



UWS Academic Portal

A review and comparison of ontology-based approaches to robot autonomy

Olivares-Alarcos, Alberto; Bessler, Daniel; Khamis, Alaa; Goncalves, Paulo; Habib, Maki K.; Bermejo-Alonso, Julita; Barreto, Marcos; Diab, Mohammed; Rosell, Jan; Quintas, Joao; Olszewska, Joanna Isabelle; Nakawala, Hirenkumar; Pignaton, Edison; Gyrard, Amelie; Borgo, Stefano; Alenya, Guillem; Beetz, Michael; Li, Howard

Published in:
The Knowledge Engineering Review

DOI:
[10.1017/S0269888919000237](https://doi.org/10.1017/S0269888919000237)

Published: 27/12/2019

Document Version
Peer reviewed version

[Link to publication on the UWS Academic Portal](#)

Citation for published version (APA):

Olivares-Alarcos, A., Bessler, D., Khamis, A., Goncalves, P., Habib, M. K., Bermejo-Alonso, J., Barreto, M., Diab, M., Rosell, J., Quintas, J., Olszewska, J. I., Nakawala, H., Pignaton, E., Gyrard, A., Borgo, S., Alenya, G., Beetz, M., & Li, H. (2019). A review and comparison of ontology-based approaches to robot autonomy. *The Knowledge Engineering Review*, 34, [e29]. <https://doi.org/10.1017/S0269888919000237>

General rights

Copyright and moral rights for the publications made accessible in the UWS Academic Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please contact pure@uws.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

A Review and Comparison of Ontology-based Approaches to Robot Autonomy

Alberto Olivares-Alarcos^{1,*}, Daniel Beßler^{2,*}, Alaa Khamis³, Paulo Goncalves⁴, Maki K. Habib⁵, Julita Bermejo-Alonso⁶, Marcos Barreto⁷, Mohammed Diab⁸, Jan Rosell⁸, João Quintas⁹, Joanna Olszewska¹⁰, Hirenkumar Nakawala¹¹, Edison Pignaton¹², Amelie Gyrard¹³, Stefano Borgo¹⁴, Guillem Alenyà¹, Michael Beetz², and Howard Li¹⁵

¹*Institut de Robòtica i Informàtica Industrial, CSIC-UPC Llorens i Artigas 4-6, 08028 Barcelona, Spain.*
E-mail: aolivares@iri.upc.edu

²*Institute for Artificial Intelligence, University of Bremen, Germany.*
E-mail: danielb@uni-bremen.de

³*Centre for Pattern Analysis and Machine Intelligence, University of Waterloo, Canada.*

⁴*IDMEC, Instituto Politécnico de Castelo Branco, Portugal.*

⁵*The American University in Cairo, Egypt.*

⁶*Universidad Isabel I, Burgos, Spain.*

⁷*Computer Science Dept., Federal University of Bahia, Brazil.*

⁸*Institute of Industrial and Control Engineering, Universitat Politècnica de Catalunya, Barcelona, Spain*

⁹*Instituto Pedro Nunes, 3030-199 Coimbra, Portugal.*

¹⁰*University of the West of Scotland, UK.*

¹¹*Department of Computer Science, University of Verona, Italy.*

¹²*Informatics Institute, Federal University of Rio Grande do Sul, Brazil.*

¹³*Knoesis, Wright State University, USA.*

¹⁴*Laboratory of Applied Ontology IST C-CNR, Trento, Italy.*

¹⁵*University of New Brunswick, Canada.*

**Both authors contributed equally to this manuscript.*

Abstract

Within the next decades, robots will need to be able to execute a large variety of tasks autonomously in a large variety of environments. To relax the resulting programming effort, a knowledge-enabled approach to robot programming can be adopted to organize information in re-usable knowledge pieces. However, for the ease of re-use, there needs to be an agreement on the meaning of terms. A common approach is to represent these terms using ontology languages that conceptualize the respective domain. In this work, we will review projects that use ontologies to support robot autonomy. We will systematically search for projects that fulfill a set of inclusion criteria, and compare them with each other with respect to the scope of their ontology, what types of cognitive capabilities are supported by the use of ontologies, and which is their application domain.

1 Introduction

In the next decades, research and industry are expected to develop a large variety of autonomous robots for a large variety of tasks and environments. The envisioned robots include, for example, domestic service robots preparing meals, setting the dinner table, and cleaning it up; agriculture robots monitoring fields, analyzing the growth of plants, detecting and classifying plant diseases, cultivating the fields, and collecting the harvest; and robots in retail stores doing the inventory, and replenishing the shelves if needed. One of the biggest challenges is that every combination of task, robot, and environment requires a specific robot control program.

One way to make the resulting programming effort feasible is to adopt a *knowledge-enabled robot programming* paradigm (Beetz, Jain, et al. 2012). The basic idea of knowledge-enabled programming is to separate knowledge from the program, and to modularize it into small broadly applicable chunks, such as *grasp objects by their handles, hold filled containers upright, do not squeeze breakable items too hard*, etc. For example, the program would ask: *How should I pick up the object in front of me?* A reasoning mechanism would then collect the relevant knowledge units and combine them in order to propose the appropriate grasp.

The advantage of knowledge-based approaches to robot programming is that these knowledge pieces apply to many applications and can dramatically accelerate the realization of new robot applications. For instance, the knowledge about safety in industrial collaborative robotic environments is general and applies to several different tasks that humans and robots can perform cooperatively. Therefore, knowledge-enabled programming empowers programmers to write control programs that are fairly independent of tasks, robots, and environments, and to state the necessary knowledge about the tasks, robots, and environments as separate modular knowledge bases. Particularly promising is the concept of web-based knowledge services such as RoboEarth (Waibel et al. 2011) and openEASE (Beetz, Tenorth, and Winkler 2015) that enable robots to collect and share knowledge of many robots.

The sharing of knowledge between different robots, and between different subsystems of a single robot’s control system requires that concepts used for communicating knowledge are machine understandable and consistent. Consider, for example, the knowledge chunk *grasp objects by their handles*. This knowledge chunk can only be effectively transferred between different robot applications if the robot generating the knowledge chunk, and the one using it agree on the meaning of *grasping*, *objects*, and *handles*. Thus, it is needed to define the conceptualization of these terms in a machine-understandable form that is accepted by robots generating and using this knowledge. This is exactly what ontologies are for.

In this article, we present a review on works about ontologies for autonomous robot’s applications. We are interested in particular frameworks that employ ontologies to support robot autonomy rather than on particular ontologies that are designed for robots.

The presented work has been conducted in the context of the IEEE-SA P1872.2 Standard for Autonomous Robotics (AuR) Ontology, originated as a sub-group of IEEE WG *Ontologies for Robotics and Automation* (ORA). The IEEE ORA WG started as a study group in early 2011 and became an official working group in November 2011. Since the beginning, IEEE ORA WG was divided into subgroups which were each in charge of studying a specific robotics and automation sub-domain, such as robotic tasks¹, and autonomous robotics².

There have been several other authors discussing literature about topics related to ontologies and robotics. Some of which are focused on particular robotic tasks, and compare different knowledge-based approaches with respect to their suitability for the investigated task. For example, Thosar et al. (Thosar et al. 2018) investigate the suitability of nine robot knowledge bases for the task of replacing missing objects, that play a role in a task (e.g. a tool), with similar ones. Our work here is similar in that we also look into different robot knowledge bases, but not from the perspective of a specific task but rather from the more general viewpoint of how ontologies support robot autonomy. Another similar study is presented by Paulius et al. (Paulius and Sun 2018) where the authors provide a discussion on different knowledge representation principles with respect to how the knowledge is acquired, represented, and re-used in robotic systems. Based on this discussion, the authors argue that a set of key-components must be implemented for any ”sufficient” knowledge representation for robots. The purpose of the work presented in our article is rather to draw a landscape of how ontologies are used to support robot autonomy, instead of proposing a list of key functionalities.

The contributions of this work are:

¹https://standards.ieee.org/project/1872_2.html

²https://standards.ieee.org/project/1872_2.html

- A classification of ontology-based approaches to robot autonomy; and
- a discussion of existing approaches in autonomous robotics that use ontologies to support autonomy, and a comparison of them.

The paper is structured as follows. In Section 2, we provide background information about the two disciplines in whose intersection this work revolves: ontologies and autonomous robots. In Section 3, we discuss a classification scheme for approaches that use ontologies to support robot autonomy, which is used to compare several approaches that fulfill a set of inclusion criteria in Section 4. Finally, the paper concludes with a discussion in Section 5.

2 Basics of ontology and autonomous robotics

In this section, we first introduce ontology as a knowledge artifact, and list some ontology classifications that are relevant for later discussion. Then, we review the terminology used in autonomous robotics, and some of the (computational) problems and capabilities that are considered essential for agent's autonomy.

2.1 Ontologies

One of the first attempts to define ontology was made by Gruber in (Gruber 1993) stating: *an ontology is an explicit specification of a conceptualization*. Gruber's definition was informal and several authors tried to refine it either by specifying what a conceptualization is (Guarino and Giarretta 1995), or by adding further requirements like being *formal* and *shared* (Borst, Akkermans, and Top 1997; Studer, V. Benjamins, and Fensel 1998). We had to wait another decade to settle the problem. In Guarino et al. (Guarino, Oberle, and Staab 2009) a technical discussion of the terminology was presented and a formal definition of ontology was finally given.

Starting from Gruber's intuition that an ontology depends on a conceptualization, Guarino et al. explain that a conceptualization is (mathematically speaking) an intentional relational structure, i.e., a domain of discourse (a set of entities), a set of possible worlds (possible layouts of the entities), and a set of relations (stating which properties entities have in each possible world). Once the formal notion of conceptualization is given, it suffices to fix a vocabulary and the ontological principles one commits to, the so-called *ontological commitments*. The purpose of the commitments is to constrain the interpretation of the vocabulary to the given conceptualization. At this point, an ontology is defined to be a *logical theory* consisting of a set of formulas whose models approximates as well as (formally) possible the intended models, i.e., those models that satisfy the conceptualization and the ontological commitments.

Since an ontology is a logical theory, it consists of individuals, classes, functions, relations and axioms. The exact list changes depending on the specific logic language one adopts. Usually, an ontology is given in First-Order Logic (FOL), or in Web Ontology Language (OWL). *Individuals* are the objects in the ontology, the things the ontology is about. *Classes* are properties and are used to identify the individuals that satisfy that property. *Functions* provide ways to identify and relate individuals. *Relations* are connections across individuals. *Axioms* are expressions in the language that use the previous elements to state what is true in the ontology.

In domain studies, the term ontology is used to refer to a variety of things. For instance, for Chandrasekaran et al. (Chandrasekaran, Josephson, and V. R. Benjamins 1998) an ontology is a *representation vocabulary* specialized to some domain and constrained by a conceptualization. It is also understood as a *domain theory* about objects, properties and relationships among those objects that are possible within a specified domain of knowledge. The purpose of an ontology in this sense is to provide the knowledge structure for a particular domain, therefore, it focuses only on the viewpoint taken within the domain, and it includes the relevant concepts for working in such domain. In the literature, this latter use of the term is known as *domain ontology* while the characterization introduced by Guarino and colleagues (Guarino, Oberle, and Staab 2009) is

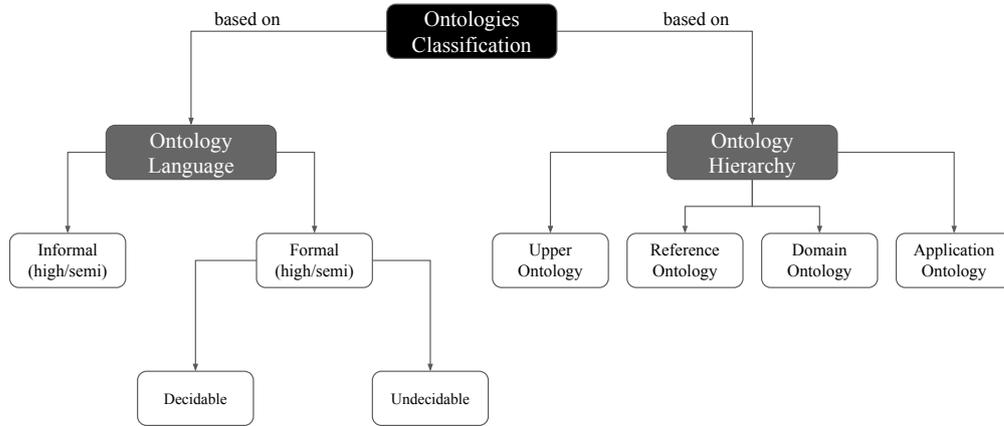


Figure 1 Different classifications of ontologies explained in this work. They are based on: (a) the language used to write the ontology and (b) the hierarchical level of the ontology.

general encompassing domain as well as foundational ontologies that contain very general terms applicable across all domains.

Developing a domain ontology, one provides the description of a particular domain without challenging the ontological perspective. The purpose is to make the domain knowledge explicit and formal, i.e., to fix in a formal language the vocabulary and what the experts' consider its correct interpretation including the valid assertions in the domain. A domain ontology can help to achieve data and model interchangeability within and across communities.

It is worth noting that there is no unique conceptualization of a domain. Having suitable domain ontologies helps to clarify the differences as well as to compare the conceptualizations.

2.1.1 Types of ontologies

Ontologies can be classified along many dimensions. In the Figure 1, it is depicted a graph containing the two classifications of ontologies considered along this work.

A classification based on the characteristics of the language used for the ontology is presented by Uschold and Gruninger (M. Uschold and Gruninger 1996). It shows that the term 'ontology' is sometimes used vaguely. That classification divides ontologies in four classes: (a) highly informal, (b) semi-informal, (c) semi-formal and (d) rigorously formal. However, following our previous discussion, we observe that the first class is not talking about ontologies: it refers to linguistic resources or to knowledge repositories in an early phase of ontology construction.

Since the language of the ontology constrains how the ontology can be used in an information system, this kind of classification is of primary relevance in the context of this paper. The first distinction we introduce is between informal and formal languages. By *informal* we mean a language which does not have an associated formal semantics, like Resource Description Framework (RDF), (part of) Unified Modeling Language (UML) and Business Process Modeling Notation (BPMN). They are mostly dedicated to representation tasks and syntactic manipulation. Automatic reasoning on these languages is not reliable because there is no systematic way to constrain their interpretation. By *formal* we mean a language endowed with formal (e.g., Tarskian) semantics, that is, languages whose interpretation is formally established. These languages, among which we find FOL and OWL, are suitable for knowledge representation and reasoning since they

are based on clear and exhaustive syntactic and semantic rules. They are the most reliable languages we have available in science. Among the formal languages, a further distinction is of interest to us: decidable vs undecidable languages. Here decidable means that, given a logical theory expressed in that language, there exists a method for determining whether an arbitrary formula is derivable or not in the theory. Since ontologies are special logical theories, an ontology written in a decidable language is decidable. A language is called undecidable if it is not decidable. An ontology that uses a decidable language is also called *computational* since it can be used at run-time for information extraction and verification, e.g., OWL DL. A formal ontology, which is not computational, like FOL and OWL Full, is not appropriate for such role since, when queried, it might not return an answer. Since in application purposes the answer to a query might need to be available very quickly, decidability is enriched with computational complexity considerations (Papadimitriou 2003).

Ontology classifications usually divide ontological systems in upper-level, reference, domain and application. An *upper-level ontology* is an ontology that focuses on widely applicable concepts like object, event, state, quality, and high-level relations like part hood, constitution, participation, dependence. Examples are SUMO (Suggested Upper Merged Ontology) (Niles and Pease 2001), Cyc ontology (Lenat and Guha 1990), BFO (Basic Formal Ontology) (Arp, Smith, and Spear 2015) and DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) (C. Masolo et al. 2003).

