

Ontology for Autonomous Robotics

Joanna Isabelle Olszewska, Joel Luís Carbonera, Alberto Olivares-Alarcos, Sandro Rama Fiorini, Maki Habib, Signe Redfield, Alaa Khamis, Julita Bermejo-Alonso, Paulo Goncalves, S. Veera Ragavan, Abdelghani Chibani, Edson Prestes, Edison Pignaton de Freitas, Marcos Barreto, Ricardo Sanz, Bruce Spencer, and Howard Li

Abstract— Creating a standard for knowledge representation and reasoning in autonomous robotics is an urgent task if we consider recent advances in robotics as well as predictions about the insertion of robots in human daily life. Indeed, this will impact the way information is exchanged between multiple robots or between robots and humans and how they can all understand it without ambiguity. Indeed, Human Robot Interaction (HRI) represents the interaction of at least two cognition models (Human and Robot). Such interaction informs task composition, task assignment, communication, cooperation and coordination in a dynamic environment, requiring a flexible representation. Hence, this paper presents the IEEE RAS Autonomous Robotics (AuR) Study Group, which is a spin-off of the IEEE Ontologies for Robotics and Automation (ORA) Working Group, and and its ongoing work to develop the first IEEE-RAS ontology standard for autonomous robotics. In particular, this paper reports on the current version of the ontology for autonomous robotics as well as on its first implementation successfully validated for a human-robot interaction scenario, demonstrating the developed ontology’s strengths which include semantic interoperability and capability to relate ontologies from different fields for knowledge sharing and interactions.

I. INTRODUCTION

In early 2015, the IEEE-RAS Ontologies for Robotics and Automation Working Group (IEEE ORA WG) published the IEEE 1872-2015 standard, the first-ever standard from the IEEE Robotics and Automation Society. This standard defines a set of ontologies related to robotics and automation (R&A), including the core ontology for robotics and automation (CORA), which specifies the main and most general concepts and axioms in the R&A domain. Due to this achievement, in December 2015, IEEE ORA WG was the recipient of the Emerging Technology Award, a prize given annually by the IEEE Standards Association¹.

IEEE ORA WG was divided into different subgroups, each in charge of studying a specific R&A subdomain, such as industrial robotics, service robotics, and autonomous robotics (AuR) [1]. In 2016, AuR received approval from the IEEE RAS Standing Committee to establish a study group conducting standards activities in AuR [2], which will impact on all R&A domains.

Indeed, the main benefit of a domain ontology is to set standard definitions of shared concepts identified in the requirement phase and to define appropriate relations between the concepts and their properties [3]. Hence, a standard ontology in AuR aims to provide the underlying semantics of

the vocabulary used, e.g. in developments or communications of heterogeneous autonomous systems.

This paper presents in-progress work carried out by the AuR subgroup to extend CORA ontology, in order to represent specific concepts and axioms commonly used in AuR, based on studies of various R&A subdomains such as flying robots, mobile robots, field robots, marine systems, etc., and to identify the basic components, including hardware and software, necessary to endow robots with autonomy.

The remainder of this paper is structured as follows. Background information about the development of our ontology for AuR is presented in Section II, while our ontology itself is described in Section III. Its validation in case of human-robot interactions is reported in Section IV and conclusions are given in Section V.

II. ONTOLOGY BACKGROUND

An ontology is more than a classification of concepts, i.e. a taxonomy. Indeed, an ontology is a formal and explicit specification of a shared conceptualization [4]. Such conceptualization specified by an ontology includes the concepts related to the types of entities that are supposed to exist in a given domain, according to a community of people. Hence, an ontology captures a common understanding about a given domain. Due to this, ontologies can be used for promoting the *semantic interoperability* among stakeholders, because *sharing a common ontology is equivalent to sharing a common view of the world*.

Since the specification of the conceptualization captured by an ontology should be formal and explicit, the meaning of every concept has to be rigorously specified, in order both humans and machines can use them without ambiguity [5]. Sharing a conceptualization is a prerequisite for communication. Human-Robot Interaction shall necessarily be based on such common conceptualization. Thus, an ontology can serve as the common basis for communication between humans and machines, and this is one of the main purposes of our ontology under development.

Our ontology for AuR we propose in Section III has been built following *METHONTOLOGY* [6] methodology, which is a mature ontology development methodology, independent of any specific application. We have also decided to adopt a *middle-out* [7] approach for specifying concepts. According to this strategy, we start by specifying the *most common concepts*, branching out to the *most general* and then to the *most specific* ones. This allows us to focus on the

¹For more information see <http://standards.ieee.org/develop/awards/etech/>

papers can be thought as instances of ROA. In essence, the conceptualization described below is akin to the meta-models of concept representation languages such as Unified Modelling Language (UML) (see Fig. 1).

