge, but also the problem–solving knowledge. The set of concepts related to the latter are not yet conceptualised in the current ontology. Among other elements, future development should pay attention to the decomposition methods for task trees, to behaviour trees handling, or to other algorithms to obtain the final plan according to the selected cost or evaluation, where actors and resources are clearly allocated to specific actions.

We would also explore using UML (OMG, 2015b) and SysML (OMG, 2015a), to represent both domain knowledge and the problem-solving knowledge. 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 purpose would be to integrate metamodelling, modelling and ontologies for the planning domain from a systems engineering perspective: to define the plan requirements based on the domain knowledge ontology, how preconditions and constraints are considered, how to allocate the actions to specific actors, how the reasoning and deliberation process takes place to obtain the possible and candidate plans from the ontology, and how the evaluation process can be implemented to select among feasible plans.

ACKNOWLEDGEMENTS

We acknowledge the support of the Spanish Government through grants RTC-2016-5191-8 Sistema Autónomo de VIgilancia y SEguridad basado en multirrotores (ADVISE), and RTC-2016-5059-8 Sistema Autónomo para La INtervención en Emergencias (SALINE).

REFERENCES

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. F., editors (2010). *The Description*

Logic Handbook: Theory, Implementation and Applications. Cambridge, 2nd edition edition.

Balakirsky, S. (2015). Ontology based action planning and verification for agile manufacturing. *Robotics and Computer–Integrated Manufacturing*, (33):21–28.

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., and Haidegger, T. (2017). Towards a robots task ontology standard. In *Proc. ASME 2017 International Manufacturing Science and Engineering Conference*, Los Angeles, CA, USA.

Behnke, G., Bercher, P., Biundo, S., Glimm, B., Ponomaryov, D., and Schiller, M. (2015a). Integrating ontologies and planning for cognitive systems. In *Proc. 28th International Workshop on Description Logics (DL 2015)*. CEUR Workshop Proceedings.

Behnke, G., Ponomaryow, D., Schiller, M., Bercher, P., Nothdurft, F., Glimm, B., and Biundo, S. (2015b). Coherence across components in cognitive systems - on ontology to rule them all. In *Proc. 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 1442–1449. AAAI Press.

Benjamins, R., de Barros, L. N., and Valente, A. (1996). Constructing planners through problem-solving methods. At http://ksi.cpsc.ucalgary.ca/KAW/KAW96/benjamins/.

Bermejo-Alonso, J., Salvador, J., and Sanz, R. (2018). Towards an ontology for task and planning in autonomous systems: An emergency scenario. In *Advances in Intelligent Systems and Computing,*, volume 693, pages 429–440. Springer, Cham.

Bermejo-Alonso, J., Sanz, R., Rodríguez, M., and Hernández, C. (2011). An ontological framework for autonomous systems modelling. *International Journal on Advances in Intelligent Systems*, 3(3):211–225.

Bermejo-Alonso, J., Sanz, R., Rodríguez, M., and Hernández, C. (2013). Ontology engineering for the autonomous systems domain. In *Communications in Computer and Information Science*, volume 348, pages 263–277. Springer Berlin Heidelberg.

Borgo, S. and Masolo, C. (2010). *Ontological Foundations of Dolce*, pages 279–295. Springer Netherlands.

Bouillet, E., Feblowitz, M., Liu, Z., Ranganathan, A., and Riabov, A. (2007). A knowledge engineering and planning framework based on owl ontologies. In *Proceedings of the Second International Competition on Knowledge Engineering*.

Chandrasekaran, B., Josephson, J., and Benjamins, V. R. (1998). Ontology of tasks and methods. In *Proc. of the 11th Workshop on Knowledge Acquisition, Modeling and Management (KAW98)*, Banff, Alberta, Canada.

Colledanchise, M. and Ogren, P. (2018). *Behavior Trees in Robotics and AI: An Introduction*. Taylor and Francis.

Freitas, A., Schmidt, D., Panisson, A., Meneguzzi, F., Vieira, R., and Bordini, R. H. (2014). *Semantic Representations of Agent Plans and Planning Problem Domains*, volume 8758 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 351–366. Springer International Publishing.

Gangemi, A., Borgo, S., Catenacci, C., and Lehmann, J. (2005). Task tanomies for knowledge content. Technical Report D07, Laboratory for Applied Ontology.

Gil, Y. (2005). Description logics and planning. *AI Magazine*, 26(2):73–84.

Gil, Y. and Blythe, J. (2000). PLANET: a shareable and reusable ontology for representing plans. In *Proc. of the AAAI Workshop on Representational Issues for Real-world Plannning Systems*, pages 28–33.

Hartanto, R. (2009). *Fusing DL Reasoning with HTN Planning as Deliberative Layer in Mobile Robotics*. PhD thesis, Universität Osnabrück.

Hartanto, R. and Hertzberg, J. (2009). On the benefit of fusing dl-reasoning with htn-planning. In B. Mertsching, M. H. and Aziz, Z., editors, *Lecture Notes in Artificial Intelligence*, volume 5803, pages 41–48. Springer-Verlga Berlin Heidelberg.

Moreno, R., Borrajo, D., and Meziat, D. (2000). Process modelling and AI planning techniques: A new approach. In *Proc. 2nd International Workshop on Information Integration and Web-based Applications and Services*, Indonesia.

Motta, E. (1997). An overview of the OCML modelling language. Technical report, KMI.

OMG (2015a). OMG systems modeling language (SysML) version 1.4.

OMG (2015b). Unified modeling language (UML) version 2.5 modeling language (SysML) version 1.4.

Rajhpathak, D. and Motta, E. (2004). An ontological formalization of the planning task. In *Proc. of the International Conference on Formal Ontology in Information Systems (FOIS)*, Torino, Italy.

Sánchez-Ruiz, A., González-Castro, P. A., and Díaz-Aguado, B. (2007). Planning with description logics and syntactic updates.

Sirin, E., Parsia, B., Wu, D., Hendler, J., and Nau, D. (2004). HTN planning for web service composition using SHOP2. *Journal of Web Semantics*, 1(4):377–396.

Tate, A. (1996). Towards a plan ontology. *AI*IA Notizie (Journal of the Italian Association for AI*, 9(1).

Tate, A. (1998). Roots of SPAR - shared planning and activity representation. *The Knowledge Engineering Review*, 13(1):121–128.

Thorisson, K. R., Bieger, J., Thorarensen, T., Sigurdardottir, J. S., and Steunebrink, B. R. (2016). Why artificial intelligence needs a task theory and what it might look loke. In *Artificial General Intelligence*, number 9782 in LNAI, pages 118–128. Springer International Publishing.

Valente, A. (1995). Knowledge-level analysis of planning system. *SIGART Bulletin*, 6(1):33–41.

W3C (2012). *OWL 2 Web Ontology Language*, second edition edition.