A *reference ontology* is an ontology that focuses on a discipline with the goal of fixing the general terms in it. It is highly reusable within the discipline, e.g., medical, engineering, enterprise, etc. (Guarino 1998). When the ontology focuses in a more limited area, e.g. manufacturing or tourism, we call it a *domain ontology*. This kind of ontology provides vocabulary about concepts within a domain and their relationships, about the activities taking place in that domain, and about the theories and elementary principles governing that domain. The concepts in domain ontologies are mostly specializations of concepts already defined in upper-level and in reference ontologies, and the same might occur with the relations (Gómez-Pérez, Fernández-López, and Corcho 2004). An *application ontology* contains all the definitions needed to model the knowledge required for a particular application, e.g., a CAD/CAM system or an ERP. Another major classification system is useful to differentiate reference and domain ontologies, and focuses on topic coverage. The Enterprise ontology (Mike Uschold et al. 1998), the Process Specification Language (Grüninger 2004) and CORA (Schlenoff et al. 2012) are examples that fall into this classification. There are different ways to classify topics as shown by the variety of library classification systems around the world.

2.2 Autonomous robotics

Autonomy is a desirable quality for robotic agents in many application domains, especially when the robot needs to act in real-world environments together with other agents, and when the environment changes in unforeseeable ways. Robot autonomy is further critical when employed under certain legal and ethical constraints (e.g., a robot assistance at the hospital, or an industrial collaborative robot cooperating with humans). Apart from the dictionary and subjective definitions, there exist several attempts to define the term. However, no broad consensus on this matter has been reached so far. Beer et al. (Beer, Fisk, and W. A. Rogers 2014) present a comprehensive analysis of existing definitions in several domains including robotics. The definition of *autonomy* provided by the authors is the following one:

The extent to which a robot can sense its environment, plan based on that environment, and act upon that environment with the intent of reaching some task-specific goal (either given to or created by the robot) without external control

Computational problems and capabilities Autonomous systems can be based on different architectures with different levels of complexity ranging from simple reactive architectures, to deliberative architectures to cognitive architectures. Reactive systems are based on simple *sense-act* loops, while deliberative systems employ more sophisticated *sense-decide-act* loops to endow the system with reasoning and decision making capabilities. However, it is still an open question what the computational capabilities are that enable human-level cognition, and how these are structured in a cognitive architecture. Vernon et al. provide a thorough discussion about this topic (Vernon, Metta, and Sandini 2007). The authors present a broad survey of the various paradigms of cognition, addressing cognitivist approaches (physical symbol systems), emergent, connectionist, dynamical, and enactive systems, and also efforts to combine the different approaches in hybrid systems. Then, a review of several cognitive architectures drawn from these paradigms is surveyed. An extension of this survey is provided by Vernon (Vernon 2014). In this last work, Vernon et al. referred to the key architectural features that systems capable of autonomous development of mental capabilities should exhibit (Langley, Laird, and S. Rogers 2009).

In this work, we follow Langley’s thoughts, precisely concerning about *what* are the functional capabilities an autonomous robot should demonstrate, more than *how* those functional modules should interconnect (ibid.). Therefore, instead of proposing a novel cognitive architecture, which is out of the scope of this paper, we only make sure to address all terms related to those functional capabilities provided in Langley’s work, which are explained in the Section 3.2 and listed below:

1. Recognition and categorization;
2. Decision making and choice;
3. Perception and situation assessment;
4. Prediction and monitoring;
5. Problem solving and planning;
6. Reasoning and belief maintenance;
7. Execution and action;
8. Interaction and communication; and
9. Remembering, reflection, and learning.

3 A classification of ontologies for autonomous robots

In this section we present the classification which will be utilized to structure and perform the review of the selected works in Section 4. The classification is split into three dimensions: (a) ontology scope, (b) reasoning scope and (c) application domain scope.

3.1 Ontology scope

Ontologies can be organized as networks of modules, which can themselves be ontologies, each focusing on a specific topic. The scope of a module, is given by the range of categories that it covers. In this section, we propose a list of categories particularly relevant in autonomous robotics such as **Sensor**, **Capability**, and **Action**. Our aim is to find how these categories have been used in the literature; and to provide an initial discussion about their actual meaning. References to more detailed discussions in the literature are also provided. The Oxford dictionary (Oxford-University 2019) is used to provide informal definitions, to discuss how the terms are understood in common and conventional ontologies, and to highlight the different usage in the robotics field. In Section 4.3.1, we analyze whether or not each of the surveyed projects defines these categories and how.

Note that many concepts relevant for autonomous robotics lack a universally agreed meaning. Moreover, terms commonly used in the robotics field are also often part of everybody’s everyday speech. Hence, everybody has *some* intuitive definition of these terms that is often heavily influenced by personal experience, and thus it may be substantially different from that of other

people. In this article, we will only cover the most widely agreed meanings of concepts, and provide a brief comparison between the different viewpoints.

Object The Oxford dictionary defines the term **Object** as *a material thing that can be seen and touched*. But often a more fine-grained definition is needed that is not limited to material objects, e.g. holes, or to things in physical space, e.g. behavioral patterns. More generally, mental and social objects depend on material acts (like brain activities and communication acts) but they may be neither material (made of matter) nor physical (located in a region of space). A significant number of foundational ontologies make a distinction between **Endurants** and **Perdurants** (C. Masolo et al. 2003; Niles and Pease 2001). Endurants (aka continuants or objects) are wholly present at any time, but may change over time. Perdurants (aka occurrents or events), on the other hand, are extended in time, and only partially present at any time. This dichotomy is crucial in systems that have to cope with time. Physical objects are often further classified into **Artifact** and **Non-Artifact**, where artifacts are intentionally created, often according to a design, to fulfill a certain function, etc (Borgo, Franssen, et al. 2014). Objects may further be classified as **Agent** or **Non-Agent**, where agents are capable to generate intentional behavior.

We humans tend to categorize objects because of the variety of **Qualities** and **Properties** that they exhibit, which give us a way to cluster them into similarity classes. There has been long philosophical discussions about what qualities and properties are, and among them which are primary or not, and if they can be exhaustively listed. One important reason to focus on qualities and properties is the understanding of (qualitative) change. One branch of formal models considers individual qualities (roughly, the way an individual manifests characteristics like weight, size, shape etc.) as basic entities in the ontology. Each individual quality is existentially dependent on an unique endurant (or perdurant) and associated to a quale (plural: qualia). Qualia are used to compare entities, and thus to discuss similarity/dissimilarity (w.r.t. the associated quality) across objects and events (Claudio Masolo and Borgo 2005). In this view, qualities form a third fundamental category along with endurants and perdurants, and the associated qualia change over time to explain the changes in their corresponding endurants (or perdurants). Qualia are further organized in **Spaces** (e.g. the space of weight, the space of colors etc.) and can be given a quantitative/qualitative value (e.g. numerical) once the space is enriched with a reference system and unit of measure. An alternative approach uses the notion of **Tropes** (also called ‘abstract particulars’) where qualitative change is expressed through the substitution of tropes (Neuhaus, Grenon, and Smith 2004). Thus, when an object changes, this modeling view assumes that the existing trope ceases to exist and a new one is created. Continuous change (like the increasing of room temperature) is often considered problematic to model in this latter approach.

Environment Map The term **Environment** is defined in the Oxford dictionary as *the surroundings or conditions in which a person, animal, or plant lives or operates* while **Map** is described as *a diagrammatic representation of an area of land or sea showing physical features, cities, roads, etc.* Although some general way to understand the meaning of environment in robotics has been proposed (Borgo, Cesta, et al. 2019), in this domain the focus is more often oriented to the representation of the environment (Chella et al. 2002). The format and information content of a map is not only diagrammatic in robotics as it is influenced by how and for what the map is to be used by the robot. Collision maps, for example, encode 3D geometric information of the environment to support generating collision free motions in 3D space, while navigation maps usually only use a 2D geometrical representation to support finding collision free navigation paths. The term **Semantic Environment Map** is often used to refer to environment representations that make explicit the semantics of the environment and objects in the environment. One prominent example, called SOM (Semantic Object Maps), was introduced by Rusu et al. (Rusu et al. 2009). SOM’s also encode spatial information about the environment but, in addition, enrich

the information content with encyclopedic and common-sense knowledge about objects, and also include knowledge derived from observations.

Affordance The term **Affordance** was introduced by Gibson as *what the environment offers the animal, what it provides or furnishes, either for good or ill* (Gibson 1979). More recently, the meaning has shifted towards *"(perceived) possibility for action"* (Norman 2002), i.e., something the object offers that allows the agent to interact with it or, more generally, something that allows objects to participate in actions or processes. However, there is no common agreement in the ontology engineering community on how this concept should be modeled. One way to model affordances is as individual qualities of an object (Ortmann and Kuhn 2010), or as relational qualities of a pair object-agent (Turvey 1992). Another approach is to model them as events, as proposed by Moralez (Moralez 2016). A notion of affordance is relevant to talk about possibilities as it enables to answer questions such as *"what can the robot do with an object?"*, and *"is it possible for an object to take a particular role when some task is performed?"*.

Action and Task There has been a lot of confusion about the meaning of the words **Task** and **Action**, and how these relate to each other. The Oxford dictionary defines an action as *the fact or process of doing something, typically to achieve an aim*. There have been several attempts to define action in different disciplines. A notable one is Donald Davidson's philosophy of action, where he defined an action as something *intentional under some description* (Davidson 2001). Krüger and colleagues surveyed the meaning of action in the robotics field (Krüger et al. 2007), and argued that a notion of action in robotics needs to take into account several aspects including perception, actuation, embodiment and learning.

A task can be understood *as a piece of work that has to be done* (Oxford-University 2019). Hence, tasks denote pending work, independently from how an agent exactly accomplishes this work. In this view, an action would be a way to execute a task. Technically, one can approach this by defining tasks as types (of event) used to classify actions, which then allows to explicate that a task can be accomplished in different ways, and to talk about individual tasks independently from their possible executions. A notion of task in robotics has been proposed recently by Balakirsky et al. (Balakirsky et al. 2017).

Tasks and actions may further be classified according to their complexity, temporal extension, inter-task (inter-action) relationships, etc. However, such classifications are often not clear, e.g., the distinction simple vs. complex task would be dependent on the adopted granularity or robot's capabilities.

Activity and Behavior The Oxford dictionary defines an **Activity** as *the condition in which things are happening or being done*. The term **Behavior**, on the other hand, is defined as *the way in which one acts or conducts oneself, especially towards others*. Hence, both terms refer to situations in which an agent performs actions, but with different viewpoints. Activities rather having an intrinsic, and behaviors an extrinsic viewpoint, e.g., was it good or bad behavior, how it affected other agents, and so on. Note that in the case of behavior, it can also apply to non-agents as it is common to talk of the behavior of devices or tools, for instance.

In the 80's, Rodney Brooks and his colleagues did fundamental work in the field *behavior-based robotics* where the term behavior also refers to extrinsic characteristics of task execution. The field of behavior-based robotics is motivated by the observation that complex behavior can be generated by simple control systems, and that intelligence lies in the eye of the observer (Brooks 1991). Brooks has also postulated that the world is its own best model, and hence argues that simple *Sense-Act* loops can be used to directly interact with the world without relying much on symbolic representations.

Other authors have focused on the terms **Behavior** and **Function**, for instance claiming that the function of an object denotes its intrinsic aspects (i.e., how it works), and behavior the

extrinsic aspects (i.e., what it does). An engineering discussion of this dichotomy is provided by Salustri for the context of computer-based design tools (Salustri 2000) while an ontological assessment is provided by Mizoguchi et al. (Mizoguchi, Kitamura, and Borgo 2016).

Plan and Method A **Plan** is a *detailed proposal for doing or achieving something* (Oxford-University 2019). Similarly, the DOLCE+DnS Plan Ontology (Gangemi, Borgo, et al. 2004) defines **Plan** as a *description that defines or uses at least one task and one agentive role or figure, and that has at least one goal as a part*. Hence, plans have explicit goals to be achieved when the plan is executed by appropriate sequences of actions that comply with the plan. An execution of the plan can succeed, fail, be postponed, aborted, etc.

The generation and assessment of plans is a long-standing sub-area of AI. A prominent approach is the Planning Domain Definition Language (PDDL) (McDermott et al. 1998). PDDL *tasks* denote initial and goal state, and how the state can be modified by applying actions or operators. General purpose solvers are then used to generate a plan given the domain definition. Several authors have further combined standard planning techniques, such as PDDL, with more expressive representations. A survey about these approaches is provided by Gil (Gil 2005).

A **Method**, on the other hand, is more abstract than a plan. The Oxford dictionary defines it as a *particular procedure for accomplishing or approaching something, especially a systematic or established one*. In a sense, *methods* are guidelines for agents to choose actions towards achieving a specific goal instead of specifying beforehand an explicit sequence of actions that would cause the goal to be achieved.

Capability and Skill According to the Oxford dictionary, **Capability** is *the power or ability to do something*. Hence, a distinction is made between capabilities that are enabled by physical qualities and those that are enabled by social role(s) within a certain community. The term **Skill**, according to the Oxford dictionary, is more restrictive, namely, *the ability to do something well*. Thus, it only includes what the agent can do because of its physical qualities and, in addition, it implies that the achievement is positively qualified (in terms of manners and results) (Fazel-Zarandi and Fox 2013). One widespread use of the term in robotics is *skill learning* where it is used to refer to the ability of the robot to achieve something via a behavior learned through observation, communication, experimentation or simulation. However, both terms are also often used as synonym of each other, for example by Perzylo and colleagues in their work on the description and orchestration of manufacturing skills (Perzylo, Grothoff, et al. 2019).

Having capabilities represented in a formal model, the robot can reason about whether the necessary capability is present to perform a certain task in a given situational context and, if not, how the task could be accomplished otherwise. This is usually approached by defining capabilities with respect to hardware and software components of the robot (Kunze, Roehm, and Beetz 2011; Buehler and Pagnucco 2014). A navigation capability would be enabled by a mobile base which is controlled by a navigation software component that interfaces with the mobile base. Tididi et al. have used a notion of capability to provide a more intuitive, capability-based interface to control robots (Tididi et al. 2017).

Capabilities may not be manifested in arbitrary situations. For example, wheeled robots are not able to navigate along stairs and thus might not be able to reach a target location on another floor. However, if an elevator can be used and the robot is able to operate it, the robot may still be able to reach its navigation goal. Hence, capabilities do not automatically enable the robot to perform a task. Their use depend on suitable conditions of the situational context in which the robot should operate – e.g. who can perform the task, what specific variant of the task can be performed, and where the task can be performed. The degree of how capable a robot is may change over time to the point that a capability cannot be manifested at all, for example, due to attrition of hardware, broken hardware or missing (hardware or software) components.

Hardware components Ontologies may be used to explicate what chains of robot links and joints form what `BodyParts`, and how body parts can contribute to perform, for instance, tasks and capabilities.

One of the most widely used formats to represent hardware components of robotic agents is the *Unified Robot Description Format* (URDF)³. URDF allows to represent kinematic chains made of links and joints, and also to define the limits of each joint. This information is used, e.g., by inverse kinematics solvers to find a valid joint configuration in which the end-effector of the robot reaches a dedicated goal pose⁴. URDF files include both *actuators* (e.g. servos of the joints, grippers, etc.) which act in the environment and *sensors* (e.g. cameras, sonars, etc.) which used to perceive the environment.

A **Sensor** is a device which detects or measures a physical property and records, indicates, or otherwise responds to it (Oxford-University 2019). Sensors can be used for different objectives such as measuring robot parameters for control loops, correcting for errors in the robot’s models of itself and of the world, detecting and avoiding failure situations, among others. The Semantic Sensor Network (SSN) is an ontology for describing sensors and their observations, the involved procedures, the studied features of interest, the samples used to do so, and the observed properties, as well as actuators (Compton et al. 2012). SSN includes a lightweight but self-contained core ontology called SOSA (Sensor, Observation, Sample, and Actuator).

Software components A notion of robot software components in ontologies is crucial when these shall be automatically introspected and integrated into task execution. One of the most widely employed ontologies for modeling software in ontologies is the Ontology of Information Objects (IO) (Gangemi, Borgo, et al. 2004) where a distinction is made between an abstract `DataStructure` and the `DigitalResource` that concretely realizes the data structure within some physical storage medium. The broad goals for software ontologies in robotics are to enable the robot automated software discovery and installation to dynamically compose its control system for a given task, to decide for a given control system whether some capability can be realized by invoking some of the software components, and to support introspection in case some software failure occurred.

One of the most widely used middlewares in robotics nowadays is the Robot Operating System (ROS)⁵. ROS organizes robot software components in a communication graph, where each node is a piece of software either listening or publishing messages on named topics, or offering a service that can be called via the node. Messages are defined using an abstract syntax, and concrete realizations of the message type in different target languages such as C++ and Python are generated automatically by ROS.

Interaction and Communication *Interaction is a reciprocal action or influence* (Oxford-University 2019) between two or more entities. This comprehensive definition includes those interactions in which there is not an explicit exchange of information. For example, interactions happening at atomic level or stigmergic interaction, through the environment in which agents act. Work in robotics tends to concentrate on information exchange. Indeed, in the literature, both the Human-Computer Interaction (Dix 2009) and the Human-Robot Interaction (A. Yanco and J. Drury 2004), (Yanco and J. L. Drury 2002) domains, tend to provide a less general formal definition of **Interaction** which may be closer to **Communication**.

The term **Communication** *is the imparting or exchanging of information by speaking, writing, or using some other medium* (Oxford-University 2019). Gangemi et al. (Gangemi and Mika 2003) proposed to formalize this term within the *Description & Situation* Ontology viewpoint

³<http://wiki.ros.org/urdf>

⁴An end-effector is a device located at the end of a kinematic chain, designed to interact with the environment. It’s task depends on the application of the robot.

⁵<http://www.ros.org/>

distinguishing two cases: (a) an ontology for communication situations and roles, and (b) an ontology for peer-to-peer communication.

3.2 Reasoning scope

This scope of reasoning is our second classification criterion for the comparison between ontology-based approaches in autonomous robotics. In the remainder of this section, we provide a categorization of ontology-based reasoning tasks that are in particular relevant for autonomous robotics, and that have been considered in previous works. Indeed, this categorization is based on the nine functional capabilities presented in Section 2.2 which every autonomous robot should exhibit (Langley, Laird, and S. Rogers 2009). In Section 4.3.2, we discuss how the surveyed projects use ontologies to support each of these nine capabilities.

One of the keys to the success of knowledge-based approaches in autonomous robotics, is the use of ontologies as *enabler* to help the robot to understand and reason about its environment when executing tasks. For instance, implementing robotic applications that would not be possible without the use of KR techniques, and that substantially enhance the state of the art in autonomous robotics. The main research question is how the different software components of an integrated robot control system could be enhanced through the use of ontologies and automated reasoning.

Recognition and categorization For the purpose of establishing a contact between its environment and its knowledge, a robot must be able to recognize events or situations (static and dynamic) and categorize them as named instances of already known patterns. For instance, let's consider a collaborative industrial robotic arm which picks pieces from a conveyor and places them on a table where a human operator can access to. The robot must recognize and categorize the conveyor and the different pieces to manipulate (static), as well as the human collaborator's movements and actions (dynamic). Both, recognition and categorization, operate on the output generated by perception systems and often are seen as a unique capability. Nevertheless, Langley et al. addressed them separately because both *can operate on abstract mental structures* (ibid.). The authors emphasize that in order to support recognition and categorization, a cognitive architecture shall provide a form to represent patterns and situations in memory.

Decision making and choice An autonomous robot requires the ability to choose among several alternatives, which usually is considered together with the recognition and categorization problem in a recognize-act cycle. Nonetheless, Langley et al. considered the capability of decision making independently. It is important not to mistake this capability with planning, whose focus is on the achievement of a goal and it will be explained later along this section. For example, a collaborative robot would apply decision making to choose between moving or not when the operator is close, while planning would be used to find the sequence or sequences of actions which are likely to lead to a successful collaborative task. A cognitive architecture should be able to represent the different choices in a format the robot understands. Indeed, that representation would also be used to improve the decision making through learning.

Perception and situation assessment The environment where the robot exists, must be sensed, perceived and interpreted. First, the robot senses its surroundings through possibly multi-modal sensors. Then, using the gathered information and relying on recognition and categorization, discussed earlier, and on inferential mechanisms, which will be covered shortly, the robot is able to perceive the environmental entities (e.g. objects and events). Finally, the situation assessment takes place when the perceived objects and events are interpreted. Following with the example used before, a collaborative industrial robot would look at the conveyor, the pieces and at the operator's movements to sense the environment. The pieces, their poses and other information would be recognized and categorized in order to assess the environmental situation

so that the robot could, for example, interpret that an approaching piece should be picked. Just as occurred with previous cognitive capabilities, the inherent knowledge of the whole process must be represented in a manner the robot understands. Note that the representation requires memory, a resource which is often limited. Hence, the notion of attention emerges, meaning that the robot not only has to perceive but also it could be asked to decide to focus only on a specific region of the environment.

Prediction and monitoring Prediction is a cognitive capability which requires the representation in memory of a model (e.g. ontology-based) of the environment, the actions that can take place and their effects. Therefore, the robot could predict future events and situations which did not occur yet by means of a proper mechanism which utilizes the representation. Applied to the collaborative robotics' example, it would be possible for the robot to predict the operator's actions, so that the robot could adapt better to what it is expected. Note that prediction enables robots to also monitor processes. When the perceived situation differs from the expected one, it means that either our knowledge is not complete or something did not go as it was supposed to. In the former case, it would be possible to store the facts in memory for posterior learning, in the later case, an alarm or error could be triggered. In the example, the robot could detect a malfunctioning in the conveyor if the pieces stopped arriving close to the robot's workspace.

Problem solving and planning In novel situations where robots are meant to achieve their goals, it is necessary for them to be able to plan and solve problems. For the purpose of generating a plan, the robot needs a model of the environment utilized to predict the effects of its actions. Furthermore, the cognitive architecture must be able to represent a plan as an (at least partially) ordered set of actions, their expected effects, and the manner in which these effects enable subsequent actions. Sometimes, a robot could have a memory with previous plans which could be re-used with and without further modifications. Note that it is also considered the case of having conditional actions and different branches which depend on the outcome of previous events. Despite often being viewed intimately related, planning is somewhat less general than problem solving. In particular, the former usually refers to cognitive activities within the robot's head, whereas the later can also occur in the world. Concretely, when a problem to be solved is complex and the available memory is limited, a robot may search for solutions by executing actions in the environment, rather than constructing a complete internal plan. As an illustration, a collaborative robot could solve a problem by mixing the execution of actions such as 'asking for operator's help' (external behavior) and the generation of actions' sequences (internal planning).

Reasoning and belief maintenance Reasoning is a cognitive activity which allows a robot to expand its knowledge state, drawing conclusions from other beliefs or assumptions the robot already maintains. Thus, it is required the existence of a representation of beliefs and the relationships among them. A common formalism used to encode such knowledge is first-order logic (FOL). Ontologies are often written in languages based on less expressive formalisms than FOL (e.g. OWL-description logic) in order to reduce the computational cost of inference. These formalisms, allow the use of different sorts of reasoning such as: deductive or inductive. For the robot of the previous example, it would be possible to infer how operators will react to unexpected interactions by knowing general information about human-robot interaction (deductive). Or the opposite, from specific operator's behaviors, inferring the norms to follow during human-robot interaction (inductive). Note that reasoning is not only relevant to infer new beliefs but also to decide whether to hold existing ones (belief maintenance). Such belief maintenance is especially important for dynamic environments in which situations may change in unexpected ways, with implications for the robot's behavior.

Execution and action Cognition takes place to support and drive activity in the environment. To this end, a cognitive architecture must be able to represent and store motor skills that enable such activity. In the example, a collaborative robotic arm should have skills or policies for manipulating its surroundings and for collaborating with other agents (e.g. humans). A robot should also be able to execute those skills and actions in the environment, what can happen in a reactive form. Nevertheless, a cognitive architecture should enable a robot to maintain a continuum loop of execution. Hence, the robot would be able to interpret how the execution of actions is affecting the state of the environment and could adapt its behavior. A proper representation of the ongoing actions occurring in the environment, is essential for aspects related to robot action execution: robot adaptation, new skills learning, action-execution-related knowledge, etc.

Interaction and communication Sometimes, the most effective way for a robot to obtain knowledge is from another agent (e.g. humans, robots, etc.), making communication another important ability that an architecture should support. Going back to the example used before, a collaborative robot could request for further information about how to perform a task or which are the preferences of the specific plant operator about where to place the picked pieces. Regardless of the modality or mean of communication, there should be a way to represent the transferred knowledge so that it is accessible to and understandable for the robot. Indeed, this should be bi-directional, meaning the robot must be able to transform stored knowledge into the concrete medium through which it will be communicated.

Remembering, reflection, and learning There are some capabilities which cut across those described before, whose use could enhance the performance of autonomous robots while not being strictly necessary for robot autonomy: remembering, reflection and learning. Remembering is the ability to encode and store the results (facts) of cognitive tasks so that they can be retrieved later. Once again, based on the previous example, a collaborative robot could store the results of an entire day of work (e.g. successful experiences, human-robot interactions, etc.). On the other hand, reflection stands for the *serious thought or consideration* (Oxford-University 2019) about something which usually is represented and stored in memory and can be retrieved. A collaborative robot could take into account a stored memory in order to explain which was the rationale behind its actions. Finally, learning, which usually involves generalization beyond specific beliefs and events. In the example, the collaborative robot would use the stored memories about successful and failed actions to generalize and learn from them. The knowledge used to learn might come from distinct sources, the observation of another agent, the result of previous experiences, or through kinesthetic teaching. No matter the source of experience, all of them require the existence of a memory in which the experiences are represented.

3.3 Application domain scope

The last classification criterion for the comparison between ontology-based approaches in autonomous robotics is regarding to the application domain. In this section, we comment some of the different application domains of robotics and the two principal domains we consider. In Section 4.3.3, we discuss for which of the two domains each of the reviewed projects has been designed and used.

Robotics is a multidisciplinary and versatile discipline whose application is present in wide range of domains: Medicine, Industry, Service, Entertainment, Space, Military. Therefore, it exists such a broad spectrum of application domains for robotics, that it is not possible to go through all of them without excessively extending this article. In accordance with this thought and following the classification of robotics devices published in the ISO 8373:2012⁶ (*Robot and Robotics Devices*

⁶<https://www.iso.org/standard/55890.html>

– *Vocabulary* 2012), we focus on two domains: Industrial and Service domains. These two domains include, if not all at least many, of the application sub-domains of interest for robotics (e.g. medicine, military, rescue, social, etc.). The ISO 8337:2012 specifies a vocabulary used in relation with robots and robotic devices operating in both industrial and non-industrial environments (service). It also provides definitions and explanations of the most commonly used terms. For instance, below find the definitions for Industrial and Service robot, which were partially extracted from the ISO.

Industrial Robots The term of *Industrial Robots* includes all those robots which are automatically controlled, reprogrammable, multipurpose manipulator, programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications⁷. Typical applications of industrial robots include welding, painting, assembly, pick and place for printed circuit boards, packaging and labeling, palletizing, product inspection, testing, and material handling. Industrial robots perform with high endurance, speed, and precision in all of those tasks.

According to International Federation of Robotics (IFR)⁸, in 2017 the total worldwide stock of operational industrial robots significantly increased by 15% to 2,097,500 units. Since 2012, the stock has been increasing considerably by 11% on average per year.

Service Robots The term of *Service Robots* means that robots perform useful tasks for humans or equipment excluding industrial automation applications. Where industrial automation applications include, but are not limited to, manufacturing, inspection, packaging, and assembly. Also note that while articulated robots used in production lines are industrial robots, similar articulated robots used for serving food are service robots⁷. Typical applications of service robots include those tasks which are dirty, dull, distant or dangerous. Service robots can be classified into *service robots for professional use* and *service robots for personal use*. Service robots for professional use are commonly used in different applications such as defense, intelligence, surveillance, reconnaissance, search & rescue, firefighting, healthcare, construction, logistics, smart factories, smart farming, hazardous environment monitoring and space exploration to name just a few. Applications of service robots for personal use include, but are not limited to home automation, personal assistant for elderly and physically challenged in support for aging in place, entertainment and vacuum cleaners.

The Executive Summary released by the International Federation of Robotics⁹, shows that the total number of service robots for personal and domestic use increased by 25% to about 8.5 million units in 2017. The value was up by 27% to US\$ 2.1bn. Moreover, in 2018, total unit sales of professional service robots are estimated to increase by 32% to about 165,000 units with a value of almost US\$ 8.7bn. From 2019 to 2021, almost 736,600 units will be sold.

4 Ontologies to support robot autonomy

In this section, we provide a discussion and comparison of frameworks that use ontologies to support robot autonomy. We restrict our investigation to projects that match a list of criteria that will be described in Section 4.1. For each framework that fulfills these criteria, we will provide a brief discussion in Section 4.2, followed by a comparison of them in Section 4.3. The section concludes by providing a short justification of why some relevant projects that do not fulfill all criteria were not included in Section 4.4.

⁷ Term defined along the ISO 8373:2012

⁸https://ifr.org/downloads/press2018/Executive_Summary_WR_2018_Industrial_Robots.pdf

⁹https://ifr.org/downloads/press2018/Executive_Summary_WR_Service_Robots_2018.pdf

4.1 Literature search and inclusion criteria

One of the goals of this work is to provide a systematic and fair comparison of projects located at the intersection of the fields autonomous robotics and ontologies. In order to select a potential list of candidates for discussion, we have followed a systematic examination of the state of the art, and, in addition, we have filtered the results by a set of inclusion criteria. In this section, we discuss the search procedure, and provide a list of criteria that need to be fulfilled by considered approaches.

4.1.1 Literature search

For the purpose of finding literature focused on using ontologies to enhance robot autonomy, we started searching on scientific databases utilizing related keywords. Concretely, we used the literature browser *Web of Science*¹⁰, previously known as *Web of Knowledge*, which is an online subscription-based scientific citation indexing service that provides a comprehensive citation search. It gives access to multiple databases that reference cross-disciplinary research, which allows for in-depth exploration of specialized sub-fields within an academic or scientific discipline.

Typing the keywords *ontology robot autonomy* and *ontology autonomous robotics* yields just 24 and 63 results, respectively. We considered that the number of papers was not enough for our purpose so we went on searching. In the interest of finding a larger list of results, we tried a more general set of keywords: *knowledge representation autonomous robotics*, which returned a list of 306 papers. Going through them, we realized the works were too general, indeed several of them were not even using knowledge representation approaches, therefore, we discarded this list too. The next step was to include the application domain scope in the search (see Section 3.3), hence, the set of keywords was: *knowledge representation industrial robotics* and *knowledge representation service robotics*, with 133 and 148 papers respectively.

The two lists found during the previous step, were combined in a single list with 281 articles in total. In the interest of identifying projects or initiatives that use ontologies to enhance robot autonomy, we have reduced the list of papers following a specific criteria:

- It is proposed to use knowledge representation techniques (ontologies) in robotics applications to enhance robot autonomy;
- the work is part of a project or a big initiative, not just a single article; and
- case studies where the knowledge base is used by a robot exist.

After applying this criterion, the list was reduced to 21 articles, which correspond to five different projects: KnowRob (Tenorth and Beetz 2009), IEEE-ORA (Schlenoff et al. 2012), ROSETTA (Stenmark and Malec 2013), CARESSES (Bruno, Chong, et al. 2017b), and RehabRobo-Onto (Dogmus, Erdem, and Patoglu 2019). This set of works, was enlarged by other four ones extracted from one of the surveys (Thosar et al. 2018) explained along the Section 1. In that work, Thosar et al., reviewed a list of nine works, which, as in our work, were chosen following a systematic search and inclusion criteria. We consider that only five of those nine works fit our purpose but one of them is KnowRob, already included before, thus, we only adopt four: ORO (Lemaignan, Ros, Mösenlechner, et al. 2010), RoboBrain (Saxena et al. 2014), OUR-K (Gi Hyun Lim, Il Hong Suh, and Hyowon Suh 2010), and OMRKF (Il Hong Suh et al. 2007).

It is worth mentioning that the project IEEE-ORA does not actually provide a complete framework which is available to be used, it consists of just an ontology. Indeed, the ontology developed in the framework of that project contains general concepts of the domain, so that it is not really useful in specific application scenarios. However, we understand that it is relevant enough to be considered in our work, since it aims at standardizing the representation of knowledge in the robotics domain. Therefore, we have tried to identify possible extensions of the original work which have been used under the umbrella of available frameworks. Following

¹⁰<https://www.webofknowledge.com/>

a similar approach as before (using the Web of Science’s browser), we searched for papers that cited the most cited article related to the project (Schlenoff et al. 2012). In this case, from the 34 initial works which cite it, only two followed the whole inclusion criteria presented in this section: OROSU (Gonçalves and P. M. Torres 2015) and PMK (Diab, Akbari, Ud Din, et al. 2019). In Section 4.3, where the projects are compared, we consider them as two individual frameworks. Nevertheless, along Section 4.2, where the approaches are explained, OROSU and PMK are grouped together.

4.1.2 Inclusion criteria

We have already discussed how we have selected the ten frameworks or projects which are considered to be object of the analysis performed in this work. However, among them, we want to focus only on the discussion (Section 4.2) and the comparison (Section 4.3) on the most influential approaches. Hence, this section provides a list of inclusion criteria to refine the list of surveyed projects. In the Section 4.4, we briefly introduce the excluded approaches and provide some justification for our decision. Projects or frameworks are only considered in the scope of this work if they satisfy all of the following criteria:

1. *Ontology scope*: The project uses an ontology that defines one of the terms that we have identified as particularly relevant for autonomous robotics (see Section 3.1);
2. *Reasoning scope*: It uses ontologies to support robots manifesting at least one of the cognitive capabilities that we have discussed earlier (see Section 3.2);
3. *Transparency*: It is transparent. Meaning that some material (e.g., websites, publications) is openly available that describes the overall goal of the project, what cognitive capabilities are considered, and how and what ontologies are used;
4. *Curation*: It is maintained. Meaning that recent developments or future plans are evident or at least possible; and
5. *Accessibility*: There exists – at least a prototypical – software that is accessible, and that demonstrates how ontologies are used to support a cognitive capability.

4.2 Discussion of frameworks/projects

In this section, we give an overview of the six frameworks/projects that have been subject of study in our review. For each of them, their underlying principles and foundations are discussed, as well as what application domain the system was designed for. We also describe how the frameworks evolved over time, and what impact they have had so far. The selection of the presented projects has been done based on the selection criteria presented in Section 4.1. To the best of our knowledge, we have included in this section all projects that satisfy these criteria.

4.2.1 KnowRob

KnowRob (Knowledge processing for Robots)¹¹¹² is an open source knowledge processing system that is designed for autonomous service robots. It was first introduced in 2009 (Tenorth and Beetz 2009). Tenorth and Beetz argue that autonomous robot control demands a KR&R system that addresses several aspects that are commonly not sufficiently considered in AI KR&R systems. One of these aspects is that robots need a more fine-grained action representation. This was discussed, in more detail, in another work where Tenorth and Beetz argue that service robots should be able to cope with (often) shallow and symbolic instructions, and to fill in the gaps to generate detailed, grounded, and (often) real-valued information needed for execution (Tenorth and Beetz 2017).

Recently, a second generation of the KnowRob system was introduced where the focus has shifted towards the integration of simulation and rendering techniques into a hybrid knowledge

¹¹<http://knowrob.org/>

¹²<https://github.com/knowrob/knowrob>

processing architecture (Beetz, Beßler, Haidu, et al. 2018; Haidu et al. 2018). The rational is to re-use components of the control program in virtual environments with physics and almost photorealistic rendering, and to acquire experiential knowledge from these sources. Experiential knowledge, called *narrative enabled episodic memory* in KnowRob, is used to draw conclusions about what action parameterization is likely to succeed in the real world (e.g., through learning methods) – this principle is inspired by the simulation theory of cognition (Hesslow 2012).

KnowRob has also been used in several research initiatives including the European projects RoboHow (Beetz, Beßler, Winkler, et al. 2016), RoboEarth (Waibel et al. 2011), SAPHARI (Beetz, Bartels, et al. 2015), and SHERPA (Marconi et al. 2012). RoboEarth, for example, is a pioneer work to consider exchanging knowledge between robots using the World Wide Web, OWL, and Linked Data principles. It was demonstrated, e.g., how such an infrastructure can be used to execute tasks that were not explicitly planned at design time. More recently, KnowRob has been used by the openEASE web knowledge service which is designed for the acquisition, storage, curation, visualization, and analysis of experiential robot knowledge (Beetz, Tenorth, and Winkler 2015). KnowRob plays further a central role in the ongoing DFG funded collaborative research center *Everyday Activity Science & Engineering (EASE)*¹³ which has the goal to uncover principles underlying everyday activities by first acquiring experiential knowledge with different modalities, and second building models that generalize over these modalities (Bateman et al. 2017).

The main programming language used in KnowRob is (SWI) Prolog which has its roots in first-order logic. SWI Prolog comes with a library to manage RDF triples, which is used by KnowRob to represent explicit knowledge in memory such as facts encoded in OWL ontologies. Initially, KnowRob was deriving its concept definitions from the Cyc ontology. However, only rather shallow symbolic representations were used that were tailored to provide useful information for task execution without enforcing consistency in the KB. In recent years, KnowRob has shifted towards the use of the DOLCE+DnS Ultralite (DUL) ontology and a more careful and principled modeling of foundational concepts for autonomous robotics. Another important principle underlying KnowRob is about how data that already exists in the robot control system can be made *knowledgeable* – that is how this data can be integrated into symbolic reasoning. KnowRob employs the notion of *virtual knowledge bases* that are computed on demand using control-level data such as data structures used by the perception and planning component of the robot control system. The computation is out carried by, so called, *computable properties* which are computation methods attached to symbolic relation defined in an ontology.

Without a doubt, KnowRob is one of the most influential KR&R systems for autonomous robots nowadays. This is evident through many research papers and projects that have been using and extending KnowRob since it was initially released. However, there are a couple of limitations worth mentioning here. First, KnowRob has been using only a very shallow symbolic representation following the principles of behavior-based robotics, and in particular the claim that the world itself is its own best model. But having a lot of information only encoded implicitly in data structures of the control program also creates some problems such as the computational cost of abstraction when symbolic inference is performed. Second, despite its long history, no representational standards were proposed by the KnowRob developers. Finally, even though it is one of the most used systems, and openly available, KnowRob has not yet succeeded in creating a large user community, but still has a huge potential to do so in the future.

4.2.2 ROSETTA

ROSETTA¹⁴ ¹⁵ stands for *RObot control for Skilled ExecuTion of Tasks in natural interaction with humans; based on Autonomy, cumulative knowledge and learning*. Its origin can be traced

¹³<https://ease-crc.org/>

¹⁴<http://www.fp7rosetta.org/>

¹⁵https://github.com/jacekmalec/Rosetta_ontology

to the European projects SIARAS (Haage et al. 2011) and RoSta¹⁶. During the development of those projects, a set of ontologies of robot skills was implemented with the goal to create an intelligent support system for reconfiguration and adaptation of robot-based manufacturing cells. Those ontologies evolved throughout the scope of other two European projects, ROSETTA and PRACE (Stenmark and Malec 2013). The former gave its name to the current ontology. The ROSETTA ontology has further been employed in the research projects SMERobotics (Perzylo, Rickert, et al. 2019) and SARAFun (G. Riva and E. Riva 2019). In these projects, the ontology has been used to enhance cognitive abilities of robots that are required to plan and execute assembly tasks. The core ontology has been reorganized after the initial release (Jacobsson, Malec, and Nilsson 2016), and new case studies on robot programming support (Elin Anna Topp and Malec 2018) and skill reusability in industrial scenarios (Elin A Topp et al. 2018) have been developed.

Originally, the ROSETTA ontology did not rely on any upper ontology, however, for more general terms regarding the robotics domain, it currently uses CORA (Schlenoff et al. 2012). Since CORA relies on SUMO (Niles and Pease 2001), one can assume that ROSETTA utilizes SUMO as its upper ontology. Even though SUMO is written in SUO-KIF¹⁷, ROSETTA is distributed in OWL. The *Knowledge Integration Framework* (Persson et al. 2010)¹⁸ connects all heterogeneous parts of the ROSETTA system: user GUI, simulation, external knowledge sources, task demonstration and the robot. It is the core of the whole system and its goal is to represent, store, adapt, and distribute knowledge across engineering platforms. The data, available in the AutomationML data exchange format¹⁹, is abstracted using RDF triples.

Along this section, we have provided some links to git repositories which are proof of the availability of both the ontology and some parts of the proposed software. However, there is no unique repository containing all different pieces of the complete system as a whole, which reduces the degree of accessibility. Indeed, some parts of the code (e.g. user GUI to program the robot) have not been found at all. On the other hand, the OWL file does not contain the definitions in natural language of the ontological terms, which would help for a better understanding of the formalization. Regarding the scalability of the system, crucial element of any industrial environment, it is not possible to say much due to the small size of the conducted experiments.

4.2.3 IEEE Standard Ontologies for Robotics and Automation

The 1872–2015 IEEE Standard Ontologies for Robotics and Automation²⁰ (Schlenoff et al. 2012), was developed in the context of the IEEE ORA WG. This standard defines an overall ontology that allows for the representation of, reasoning about, and communication of knowledge in the robotics and automation domain. This ontology includes key terms as well as their definitions, attributes, constraints, and relationships. Sub-parts of this standard include a linguistic framework, generic concepts (an upper ontology), a methodology to add new concepts, and sub-domain ontologies.

The purpose of the standard is to provide an overall ontology and an associated methodology for knowledge representation and reasoning in robotics and automation, together with the representation of concepts in an initial set of application domains. The standard provides a unified way of representing knowledge and provides a common set of terms and definitions, allowing for unambiguous knowledge transfer among any group of human, robots, and other artificial systems.

The proposed ontology is too general to be useful in complex applications, indeed creating a complete framework was out of the scope of the ORA working group initiative. Nevertheless, the ontology has still been used, for example, Jorge et al., present a scenario where a human ask for a pen and two robots are meant to cooperate in performing the task of collecting and

¹⁶Robot standards and reference architectures Project

¹⁷Variant of the Knowledge Interchange Format (KIF), a knowledge representation language.

¹⁸https://git.cs.lth.se/mathias/KIF_installation

¹⁹AutomationML is an on-going standard initiative that aims at unifying data representation and APIs used by engineering tools, <http://www.automationml.org/>

²⁰<https://github.com/srfiorini/IEEE1872-owl>

delivering it (Jorge et al. 2015). Specifically, one robot grasps the pen and poses it on a mobile platform from which the user is supposed to pick the pen up. Further, non-official extensions of the standard have emerged along the last years. In this section, we give a flavor of some of those extensions and how their use enhances robots' autonomy.

OROSU An Ontology for Robotic Orthopedic Surgery (OROSU)²¹ (Gonçalves and P. M. Torres 2015) was developed and then applied for hip resurfacing surgery (e.g., for trimming the femoral head). In this scope, the main goal of the research, related to ontologies, was to build a knowledge-based framework for this surgical scenario, along with a formal definition of components and actions to be performed during the surgery. The developed ontology was partially based on the 1872–2015 – IEEE Standard Ontologies for Robotics and Automation (Schlenoff et al. 2012). The work was developed under the HIPROB and ECHORD projects, funded by the Portuguese Science Foundation and the EU-FP7, respectively. The framework is among the first to integrate robotic ontologies in the domain of surgical robotics.

The application ontology OROSU, relies on SNOMED CT (Wang, Sable, and Spackman 2002), the CORA ontology (Schlenoff et al. 2012) and the KnowRob framework (Tenorth and Beetz 2013), which were adopted as the upper and reference ontologies. The formal language used to write the ontology was OWL.

It is partially accessible, but it would be desired to have more available material. Indeed, the ontology lacks of natural language definitions, what makes more difficult to understand the specific meaning of the terms. Moreover, the system seems not to have been used by other researchers apart from the developers.

PMK Perception and Manipulation Knowledge (PMK)²² (Diab, Akbari, Ud Din, et al. 2019) is a knowledge-based reasoning framework that includes some reasoning processes for autonomous robots to enhance Task and Motion Planning (TAMP) capabilities in the manipulation domain. A perception module can be integrated with the framework to capture a rich semantic description of the scene, knowledge about the physical behavior of the objects, and reasoning about the potential manipulation actions. The reasoning scope of PMK is divided into four parts: reasoning for perception, the reasoning for object features, the reasoning for a situation, and reasoning for planning.

PMK follows the preliminary structure of modeling of OUR-K, which divide the knowledge into three gradual layers called, metaontology, ontology-schema, and ontology instance (G. H. Lim, I. H. Suh, and H. Suh 2011). PMK enlarged the OUR-K structure by adding some concepts related to the manipulation domain. Moreover, aiming at being shared and reused, PMK ontology relies on other upper and reference/domain ontologies: SUMO (Niles and Pease 2001) and CORA (Schlenoff et al. 2012).

PMK is meant facilitate the process of manipulation by providing the required components for task and motion planning such as geometric reasoning, dynamic interactions, manipulation and action constraints. The use of PMK could become useful in the domain of robotic manipulation, however, since the system has been recently published, it has not yet been widely extended among other researchers.

PMK was implemented using ontology web language (OWL). Ontology instances can be asserted using information processed from low-level sensory data. The perception module used in the framework, consist of a 2D camera with the C++ library `ar_track_alvar` (http://wiki.ros.org/ar_track_alvar). Queries over the PMK are based on SWI-Prolog and its Semantic Web library, which serves for loading and accessing ontologies represented in the OWL using Prolog predicates.

²¹<https://github.com/pbsgoncalves/OROSU>

²²<https://github.com/MohammedDiab1/PMK>

4.2.4 ORO

ORO²³ is a project focused on the implementation of a common representation framework for autonomous robots with special emphasizes on human-robot interaction (Lemaignan, Ros, Mösenlechner, et al. 2010). The proposed framework was meant to enhance robot’s interaction with complex and human-inhabited environments, where robots are expected to exhibit advanced cognitive skills, such as: object recognition, natural language interaction, task planning with possible dynamic re-planning, ability to cooperate with other robots or humans, etc. The authors stated that, these functions, partially independent from each other, need to share common knowledge of the environment where the robot operates.

The ORO’s primary component is the *OpenRobots Common Sense Ontology*²⁴, which precisely provides an upper set of concepts upon which the robot can add and connect new statements of the world (ibid.). This ontology is built upon the OpenCyc upper ontology²⁵. The *minimalKB*²⁶ ontology is a lightweight version of ORO ontology, which shares the same objective and functionality as its predecessor.

ORO is mainly written in Java, while, *minimalKB*, its lightweight version, is based on Python. The underlying RDF triples storage is done using the open-source Jena framework, which is used together with the Pellet (Sirin et al. 2007) reasoner. ORO provides several wiki pages with detailed explanations of how to use it²⁷ and how to extend it²⁸. In addition, the OWL file contains some natural language definitions, which facilitates the understanding of the ontology.

4.2.5 CARESSES

CARESSES²⁹ is an international research project whose goal is to design the first robots that can assist older people and adapt to the culture of the individual they are taking care of (Bruno, Chong, et al. 2017b), (Bruno, Chong, et al. 2017a). The robots are expected to help the users in many ways including reminding them to take their medication, encouraging them to keep active, helping them keep in touch with family and friends. Each action should be performed with attention to the older person’s customs, cultural practices and individual preferences.

CARESSES’s principle aim is built upon four fundamental backbones: (a) transcultural robotic nursing, (b) cultural knowledge representation, (c) culturally sensitive planning and execution and (d) culture-aware human-robot interaction. Cultural knowledge, mainly represented using ontologies, enhances the robotic nursing integrating that knowledge into most of the robot processes (e.g. task planning, task execution, human-robot interaction, etc.). Nevertheless, other methodologies such as fuzzy logic and Bayesian networks, are also employed.

The knowledge based proposed within CARESSES consists of three layers: TBox, CBox and PBox. The former is the usual TBox found in any ontology, containing the statements which describe a conceptualization of the domain by defining different sets of individuals described in terms of their characteristics. The second and third layer stand for the usual ABox, which include TBox-compliant statements about individuals belonging to these sets. In this case, in the CBox the statements are related to cultural knowledge while in the PBox the knowledge is about one single person. As it is structured, the knowledge base aims at enhancing the robot adaptation to different cultural elderly people but also to the specific preferences of each individual.

Despite the short life of CARESSES, it has become a prominent application example of how ontologies could enhance the autonomy of robots. Nonetheless, the project still presents some drawbacks. For instance, not all of the implemented solutions are publicly available³⁰. On the other hand, the ontology’s quality is questionable. In the first place, the OWL file lacks of natural

²³<https://www.openrobots.org/wiki/oro-server/>

²⁴<https://www.openrobots.org/wiki/oro-ontology>

²⁵<http://www.opencyc.org/>

²⁶<https://github.com/severin-lemaignan/minimalkb>

²⁷<https://www.openrobots.org/wiki/oro-server-bindings>

²⁸<https://www.openrobots.org/wiki/oro-server-plugins>

²⁹<http://caressesrobot.org/en/>

³⁰<https://github.com/Suman7495/Robot-Navigation-for-Vision-Based-HAR>

language definitions, which hinders the understanding of it. In the second place, some of the entities seem not to be properly defined in a taxonomic view (e.g. event and object are subclasses of an entity named *topic*, defined as any topic the robot can talk about). Lastly, we would like to discuss the scalability of this approach, which opens some controversial issues. While being a reasonable way of inferring advantageous information of users' preferences, the proposed usage of cultural knowledge, at bigger scale, might result in robots with a strong bias. Depending on the context, users would probably prefer not to feel they are being prejudged (sometimes unfairly) by a robot.

4.3 Comparison of frameworks/projects

For the purpose of comparing the frameworks and projects described in Section 4.2, in this section we explore how each of the projects addresses the different aspects included in the classification of ontologies proposed along the Section 3. Concretely, the ontological, reasoning and application domain scopes of each of the selected works are thoroughly examined and contrasted.

4.3.1 Ontology scope

In this section, it is explored which of the terms discussed in the Section 3.1 are defined in each of the selected frameworks/projects. Note that we only consider that a term is defined when the natural language definition and/or the formalization are compliant with our domain. Indeed, if the exact term is not defined but the desired notion is captured by a similar term, for us, the concept is defined. For a summary of which terms are and are not included in each of the considered works see Table 1.

Object Most of the compared frameworks, define **Object** from an enduring perspective. In both, KnowRob 1.0 and ORO, the exact term is not defined but it is used (from Cyc ontology) the notion of **Spatial Thing**: *The collection of all things that have a spatial extent or location relative to some other Spatial Thing or in some embedding space.* OROSU uses SUMO's definition: *Corresponds roughly to the class of ordinary objects. Examples include normal physical objects, geographical regions, and locations of Processes, the complement of Objects in the Physical class.* ROSETTA does not concern about spatial regions and only focuses on **Physical Objects**: *Every automated work cell consists of physical objects. Some objects, devices, are active and have skills, while other, work pieces, are passive and are manipulated by the devices.* PMK proposes the use of **WSObjectClass**, which is split into: **Artifact**, **Artifact Components** and **Collections**. For example, a cup (artifact) is an object that has body and handle (artifact components) and could be served with saucer (collection). On the other hand, KnowRob 2.0, based on the DUL Ontology, considers not only physical entities: *Any physical, social, or mental object, or a substance. Objects are always participating in some event (at least their own life), and are spatially located.* In CARESSES, we cannot find any definition in natural language but object is defined as a subclass of the entity **Topic**, which is *any theme a robot can talk about.* We understand that this last definition is not aligned to what is needed in our domain. Hence, we claim that CARESSES does not define **Object**.

Environment map In both versions of KnowRob, it is possible to find the concept of **SemanticEnvironmentMap** as a sub-class of **Map**. However, there is not any natural language definition and it is not aligned to DUL (current upper ontology) yet. OROSU defines places and environments where the robot works (e.g. **CTRoom**, **EngineeringRoom**, **OperatingRoom**, which are sub-classes of **Room** and are connected to actions which are expected to take place in there. PMK formalizes the notion of **Workspace** which has three gradual sub-classes, **Region** (i.e., free and occupied regions), **Physical Environment** (topology of the environment entities), and **Semantic Environment** (semantic information of the workspace).

Affordance The concept of **Affordance** is not exactly defined in any of the works which are object of study, still, it is possible to find some definitions that partially capture the same knowledge. ORO defines the property `canBeManipulated` which *indicates if an object can be manipulated and that the agent knows a grasping point for the object. Thus, if the object can be manipulated, it is movable as well.* OROSU also describes a property, in this case `CanGrab`, which *indicates that a device can grab an object.* Even though in the later it is not mentioned the existence of any grasping point, it is implicit that if a robot can grasp an object it also knows where to do it from. KnowRob 2.0 has recently also introduced a notion of affordance that can be found in published ontologies, however, no scientific publication is available that discusses this concept. In KnowRob 2.0, affordances are defined as *individual qualities of objects that have an associated space region that locates the affordance relative to the object.* Affordances are further used in the definition of tasks to state that a particular affordance type is a participant in the task execution.

Action KnowRob 1.0 uses the definition provided by the Cyc ontology in which **Action** is defined as *an Event*. ORO, while also using the formalization from Cyc, provides a more concrete natural language definition of the term: *The collection of Events that are carried out by some "doer". Instances of Action include any event in which one or more agents effect some changes in the (tangible or intangible) state of the world, typically by an expenditure of effort or energy. Note that it is not required that any tangible object is moved, changed, produced, or destroyed for an action to occur; the effects of an action might be intangible.* In this definition emerges the relevance of the agent or the entity which actually performs the action. KnowRob 2.0 takes the term from the DUL ontology: *An event with at least one agent that is participant in it, and that executes a task that typically is defined in a plan, work flow, project, etc.* Again, the figure of an agent taking part in the execution of the action is noted. PMK defines the notion of `ActionClass` with three specifications: `Task`, `Sub-task` and `AtomicFunction`. Hence, picking would be an action class whose task can be reachability-test, and it can have a sub-task that provides a list of potential grasping poses. Similarly in OROSU the notion of sub-task has also been defined to form complex actions. Some of the analyzed frameworks do not provide any natural language definition for **Action**, even though they include the term in their ontology: CARESSES and OROSU.

Task Most of the studied ontologies agree on the fact that there exists a relationship between **Task** and **Action**. However, as stated in Section 3.1, there is not a common agreement on the exact relationship, and each framework/project employs a distinct formalization. In the ROSETTA ontology, `Task` is formalized as a disjoint with: `Operation`, `Skill`, `Physical object`, and `Property`, but no further information is provided. Another example of formalization is found in PMK, where `Task` is a sub-class of `Action`. ORO views it as *an action considered in the specific context of robotics.* KnowRob 2.0 uses DUL's definition: *an event type that classifies an action to be executed.*

Activity Only CARESSES covers the term of **Activity**, which is formalized as a sub-class of **Entity** and has some sub-classes: `Cooking`, `Reading`, `Sleeping`, etc. Natural language definitions are not provided.

Behavior None of the frameworks defines **Behavior**.

Function Just PMK includes a term related to **Function**, specifically, the notion of `AtomicFunction` a sub-class of `Action`. It refers to **Function** from a computational point of view (e.g. `Algorithm`). In the other frameworks, it is used the concept of `Algorithm` to define *computational tools* (OROSU) or provide some relations to express the intended (or primary)

function of objects (KnowRob 1.0). We consider that in these two last cases, it is not possible to state that the term is covered by the ontologies.

Plan KnowRob 2.0 takes DUL’s definition of **Plan**: *a description having an explicit goal, to be achieved by executing the plan*. Each plan defines a task that can be executed by following the plan, however, there may be different plans defining the same task. The execution of a plan is a situation that satisfies the plan, and that defines what particular objects will take what roles when the plan is executed. ORO does not provide a definition in natural language, but from the formalization, **Plan** is equivalent to a thing with a temporal extent, which is either a **Situation** or a **Time Interval**. The ORO definition seems to be a bit inconsistent since, surely, a time interval is not a plan in the common sense.

Method The unique framework which captures the notion of **Method** is KnowRob 2.0, where the definition provided by DUL is adopted: *A method is a description that defines or uses concepts in order to guide carrying out actions aimed at a solution with respect to a problem*. This notion is similar to the notion of **Plan**, but more general in that variances of following the same method could satisfy different plans.

Capability The ROSETTA ontology defines **Capability** as *a property of a skill* while in PMK the property **has capability** is defined as *a property of a robot*. In KnowRob 1.0, as part of the module SRDL³¹, capabilities are considered to exist when the robot has a component which enable them. In KnowRob 2.0, **Capability** is formalized as sub-class of **Quality** meaning that agents have individual qualities that may change over time, i.e., due to attrition, new software components being available, etc.

Skill The term of **Skill** is the core of the ROSETTA ontology where *a skill represents an action, that might be performed (by a device) in the context of a production process*. Similarly to the task-action dichotomy, skills are used to classify particular actions that occurred. However, it also implies that the robot or device has the capability to manifest the skill.

Hardware component The specific term of **Hardware component** is not tackled in any of the studied works. However, most of them address one or more concepts related to its notion. KnowRob includes the Semantic Robot Description Language (SRDL), which considers representations for robot hardware, among others. OROSU, from SUMO ontology, makes use of the term **Device** which *is an artifact whose purpose is to serve as an instrument in a specific subclass of a process*, where **Artifact** refers to any object that is the product of a making. An alike definition is found in ORO, where an **Artifact** is *a specialization of inanimate object, and each instance of artifact is an at least partially tangible thing which was intentionally created by an agent partially tangible (or a group of them working together) to serve some purpose or perform some function*. ROSETTA also includes the term of **Device**: *an active physical object which has some skills*. These notions are used to define **Sensors**. In PMK it is possible to find the terms **Actor Class** (e.g. robot components), **Sensor Class** (e.g. device components), and also the term **Artifact**, but none of those terms is defined using natural language. As we can see, definitions of hardware components are closely related to the processes and events in which they play a role.

Software component KnowRob includes the Semantic Robot Description Language (SRDL), which extends KnowRob with representations for robot software, among others. Terms related to

³¹The Semantic Robot Description Language (SRDL) extends KnowRob with representations for robot hardware, robot software and robot capabilities.

Term	KnowRob 1/2	ROSETTA	ORO	CARESSES	OROSU	PMK
Objects	Yes/Yes	Yes	Yes	No	Yes	Yes
Environment map	Yes/Yes	No	No	No	Yes	Yes
Affordance	No/Yes	No	Yes	No	Yes	No
Action	Yes/Yes	No	Yes	Yes	Yes	Yes
Task	No/Yes	Yes	Yes	No	No	Yes
Activity	No/No	No	No	Yes	No	No
Behavior	No/No	No	No	No	No	No
Function	No/No	No	No	No	No	Yes
Plan	No/Yes	No	Yes	No	No	No
Method	No/Yes	No	No	No	No	No
Capability	Yes/Yes	Yes	No	No	No	Yes
Skill	No/No	Yes	No	No	No	No
Hardware	Yes/Yes	Yes	Yes	No	Yes	Yes
Software	Yes/Yes	Yes	No	No	Yes	Yes
Interaction	No/No	No	No	No	No	No
Communication	Yes/No	No	No	No	No	No

Table 1 List of relevant terms for the autonomous robotics domain, and their coverage in the different chosen works. *Yes* and *No* state for when the term is or not covered by the ontology of the specific framework. Note that in the cases when the term is needed and taken from the upper ontology used within the framework, and/or when the knowledge is captured using a similar term, it is considered that the term is covered. If the upper ontology contains the term but it is not used, we consider that the term is not included.

the notion of **Computer-based Algorithm** can be found in both OROSU and PMK. ROSETTA ontology defines **Software** as *an abstract which has some skills*.

Interaction The term of **Interaction** is not defined in any of the frameworks we have surveyed. Nevertheless, in PMK it is possible to find the notion of **Interaction Parameter**, which is defined as a data property. Note that this does not mean that the notion of **Interaction** is captured.

Communication In the ROSETTA ontology, **Communication Property** *stands for the description of the communication parameters*. ROSETTA actually includes the term of **Communication**, but as a subclass of device, which hinders the understanding of it. Therefore, we cannot say that ROSETTA defines the notion of **Communication** which is needed for our domain. Apart from that, the only complete and coherent definition related to **Communication** is found in KnowRob 1.0, where the term of **Communicating** is taken from the Cyc ontology. It *is a specialization of purposeful action* and characterized by one or more information transfer event sub-events. Each instance of **Communicating** is an event in which the transfer of information between agents is a focal action; communicating is the main purpose and/or goal of the event.

4.3.2 Reasoning scope

In this section we compare the selected frameworks/projects with respect to their reasoning scope. Specifically, we analyze whether or not the different frameworks are used to support robots to perform the cognitive capabilities presented in Section 3.2.

Recognition and categorization Ros et al. present a use-case where ORO is used to support object detection by disambiguating incomplete information extracted from human-robot interaction (Ros et al. 2010). Specifically, the work proposes a scenario in which a human provides

vague instructions such as: *look at that object*, where the object can correspond to several entities in the environment. The ontology is used to represent facts about the user’s visual spectrum, and the description of objects so that the system is able to infer/recognize which is the most likely object.

Within the framework of CARESSES, Menicatti et al. (Menicatti, Bruno, and Sgorbissa 2017), introduce an approach for human activity recognition where cultural information (represented using ontologies) drives the learning improving the performance of the classification/categorization. Three human activities are considered: lying on the floor, sleeping on a futon and sleeping on a bed. Specially, lying on the floor and sleeping on a futon are extremely similar classes, thus, cultural knowledge (e.g. user is from Japan), is shown to improve the performance of the recognition algorithm.

KnowRob 2.0 is concerned with acquiring experiential knowledge from observations. One of the considered modalities is the virtual reality where force interactions can be monitored trivially. However, the intention and the task that the human executes might be unknown. KnowRob uses ontologies to represent tasks as patterns of force interactions, and state changes to be able to recognize high-level activities given force event and state observations (Beßler, Porzel, et al. 2019).

Decision making and choice In the context of CARESSES, the robot builds a model of a person, which is represented using the ontology. The robot adapts its behavior to the facts of the knowledge base, however, the adaptation to the user is really slow using the ontology. Therefore, it is also used a Bayesian Network for speeding up the adaptation to the person by propagating the effects of acquiring one specific information onto interconnected concepts (Bruno, Recchiuto, et al. 2019).

Diab et al. describe a robot system which adapts the execution of plans with the support of the PMK ontology (Diab, Akbari, Rosell, et al. 2017; Diab, Akbari, Ud Din, et al. 2019). Based on the beliefs about the workspace (reachability of objects, feasible actions to execute, etc.), the system makes decisions about the distribution of actions among different robotic arms, and also about action’s parameters, slightly modifying the original plan.

Perception and situation assessment A situation assessment reasoner, which generates relations between objects in the environment and agents’ capabilities, is presented in the context of ORO (Sisbot, Ros, and Alami 2011). Being fully integrated to a complete architecture, this reasoner sends the generated symbolic knowledge to a fact base which is built on the basis of an ontology and which is accessible to the entire system. The authors discussed how, based on spatial reasoning and perspective taking, the robot is able to reason from the human’s perspective, reaching a better understanding of the human-robot interaction.

Ros et al. present a use-case where ORO is used to disambiguate incomplete information extracted from human-robot interaction (Ros et al. 2010). Specifically, the work proposes a scenario in which a human provides vague instructions such as: ‘look at that object’, where the object can correspond to several entities in the working environment. Therefore, the robot is able to assess which is the situation of the environment using the ontology.

One example of a knowledge-based perception system is RoboSherlock (Beetz, Balint-Benczedi, et al. 2015). It uses IBM’s UIMA architecture to implement an assemble of perception experts. The problem is that there is no single pipeline that can handle all the different perception tasks, e.g. the pipeline for detecting transparent or shiny objects would be different from the one to detect ”regular” objects. RoboSherlock uses KnowRob to define what the different perception experts are, what input they expect, what output they generate, etc. This information, together with background knowledge KnowRob provides, is used for dynamic composition of perception pipelines.

Finally, in the PMK framework, a tagged-based vision module is proposed (Diab, Akbari, Ud Din, et al. 2019). In this module, the tags are used to detect the poses and IDs of world entities and

asserting them to the PMK to build the domain knowledge. Then, a reasoning mechanism is used to provide the reasoning predicates related to perception, object features, geometric reasoning, and situation assessment. Particularly, for situation assessment, an evaluation-based analysis is proposed which generates relations between the agent and the objects in the environment based on the perception outcomes, being, these relations used later to facilitate the planning process.

Prediction and monitoring One important aspect for robots that do manipulation tasks is to predict the effects of actions. KnowRob 1.0 introduced the notion of pre- and post-actors of actions (Tenorth and Beetz 2012). Pre-actors are the entities that must be known before the robot may execute the action, and post-actors describe what is expected when the action is successfully executed. For example, the task of *cracking an egg* would have a pre-actor of type egg that takes the role of being the *destroyed* entity in the action, while the yolk and the shell would be considered as *created* entities. Hence, the robot can predict what action it needs to execute in order to obtain some egg yolk, e.g. in case egg yolk is required in a recipe that the robot tries to cook. In KnowRob 2.0, the pre- and post-actor relations are replaced by corresponding role concepts describing roles that need to be taken by some entity when an action is performed.

Problem solving and planning KnowRob 2.0 uses ontologies for dynamic plan generation. This has been elaborated with respect to assembly tasks (Beßler, Pomarlan, and Beetz 2018). The rationale is that the goal state, a fully assembled product, is described in an ontology, and that the robot compares its belief state with the goal state in order to infer what steps are required, and what objects are missing to build the product from parts that are available. KnowRob 1.0, on the other hand, was using action definitions axiomatized by roles that are separated into input and output of the action, and, in addition, defined a partial ordering on steps of a task (Tenorth and Beetz 2012). This information was used to generate possible sequences of steps that would execute a task such as making pancakes.

There are some works in which the knowledge used to generate a plan is represented using an ontology. However, unlike the other works presented above, the ontology is not directly used to generate the plan, it only complements the planning. Therefore, we exclude those works from the Table 2, but it is worth mentioning them. For instance, KnowRob 1.0 has been used to represent motion constraints that were used by a constraint-based motion planner to generate appropriate motions for the task ahead, and the objects involved (Tenorth, Bartels, and Beetz 2014). Another example is PMK (Diab, Akbari, Ud Din, et al. 2019), which can serve as tool for any planner to reason about task and motion planning inference requirements, such as robot capabilities, action constraints, action feasibility, and manipulation behaviors.

Reasoning and belief maintenance As we are only considering frameworks that use ontologies, one can also expect that some form of reasoning is supported. Be it via a standard reasoner, or by some custom rules that can infer new facts from given ones. For example, PMK and KnowRob both use predicate logic rules to work with knowledge encoded in ontologies. However, belief maintenance is not covered by all considered systems.

In the context of ORO, Warnier et al. propose a novel algorithm for belief maintenance, which relies on the use of the ORO ontology to represent facts about the environment (Warnier et al. 2012). The robot builds an individual symbolic belief state for each agent taking part in the task.

Within the CARESSES project, Bruno et al. present an algorithm for belief maintenance of person-specific knowledge, which uses culture-specific knowledge to drive the search (Bruno, Recchiuto, et al. 2019).

Diab et al. propose to use the PMK ontology to generate semantic maps of the robot’s workspace enhancing its belief maintenance (Diab, Akbari, Ud Din, et al. 2019). By means of computer vision methods, the robot detects objects and its properties (e.g. poses) and, using the ontology, it stores a symbolic representation of the workspace. A reasoning process over those

symbolic beliefs allows, for instance, to make assumptions about abstract spatial relations (e.g. cup *on* the table).

KnowRob 2.0 also maintains a belief state (Beßler, Pomarlan, and Beetz 2018), however, there is no detailed documentation about how the belief state is maintained, and how the system would cope with contradictory information. However, a notion of **SemanticMap** exists in the KnowRob ontology, and it has been used in KnowRob 1.0 to build environment representations that include spatial information and encyclopedic information about objects in the map (Tenorth, Kunze, et al. 2010).

Execution and action In order to enhance robot autonomy when executing actions, ROSETTA proposes a system which translates high-level task-oriented language (ontology-based) into either the robot native code, or calls at the level of a common API like, e.g., ROS, or both. This system is capable to handle complex, sensor-based actions, likewise the usual movement primitives (Stenmark, Malec, and Stolt 2015).

In the context of CARESSES, Sgorbissa et al. discuss how guidelines describing culturally competent assistive behaviours can be encoded in a robot to effectively tune its actions, gestures and words (Sgorbissa et al. 2018). In the same context, an online constraint-based Planner is used together with the cultural knowledge base in order to adapt the execution of the robot’s actions (Khaliq et al. 2018). When launched, the planner requests operators and actions from the Cultural Knowledge Base. During execution it listens for new goals, updates on the execution status of actions, and messages about the state of the environment and people in the environment.

Gonçalves et al. discussed how the use of the OROSU ontology is beneficial to track the execution of actions of robotics systems in medical (surgical) scenarios (Gonçalves and P. M. Torres 2015). In this work, the main purpose is to adapt the robot pose to possible unexpected motions while performing drilling tasks during a surgery. The robot pose adaptation is performed following the approach presented in (P. Torres, Gonçalves, and Martins 2015). The overall process is modeled with the OROSU ontology, which controls the robot’s actions and sub-actions and allows the user to follow the sequence of those actions.

The *Cognitive Robot Abstract Machine* (CRAM) is a plan executive that is grounding abstract plan descriptions such that they become executable by robots (Beetz, Mösenlechner, and Tenorth 2010). To find suitable task instantiations, CRAM uses KnowRob to, e.g. query for objects in the belief state, where they are located, how they can be operated, etc. The queries to the knowledge base are explicit steps in the plan instantiation procedure of CRAM. Nowadays, CRAM has switched to the second generation of KnowRob. KnowRob 1.0 has further been used to ontologically describe motion constraints that are used by a constrained-based motion controller to generate motions that execute a specific task (Tenorth, Bartels, and Beetz 2014). KnowRob 1.0 has also been used to transform vague task descriptions in natural language from the Internet to an ontological representation using WordNet to disambiguate word senses (Tenorth, Nyga, and Beetz 2010). Finally, the ontological representation are mapped into the robot’s plan language such that the robot can execute the task.

Interaction and communication Lemaignan et al. present a simple natural language processor which employs ORO to allow robots to dialog with humans (Lemaignan, Ros, Alami, et al. 2011). The robot parses English sentences and, by means of the knowledge base, infers the sense of the sentences and answers the human (both in English and with RDF statements). Again using the ORO ontology, in a scenario where a robot disambiguates the information provided by the user (Ros et al. 2010), the ontology triggers the robot-human interaction (e.g. asking to the user for further information).

In the scope of CARESSES, Bruno et al. describe two scenarios where human-robot speech-based interaction is adaptable by means of cultural knowledge-based assumptions (Bruno, Menicatti, et al. 2018; Bruno, Recchiuto, et al. 2019). The system stores knowledge about the

cultural information of the users, which is used by the robot’s finite state machine to control the interaction.

KnowRob 2.0 was used and extended in a research project that was concerned with mixed human-robot rescue tasks (Yazdani et al. 2018). The scenario is that a team of different robots have to locate an avalanche victim in hilly terrain where, first, a flying robot scans the area, and then, after the victim was found, the robot communicates the particular location, and an image captured by its camera to the human operator, and to the other robots. KnowRob’s ontology is used to represent the communication acts. However, the communication was not natural but was using a custom protocol.

Remembering, reflection, and learning Different approaches combining ontologies and robot learning are proposed in the context of ROSETTA. First, ontologies are used to support the kinesthetic teaching so that the learned primitives are semantically represented as skills (Stenmark, Haage, et al. 2018). Second, Topp et al. discuss how the representation of already learned robot’s skills enhances the transfer of knowledge from one robot to others (Elin A Topp et al. 2018).

KnowRob 2.0 introduces the notion of *narrative-enabled episodic memories* (NEEMs) (Beetz, Beßler, Haidu, et al. 2018). Each time a robot performs a task, a detailed story about the activity is stored. The story includes a narrative represented in an ontology that describes what events occurred, when they occurred, and what objects play what roles for the events. The narrative is coupled with control-level data such that learning mechanisms can correlate parts of the narrative to the control level data that was monitored during execution. Earlier, in the context of KnowRob 1.0, the knowledge web service openEASE was introduced (Beetz, Tenorth, and Winkler 2015). openEASE is used as a central storage for experiential knowledge, and it has been adopted for KnowRob 2.0 as a storage for NEEMs.

4.3.3 Application domain scope

In order to finish the comparison of the different works, in this section, we discuss in which domain they have been applied. Recall that, following the classification proposed in Section 3.3, two main domains are considered: industrial and service robotics. Along this section, we will specialize those upper-level-domains into the more specific sub-domains where the frameworks were used.

In principle, it is noticed that most of the frameworks were conceived to be used in service scenarios. Indeed, the only framework which is intended to be used in industrial scenarios is ROSETTA, whose case studies solve industrial problems such as: human-friendly robot programming, safe human-robot interaction, etc. Moving to the frameworks focused on service robotics, ORO proposes case studies where the robot is meant to perform everyday activities which usually take place in houses or similar environments such as human activity recognition, human-robot speech interaction, etc. Closely related to it, we find the case of KnowRob, which is used in the framework of household scenarios, but also in scenarios where the robot is expected to perform some professional service tasks (e.g. cooking, in-store logistic processes, etc.). PMK presents cases studies where the principle aim is to enhance robot manipulation, which is a general purpose robot ability which could potentially be used in a wide range of scenarios. Nevertheless, PMK has not been used nor thought to be used in industrial scenarios, thus, we can consider it behind the umbrella of the service robotics domain. CARESSES is entirely developed towards the assistance of elder people by means of autonomous robots with cultural-related knowledge. Finally, OROSU is mainly applied to the medical domain, particularly, the surgical robotics sub-domain. In the Table 3, we provide a summary of the previous discussion.

4.4 Excluded frameworks/projects

In this section, we give a flavor of some projects which, while having been considered for our analysis, were discarded following our inclusion criteria (see Section 4.1.2). Table 4 shows which of the criteria are met or not by these projects.

Cognitive Capability	KnowRob	ROSETTA	ORO	CARESSES	OROSU	PMK
Recognition and categorization	Beßler et al. 2019	-	Ros er al. 2010	Menicatti et al. 2017	-	-
Decision making and choice	-	-	-	Bruno et al. 2019	-	Diab et al. 2017; 2019
Perception and situation assessment	Beetz et al. 2015	-	Ros et al. 2010, Sisbot et al. 2011	-	-	Diab et al. 2019
Prediction and monitoring	Tenorth et al. 2012	-	-	-	-	-
Problem solving and planning	Beßler et al. 2018, Tenorth et al. 2012	-	-	-	-	-
Reasoning and belief maintenance	Beßler et al. 2018, Tenorth et al. 2010	-	Warnier et al. 2012	Bruno et al. 2019	-	Diab et al. 2019
Execution and action	Beetz et al. 2010, Tenorth et al. 2014; 2010	Stenmark et al. 2015	-	Sgorbissa et al. 2018	Gonçalves et al. 2015	-
Interaction and communication	Yazdani et al. 2018	-	Ros et al. 2010, Lemaignan et al. 2011	Bruno et al. 2018; 2019	-	-
Remembering, reflection and learning	Beßler et al. 2018, Tenorth et al. 2015	Stenmark et al. 2018, Topp et al. 2018	-	-	-	-

Table 2 List of cognitive capabilities for the autonomous robotics domain and their coverage in the different chosen frameworks/ontologies. It is possible to find the reference to the articles in which the different reasoning capabilities are addressed using the ontologies.

Framework	KnowRob	ROSETTA	ORO	CARESSES	OROSU	PMK
Application Domain	Service	Industrial	Service	Service	Service	Service

Table 3 Application domain for each chosen work.

Inclusion Criterion	RoboBrain	OMRKF	OUR-K	REHABROBO
1 <i>Ontology scope</i>	no	yes	yes	-
2 <i>Reasoning scope</i>	no	yes	yes	no
3 <i>Transparency</i>	yes	no	no	yes
4 <i>Curation</i>	no	no	no	yes
5 <i>Accessibility</i>	yes	no	no	no

Table 4 Inclusion criteria applied to some excluded projects. *yes* indicates that the criterion is met, *no* that it is not met, and - is written when it is unknown.

RoboBrain RoboBrain³² (Saxena et al. 2014) releases on the Web a huge robot knowledge base where robots can share their experiences and learn. RoboBrain is a large-scale computational system that learns from publicly available Internet resources (e.g., Wikipedia, WordNet,

³²<http://robobrain.me/index.html>

ImageNet, Freebase, OpenCyc), computer simulations, and real-life robot trials. The knowledge is represented in a graph with thousand of nodes and edges. This project presents a relevant effort towards the use of knowledge in robotics and it is continuously maintained. Nonetheless, the developed framework does not use, nor it seems it is planned to start doing it, a concrete formalization of the knowledge based on ontologies, which makes it eligible to be excluded from our analysis.

OMRKF Ontology-based Multi-layered Robot Knowledge Framework (OMRKF) (Il Hong Suh et al. 2007), aims at enhancing robots intelligence by integrating low-level data with high-level knowledge into the same framework. OMRKF has four levels of knowledge, each of them split into three levels: (a) model (object feature, object and space); (b) context (spatial, temporal and high-level); (c) perception (numerical descriptor, visual feature and visual concept); and (d) activity (behavior, task and service). In this context, knowledge representation helps robots to execute sequenced behaviors by just specifying the high level service and also how the robot can recognize objects even when the knowledge is not complete. This system means a good effort towards cognitive autonomous robots. However, it has not been possible to find any available material nor enough documentation, hence, the project is excluded.

OUR-K Ontology-based Unified Robot Knowledge (OUR-K) is a framework that integrates low-level data with high-level knowledge for robot intelligence in service robotics scenarios (Gi Hyun Lim, Il Hong Suh, and Hyowon Suh 2010). It seems to be an extension of OMRKF, because they share some similarities and authors. The framework consist of three parts: (a) knowledge description, (b) knowledge association, and (c) application. The former takes care of the representation of knowledge (low-level data and high-level knowledge) by using five classes of entities: features, objects, spaces, contexts and actions. Knowledge association specifies the relationships between different descriptions allowing several inference methods (logics, bayesian inference, heuristics). Finally, the descriptions and their relationships are used in several applications: navigation, action selection, object recognition, context awareness, planning and object manipulation. Lim (Gi Hyun Lim 2019), presents how OUR-K is used in some case studies where the knowledge the robot holds is incomplete. This framework seems to be beneficial to our domain and it could be included in our analysis if it were not for the absence of available resources. None the ontology nor the framework have been found in public repositories. Therefore, it is not included in our review.

REHABROBO REHABROBO-QUERY (Dogmus, Erdem, and Patoglu 2019) is a web based software which allows robot designers to add, modify and consult information about their rehabilitation robots. This information is stored using a the formal ontology REHABROBO-ONTO (Dogmus, Erdem, and Patoglu 2015). The whole system is available on the cloud, utilizing Amazon EC2. REHABROBO-QUERY is accessible through a web page³³ hosted on Amazon servers. The whole framework turns into a useful tool when your intention is to store/consult descriptions of rehabilitation robots (e.g. robot parts, capabilities, etc.). Nevertheless, it remains unclear how this system could be utilized to equip robots with autonomy, indeed, the system has not been used to solve any of the reasoning problems proposed in Section 3.2. Moreover, there does not exist a simple way of accessing to the OWL file of the ontology and the Amazon server they use is not fully free. It was decided then, not to include this work in our analysis.

5 Discussion

In this section, we summarize the most relevant findings on the review of projects/frameworks which use ontologies to support robot autonomy (see Section 4.3). Recall that following the

³³<http://ec2-54-228-52-230.eu-west-1.compute.amazonaws.com/rehabrobo/>

literature search and the inclusion criteria presented in Section 4.1, from the initial ten found projects, we focused on six (see Section 4.2). In Section 4.4, we present a brief explanation of the excluded projects and the reasons why they were not analyzed in depth.

In general, we consider that the six selected projects provided enough information to allow to reuse and reproduce the results. It is worth mentioning though, the fact that KnowRob was utterly well documented, including code, ontologies, as well as wiki pages explaining how to install and to use the different tools. It is also true that KnowRob is one of the very first frameworks which made use of ontologies for autonomous robots, meanwhile, other projects are much more recent. One important fact is that only one of the projects seems not to be maintained anymore, ORO. Probably, the reason is that it was a project partially developed by the same researchers who started developing KnowRob. However, ORO is probably the second best documented project. Therefore, even though it is not currently maintained, it would still be possible to reuse some parts of it.

The first classification criteria proposed in our work concerns about the ontological scope of the projects, namely which terms are covered by the projects' ontologies (see Section 3.1). Every project that fulfilled our selection criteria defines their own domain ontologies. These are, in all cases, derived from well known foundational ontologies (i.e., SUMO, DUL, and Cyc) that are represented in different ontology languages (i.e., SUO-KIF, OWL, CycL). Looking at the Table 1, we can see how most of the projects include the terms **Action** and **Task** to capture the notion of robots acting and causing changes in their environment. In the same table, we observe that **Behavior**, **Function** and **Method** are only rarely covered. This is rather surprising given the fact that they are extremely related to applications where agents execute actions. We think that this is the case due to their polysemous nature, therefore, more work is needed in order to come up with a standard definition for them. Some other terms such as **Plan**, **Capability**, **Hardware** and **Software** are defined by at least half of the surveyed projects, which indicates that they are relevant for the domain but still it is necessary to continue working on them. We also discovered that the terms **Interaction** and **Communication** are not defined in any of the projects, even though some projects propose scenarios where robots interact and communicate with other agents (e.g. humans). Hence, no reasoning is done in this regard. Surprisingly, some other terms with strong connotation in robotics such as **Activity** and **Skill** are also only rarely considered in formal models. In general, we have identified the existence of inconsistencies and different points of view from one framework to another. Therefore, we believe that it is necessary to work towards an agreement on the definition for the relevant terms in our domain.

The second classification criteria that we propose is related to the reasoning scope of the ontologies of each of the projects (see Section 3.2). In other words, which cognitive capabilities of autonomous robots have already been supported by those ontologies. Our review has shown that a wide range of cognitive capabilities are already covered, at least in a prototypical way, by knowledge-based approaches that employ ontologies (see Table 2). From the nine cognitive capabilities considered in this work, two are commonly tackled within the studied frameworks: reasoning and belief maintenance, and execution and action. Reasoning and belief maintenance are well supported by many ontology formalisms as standard reasoners exist that can perform this tasks automatically. On the other hand, robots are essentially developed to automatize the execution of actions; hence, it is understandable to also find several works tackling this task. Surprisingly, important cognitive capabilities such as: decision making, problem solving and planning, are only tackled in a few projects. Probably, the existence of other widely used formalisms to do planning (e.g. PDDL) is the principal reason which explains this fact. Finally, in the table we can also see how ontologies have been used for learning tasks. We think that considering the current trend of non-explainable machine learning approaches, the formal and logical nature of ontologies could be beneficial in robot learning applications.

Lastly, we also studied the application domain of the selected projects. Most of the projects were conceived with the purpose of being used in service robotics applications. Indeed, just

ROSETTA was specifically designed for industrial robotics applications. Nevertheless, the rest of the frameworks could be adapted to be useful in industrial settings too.

Considering the work presented in this article, it is possible to state that ontologies have proved to be valuable for the robotics domain in order to support robot autonomy. It is true, however, that the great effort done by all the frameworks discussed in this paper should be continued and extended with new applications. Furthermore, it is still pending to promote the reuse of existing ontologies, which seeks for homogeneity and interchangeability among different frameworks. This will only be possible if researchers share and properly document their contributions. Indeed, the spread of standard conceptualizations would also help to achieve the previous purpose. In this regard, we have summarized the content of our article on a web page³⁴ where users can access the major findings of our work. Our aim is to continuously maintain and improve this page to provide researchers and ontology users easy access to related work. Specifically, the page allows to search/select by projects and by each of the three scopes proposed along our work.

As a final remark, we consider that if the research community keeps working on the current direction and is able to solve the issues related to reusability, ontologies could become essential in topics such as explainable robots, ethics-aware robots, and robust and safe human-robot interaction as they are a tool to relax the programming effort required to deliver robots that are adaptive.

6 Conclusion

In this manuscript, we have systematically searched for projects in the intersection of ontologies and autonomous robotics, and have compared them according to what terms are defined in their ontologies, how these terms are used to support different cognitive capabilities of a robot, and in which application domain they are used. The study has shown that a wide range of cognitive capabilities is already covered by existing frameworks, however, often only in a prototypical form. A few projects were sufficiently documented to be fully assessed, and each of them builds on-top of a well known foundational ontology that is used to conceptualize parts of the autonomous robotics domain. We believe that such a formal conceptualization of the robotics domain is an important factor to relax the programming effort required to match the expectations on research and industry to deliver robots that can autonomously perform a large variety of tasks in a large variety of environments.

Acknowledgement

This work was partially funded by the following agencies: Deutsche Forschungsgemeinschaft (DFG) through the Collaborative Research Center *EASE* (1320); Regional Catalan Agency ACCIÓ through the RIS3CAT 2016 project SIMBIOTS (COMRDI16-1-0017); Spanish State Research Agency through the María de Maeztu Seal of Excellence to IRI (MDM-2016-0656); Spanish Government through the project DPI2016-80077-R and grant 2017; Foundation for Science and Technology (FCT) through project UID/EMS/50022/2019; and Interreg Programme through the Project 0043-EUROAGE-4-E.

References

- A. Yanco, Holly and Jill Drury (Jan. 2004). “Classifying human-robot interaction: An updated taxonomy”. In: vol. 3, pp. 2841–2846.
- Arp, Robert, Barry Smith, and Andrew D Spear (2015). *Building Ontologies with Basic Formal Ontology*. MIT Press.

³⁴<https://ease-crc.org/ontology-survey-2019>

- Balakirsky, Stephen et al. (2017). “Towards a robot task ontology standard”. In: *ASME 2017 12th International Manufacturing Science and Engineering Conference collocated with the JSME/ASME 2017 6th International Conference on Materials and Processing*. American Society of Mechanical Engineers, V003T04A049–V003T04A049.
- Bateman, John et al. (2017). “Heterogeneous Ontologies and Hybrid Reasoning for Service Robotics: The EASE Framework”. In: *Third Iberian Robotics Conference*. ROBOT '17. Sevilla, Spain.
- Beer, Jenay M, Arthur D Fisk, and Wendy A Rogers (2014). “Toward a framework for levels of robot autonomy in human-robot interaction”. In: *Journal of human-robot interaction* 3.2, pp. 74–99.
- Beetz, Michael, Ferenc Balint-Benczedi, et al. (May 2015). “RoboSherlock: Unstructured information processing for robot perception”. In: pp. 1549–1556.
- Beetz, Michael, Georg Bartels, et al. (2015). “Robotic Agents Capable of Natural and Safe Physical Interaction with Human Co-workers”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Hamburg, Germany.
- Beetz, Michael, Daniel Beßler, Andrei Haidu, et al. (2018). “KnowRob 2.0 – A 2nd Generation Knowledge Processing Framework for Cognition-enabled Robotic Agents”. In: *International Conference on Robotics and Automation (ICRA)*.
- Beetz, Michael, Daniel Beßler, Jan Winkler, et al. (2016). “Open Robotics Research Using Web-based Knowledge Services”. In: *International Conference on Robotics and Automation (ICRA)*. Stockholm, Sweden.
- Beetz, Michael, Dominik Jain, et al. (2012). “Cognition-Enabled Autonomous Robot Control for the Realization of Home Chore Task Intelligence”. In: *Proceedings of the IEEE* 100.8, pp. 2454–2471.
- Beetz, Michael, Lorenz Mösenlechner, and Moritz Tenorth (2010). “CRAM - A Cognitive Robot Abstract Machine for everyday manipulation in human environments”. In: *IROS*. IEEE, pp. 1012–1017.
- Beetz, Michael, Moritz Tenorth, and Jan Winkler (2015). “Open-EASE – A Knowledge Processing Service for Robots and Robotics/AI Researchers”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Finalist for the Best Cognitive Robotics Paper Award. Seattle, Washington, USA.
- Beßler, Daniel, Mihai Pomarlan, and Michael Beetz (2018). “OWL-enabled Assembly Planning for Robotic Agents”. In: *Proceedings of the 2018 International Conference on Autonomous Agents*. AAMAS '18. Finalist for the Best Robotics Paper Award. Stockholm, Sweden.
- Beßler, Daniel, Robert Porzel, et al. (2019). “Foundational Models for Manipulation Activity Parsing”. In: *Proceedings of AR and VR Conference: Changing Realities in a Dynamic World*.
- Borgo, Stefano, Amedeo Cesta, et al. (July 2019). “Knowledge-based Adaptive Agents for Manufacturing Domains”. In: *Eng. with Comput.* 35.3, pp. 755–779.
- Borgo, Stefano, Maarten Franssen, et al. (2014). “Technical artifacts: An integrated perspective”. In: *Applied Ontology* 9.3-4, pp. 217–235.
- Borst, P., H. Akkermans, and J. Top (1997). “Engineering Ontologies”. In: *International Journal of Human-Computer Studies* 46, pp. 365–406.
- Brooks, Rodney (1991). “Intelligence Without Representation”. In: *Artificial Intelligence* 47, pp. 139–159.
- Bruno, Barbara, Nak Young Chong, et al. (2017a). “Paving the way for culturally competent robots: A position paper”. In: *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, pp. 553–560.
- (2017b). “The CARESSES EU-Japan project: making assistive robots culturally competent”. In: *Italian Forum of Ambient Assisted Living*. Springer, pp. 151–169.

- Bruno, Barbara, Roberto Menicatti, et al. (2018). “Culturally-Competent Human-Robot Verbal Interaction”. In: *2018 15th International Conference on Ubiquitous Robots (UR)*. IEEE, pp. 388–395.
- Bruno, Barbara, Carmine Tommaso Recchiuto, et al. (2019). “Knowledge Representation for Culturally Competent Personal Robots: Requirements, Design Principles, Implementation, and Assessment”. In: *International Journal of Social Robotics*, pp. 1–24.
- Buehler, Jennifer Elisabeth and Maurice Pagnucco (2014). “A Framework for Task Planning in Heterogeneous Multi Robot Systems Based on Robot Capabilities”. In: *AAAI*. AAAI Press, pp. 2527–2533.
- Chandrasekaran, B., John R. Josephson, and V. Richard Benjamins (1998). “Ontology of Tasks and Methods”. In: *11th Workshop on Knowledge Acquisition, Modeling and Management (KAW’98)*. Banff, Canada.
- Chella, Antonio et al. (2002). “Modeling Ontologies for Robotic Environments”. In: *Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering. SEKE ’02*. Ischia, Italy: ACM, pp. 77–80.
- Compton, Michael et al. (2012). “The SSN ontology of the W3C semantic sensor network incubator group”. In: *Web semantics: science, services and agents on the World Wide Web* 17, pp. 25–32.
- Davidson, Donald (2001). *Essays on actions and events: Philosophical essays*. Vol. 1. Oxford University Press on Demand.
- Diab, Mohammed, Aliakbar Akbari, Jan Rosell, et al. (2017). “An ontology framework for physics-based manipulation planning”. In: *Iberian Robotics conference*. Springer.
- Diab, Mohammed, Aliakbar Akbari, Muhayy Ud Din, et al. (2019). “PMK-A Knowledge Processing Framework for Autonomous Robotics Perception and Manipulation”. In: *Sensors* 19.5.
- Dix, Alan (2009). *Human-computer interaction*. Springer.
- Dogmus, Zeynep, Esra Erdem, and Volkan Patoglu (2015). “RehabRobo-Onto: Design, development and maintenance of a rehabilitation robotics ontology on the cloud”. In: *Robotics and Computer-Integrated Manufacturing* 33. Special Issue on Knowledge Driven Robotics and Manufacturing, pp. 100–109.
- (2019). “RehabRobo-Query: Answering natural language queries about rehabilitation robotics ontology on the cloud”. In: *Semantic Web* 10.3, pp. 605–629.
- Fazel-Zarandi, Maryam and Mark S. Fox (July 2013). “Inferring and Validating Skills and Competencies over Time”. In: *Appl. Ontol.* 8.3, pp. 131–177.
- Gangemi, Aldo, Stefano Borgo, et al. (2004). *Task Taxonomies for Knowledge Content D07*. Tech. rep. Metokis Project.
- Gangemi, Aldo and Peter Mika (2003). “Understanding the semantic web through descriptions and situations”. In: *OTM Confederated International Conferences” On the Move to Meaningful Internet Systems*. Springer, pp. 689–706.
- Gibson, James J. (1979). *The Ecological Approach to Visual Perception*. Houghton Mifflin.
- Gil, Yolanda (2005). “Description Logics and Planning”. In: *AI Magazine* 26.2, pp. 73–84.
- Gómez-Pérez, Asunción, Mariano Fernández-López, and Oscar Corcho (2004). *Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. 1st. Advanced Information and Knowledge Processing. Springer.
- Gonçalves, Paulo JS and Pedro MB Torres (2015). “Knowledge representation applied to robotic orthopedic surgery”. In: *Robotics and Computer-Integrated Manufacturing* 33, pp. 90–99.
- Gruber, Thomas (1993). “A translation approach to portable ontologies”. In: *Knowledge Acquisition* 5.2, pp. 199–220.
- Grüninger, Michael (2004). “Ontology of the process specification language”. In: *Handbook on ontologies*. Springer, pp. 575–592.

- Guarino, Nicola (June 1998). “Formal Ontology in Information Systems”. In: *Proceedings of FOIS’98*. Trento, Italy: IOS Press, Amsterdam, pp. 3–15.
- Guarino, Nicola and Pierdaniele Giaretta (1995). “Ontologies and Knowledge Bases: Towards a Terminological Clarification”. In: *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing (KBKS’95)*. Ed. by Mars N. University of Twente, Enschede, The Netherlands: IOS Press, Amsterdam, The Netherlands, pp. 25–32.
- Guarino, Nicola, Daniel Oberle, and Steffen Staab (2009). “What is an Ontology?” In: *Handbook on ontologies*. Springer, pp. 1–17.
- Haage, Mathias et al. (2011). “Declarative-knowledge-based reconfiguration of automation systems using a blackboard architecture”. In: *Eleventh Scandinavian Conference on Artificial Intelligence*. Vol. 227. IOS Press, pp. 163–172.
- Haidu, Andrei et al. (2018). “KNOWROB-SIM – Game Engine-enabled Knowledge Processing for Cognition-enabled Robot Control”. In: *International Conference on Intelligent Robots and Systems (IROS)*. IEEE. Madrid, Spain.
- Hesslow, Germund (2012). “The current status of the simulation theory of cognition”. In: *Brain Research* 1428, pp. 71–79.
- Robot and Robotics Devices – Vocabulary* (Mar. 2012). Standard. International Organization for Standardization.
- Jacobsson, Ludwig, Jacek Malec, and Klas Nilsson (2016). “Modularization of skill ontologies for industrial robots”. In: *Proceedings of ISR 2016: 47th International Symposium on Robotics*. VDE, pp. 1–6.
- Jorge, Vitor AM et al. (2015). “Exploring the IEEE ontology for robotics and automation for heterogeneous agent interaction”. In: *Robotics and Computer-Integrated Manufacturing* 33, pp. 12–20.
- Khaliq, Ali Abdul et al. (2018). “Culturally aware planning and execution of robot actions”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 326–332.
- Krüger, Volker et al. (2007). “The Meaning of Action: a review on action recognition and mapping”. English. In: *Advanced Robotics* 21.13, pp. 1473–1501.
- Kunze, Lars, Tobias Roehm, and Michael Beetz (May 2011). “Towards Semantic Robot Description Languages”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China, pp. 5589–5595.
- Langley, Pat, John E Laird, and Seth Rogers (2009). “Cognitive architectures: Research issues and challenges”. In: *Cognitive Systems Research* 10.2, pp. 141–160.
- Lemaignan, Séverin, Raquel Ros, Rachid Alami, et al. (2011). “What are you talking about? grounding dialogue in a perspective-aware robotic architecture”. In: *2011 RO-MAN*. IEEE, pp. 107–112.
- Lemaignan, Séverin, Raquel Ros, Lorenz Mösenlechner, et al. (2010). “ORO, a knowledge management platform for cognitive architectures in robotics”. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3548–3553.
- Lenat, Douglas B. and Ramanathan V. Guha (1990). *Building large knowledge-based system: representation and inference in the Cyc project*. New York: Addison-Wesley.
- Lim, G. H., I. H. Suh, and H. Suh (May 2011). “Ontology-Based Unified Robot Knowledge for Service Robots in Indoor Environments”. In: *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 41.3, pp. 492–509.
- Lim, Gi Hyun (2019). “Shared representations of actions for alternative suggestion with incomplete information”. In: *Robotics and Autonomous Systems*.
- Lim, Gi Hyun, Il Hong Suh, and Hyowon Suh (2010). “Ontology-based unified robot knowledge for service robots in indoor environments”. In: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 41.3, pp. 492–509.

- Marconi, L. et al. (Nov. 2012). “The SHERPA project: smart collaboration between humans and ground-aerial robots for improving rescuing activities in alpine environments”. In: *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. College Station, Texas, USA.
- Masolo, C. et al. (Dec. 2003). *WonderWeb Deliverable D18: Ontology Library*. Technical Report. Laboratory for Applied Ontology -ISTC-CNR.
- Masolo, Claudio and Stefano Borgo (2005). “Qualities in formal ontology”. In: *Foundational Aspects of Ontologies (FOnt 2005) Workshop at KI 2005*, pp. 2–16.
- McDermott, Drew et al. (1998). *PDDL—The Planning Domain Definition Language*.
- Menicatti, Roberto, Barbara Bruno, and Antonio Sgorbissa (2017). “Modelling the influence of cultural information on vision-based human home activity recognition”. In: *2017 14th international conference on ubiquitous robots and ambient intelligence (URAI)*. IEEE, pp. 32–38.
- Mizoguchi, Riichiro, Yoshinobu Kitamura, and Stefano Borgo (2016). “A unifying definition for artifact and biological functions”. In: *Applied Ontology* 11.2, pp. 129–154.
- Moralez, Larry A. (Jan. 2016). “Affordance Ontology: Towards a Unified Description of Affordances as Events”. In: *Res Cogitans* 07, pp. 35–45.
- Neuhaus, Fabian, Pierre Grenon, and Barry Smith (2004). “A formal theory of substances, qualities, and universals”. In: *Formal Ontology in Information Systems: Proceedings of the Third International Conference (FOIS-2004)*. IOS Press.
- Niles, Ian and Adam Pease (2001). “Towards a standard upper ontology”. In: *Proceedings of the international conference on Formal Ontology in Information Systems-Volume*. ACM.
- Norman, Donald A. (2002). *The Design of Everyday Things*. New York, NY, USA: Basic Books, Inc.
- Ortmann, Jens and Werner Kuhn (2010). “Affordances As Qualities”. In: *Proceedings of the 2010 Conference on Formal Ontology in Information Systems: Proceedings of the Sixth International Conference (FOIS 2010)*. Amsterdam, The Netherlands, The Netherlands: IOS Press, pp. 117–130.
- Oxford-University, Press (2019). “Compact Oxford English Dictionary of Current English”. URL: <http://www.askoxford.com> (visited on 2019).
- Papadimitriou, Christos H (2003). *Computational complexity*. John Wiley and Sons Ltd.
- Paulius, David and Yu Sun (2018). “A Survey of Knowledge Representation and Retrieval for Learning in Service Robotics”. In: *CoRR* abs/1807.02192.
- Persson, Jacob et al. (2010). “A knowledge integration framework for robotics”. In: *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*. VDE, pp. 1–8.
- Perzylo, Alexander, Julian Grothoff, et al. (Aug. 2019). “Capability-based semantic interoperability of manufacturing resources: A BaSys 4.0 perspective”. In: *Proceedings of the IFAC Conference on Manufacturing Modeling, Management, and Control (MIM)*. Berlin, Germany.
- Perzylo, Alexander, Markus Rickert, et al. (2019). “SMERobotics: Smart Robots for Flexible Manufacturing”. In: *IEEE Robotics and Automation Magazine*.
- Riva, Giuseppe and Eleonora Riva (2019). “SARAFun: Interactive Robots Meet Manufacturing Industry”. In: *Cyberpsychology, Behavior, and Social Networking* 22.4, pp. 295–296.
- Ros, Raquel et al. (2010). “Which one? grounding the referent based on efficient human-robot interaction”. In: *19th International Symposium in Robot and Human Interactive Communication*. IEEE, pp. 570–575.
- Rusu, Radu Bogdan et al. (Nov. 2009). “Model-based and Learned Semantic Object Labeling in 3D Point Cloud Maps of Kitchen Environments”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. St. Louis, MO, USA.
- Salustri, Filippo A. (2000). “Ontological Commitments in Knowledge-Based Design Software: A Progress Report”. In: *Knowledge Intensive Computer Aided Design: IFIP TC5 WG5.2*

- Third Workshop on Knowledge Intensive CAD December 1–4, 1998, Tokyo, Japan.* Ed. by Susan Finger, Tetsuo Tomiyama, and Martti Mäntylä. Boston, MA: Springer US, pp. 41–72.
- Saxena, Ashutosh et al. (2014). “Robobrain: Large-scale knowledge engine for robots”. In: *arXiv preprint arXiv:1412.0691*.
- Schlenoff, Craig et al. (2012). “An IEEE standard ontology for robotics and automation”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.
- Sgorbissa, Antonio et al. (2018). “Encoding guidelines for a culturally competent robot for elderly care”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 1988–1995.
- Sirin, Evren et al. (2007). “Pellet: A practical owl-dl reasoner”. In: *Web Semantics: science, services and agents on the World Wide Web* 5.2, pp. 51–53.
- Sisbot, E Akin, Raquel Ros, and Rachid Alami (2011). “Situation assessment for human-robot interactive object manipulation”. In: *2011 RO-MAN*. IEEE, pp. 15–20.
- Stenmark, Maj, Mathias Haage, et al. (2018). “Supporting semantic capture during kinesthetic teaching of collaborative industrial robots”. In: *International Journal of Semantic Computing* 12.01, pp. 167–186.
- Stenmark, Maj and Jacek Malec (2013). “Knowledge-Based Industrial Robotics.” In: *SCAI*, pp. 265–274.
- Stenmark, Maj, Jacek Malec, and Andreas Stolt (2015). “From high-level task descriptions to executable robot code”. In: *Intelligent Systems’ 2014*. Springer, pp. 189–202.
- Studer, R., V.R. Benjamins, and D. Fensel (1998). “Knowledge Engineering: Principles and Methods”. In: *IEEE Transactions on Data and Knowledge Engineering* 25.1-2, pp. 161–197.
- Suh, Il Hong et al. (2007). “Ontology-based multi-layered robot knowledge framework (OMRKF) for robot intelligence”. In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 429–436.
- Tenorth, Moritz, Georg Bartels, and Michael Beetz (2014). “Knowledge-based Specification of Robot Motions”. In: *Proc. of the European Conference on Artificial Intelligence (ECAI)*.
- Tenorth, Moritz and Michael Beetz (2009). “KnowRob – knowledge processing for autonomous personal robots”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 4261–4266.
- (2012). “A unified representation for reasoning about robot actions, processes, and their effects on objects”. In: *IROS*. IEEE, pp. 1351–1358.
 - (2013). “KnowRob: A knowledge processing infrastructure for cognition-enabled robots”. In: *The International Journal of Robotics Research*.
 - (2017). “Representations for robot knowledge in the KnowRob framework”. In: *Artificial Intelligence*.
- Tenorth, Moritz, Lars Kunze, et al. (Dec. 2010). “KNOWROB-MAP – Knowledge-Linked Semantic Object Maps”. In: *10th IEEE-RAS International Conference on Humanoid Robots*. Nashville, TN, USA, pp. 430–435.
- Tenorth, Moritz, Daniel Nyga, and Michael Beetz (May 2010). “Understanding and Executing Instructions for Everyday Manipulation Tasks from the World Wide Web”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Anchorage, AK, USA, pp. 1486–1491.
- Thosar, Madhura et al. (2018). “A Review of Knowledge Bases for Service Robots in Household Environments”. In: *6th International Workshop on Artificial Intelligence and Cognition*.
- Tiddi, Ilaria et al. (2017). “An Ontology-based Approach to Improve the Accessibility of ROS-based Robotic Systems”. In: *Proceedings of the Knowledge Capture Conference*. K-CAP 2017. Austin, TX, USA: ACM, 13:1–13:8.
- Topp, Elin Anna and Jacek Malec (2018). “A Knowledge Based Approach to User Support for Robot Programming”. In: *AI for Multimodal Human Robot Interaction Workshop within the Federated AI Meeting 2018 in Stockholm*, pp. 31–34.

- Topp, Elin A et al. (2018). “Ontology-Based Knowledge Representation for Increased Skill Reusability in Industrial Robots”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 5672–5678.
- Torres, Pedro, Paulo J. S. Gonçalves, and Jorge Martins (2015). “Robotic motion compensation for bone movement, using ultrasound images”. In: *Industrial Robot: An International Journal* 42.5, pp. 466–474.
- Turvey, Michael T. (1992). “Affordances and prospective control: An outline of the ontology”. In: *Ecological psychology* 4.3, pp. 173–187.
- Uschold, M. and M. Gruninger (1996). “Ontologies: Principles, Methods and Applications”. In: *Knowledge Engineering Review* 11.2, pp. 93–155.
- Uschold, Mike et al. (1998). “The enterprise ontology”. In: *The knowledge engineering review* 13.1, pp. 31–89.
- Vernon, David (2014). *Artificial cognitive systems: A primer*. MIT Press.
- Vernon, David, Giorgio Metta, and Giulio Sandini (2007). “A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents”. In: *IEEE transactions on evolutionary computation* 11.2, pp. 151–180.
- Waibel, Markus et al. (2011). “Roboearth—a world wide web for robots”. In: *IEEE Robotics and Automation Magazine (RAM), Special Issue Towards a WWW for Robots*.
- Wang, Amy Y, Jeremiah H Sable, and Kent A Spackman (2002). “The SNOMED clinical terms development process: refinement and analysis of content.” In: *Proceedings of the AMIA Symposium*. American Medical Informatics Association, p. 845.
- Warnier, Mathieu et al. (2012). “When the robot puts itself in your shoes. managing and exploiting human and robot beliefs”. In: *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, pp. 948–954.
- Yanco, Holly A and Jill L Drury (2002). “A taxonomy for human-robot interaction”. In: *Proceedings of the AAAI Fall Symposium on Human-Robot Interaction*, pp. 111–119.
- Yazdani, Fereshhta et al. (2018). “Cognition-enabled Framework for Mixed Human-Robot Rescue Team”. In: *International Conference on Intelligent Robots and Systems (IROS)*. IEEE. Madrid, Spain.