A. Vocabulary Development

At first, We present the definitions we set for the fundamental notions like *Behavior*, *Function*, *Goal*, and *Task* to build our ontology on.

1) *Behavior*: Behavior_d relates to the actions of the robot. More specifically, it can be defined as:

(a) a specific action of the robot, regardless of whether it was specified, desired, or intended by the designer (“The robot’s looping behavior_a is preventing it from reaching the waypoint”)

(b) a generic term for the observed or desired actions of the robot (“The robot’s behavior_b wasn’t what the user wanted”)

(c) some property of the actions of the robot (“The behavior_c of this avoidance algorithm includes avoiding narrow but passable hallways.”)

(d) a self-contained set of actions relating to a specific task (that robot has an “avoid” behavior_d that’s very effective)

It is worth to note that none of these definitions differentiate between behaviors that are pre-programmed vs. learned or reactive vs. deliberative.

2) *Function*: Functions_a define goals_b at the behavioral_a level. More specifically, it can be defined as:

(a) the thing a given component is supposed to do, defined at any level (“This robot’s function_a is to clean floors.”)

(b) a procedure or routine that returns a value - this procedure or routine can be constrained by the complexity of the software implementing it, can be a behavior_d, something that implements a behavior_d, or something that is implemented by behaviors_{d,a}; it can be defined at any level of complexity

(c) a mathematical relationship between variables (“Force is a function_c of mass and acceleration; $F=g(m,a)$ ”)

3) *Goal*: The goal_a is what the robot has to do using behaviors_d to accomplish it. More specifically, it can be defined as:

(a) the externally defined desired end (or continuing) state of the system. Note that the goal_a is the thing that the operator or other external entity wants the robot to do. If the task_b has been decomposed into subtasks_b, the goal_a is the desired end (or continuing) state of each subtask_b as defined by the task_b.

(b) a subsidiary desire within the context of a larger problem - goal_b is shorthand for what a given behavior is trying to accomplish in the abstract (e.g. as the operator or designer would define it). If the overall goal_a is to survey the region, the robot or its survey behavior_d may be said to have a goal_b of following a specific list of waypoints; the follow-list-of-waypoints behavior_d may be said to have a goal_b of reaching a specific waypoint.

(c) a metric against which a given behavior_b is evaluated in the context of a specific task_a (applies when goal_a or goal_b involves quantitative elements, e.g. “The goal_c is to collect 10 blocks.”)

4) *Task*: While the goal_a defines the robot’s job from the user’s perspective, the task_a define’s the robot’s job from the robot’s perspective. More specifically, it can be defined as:

(a) a restatement of the goal from the robot’s perspective. If the goal_a is the expression of what the operator wants done, the task_a is how the robot interprets it. Subtasks can be defined to whatever depth is necessary. Tasks and subtasks are accomplished via behaviors_a and actions.

(b) a lower or higher level behavior_d. Within a given discussion, it is common for task_b to be used as a generic term to enable individuals to differentiate between the behavior_d under discussion and other lower or higher level behaviors_d. The resulting allocated tasks_a are synonymous with the behaviors required to accomplish them, and those behaviors_d are often referred to as tasks_b. This is particularly relevant during task_a decomposition/task_a allocation discussions, where the decomposition process results in subtasks that, from the perspective of the original task_a, are synonymous with the behaviors_d used to accomplish them.

It is worth noting there are as many ways of breaking up a goal_a into tasks_a and subtasks as there are robots and designers. There is also considerable confusion regarding the exact definition of *task* and *behavior* as they are commonly used. *Task* is often used to describe both the goal_a and the behavior_d, and the words used to define a specific task_a are often the same words used to describe behaviors_d. There are some attempts to separate a generic behavior_d programmed into the robot (“pick up a cup”) from a specific behavior_d instantiated by the robot in a particular situation (“pick up that cup”), but often no distinction is made.

B. Robot Architecture and Document

Based on these vocabulary terms (Section III.A), we can proceed to specify them as concepts in ROA (Fig. 1).

The main concept in ROA is *Robot Architecture*, which is a subclass of *Proposition* in SUMO. An instance of *Robot Architecture* represents a specific architecture a robot might implement. The subsumption relation with *Proposition* highlights the informational nature of architecture; architectures do not exist as physical entities, being only informational entities. The relation between a robot and an instance of *Robot Architecture* is represented by the SUMO relation *conforms* which states the robot, as a physical object, somehow follows the information contained by the architecture.

In SUMO, *Propositions* may be materialized by *Content Bearing Objects*, through the relation *containsInformation*. These are the actual physical artifacts that encode propositions, and include documents, computer files, formulas, strings, etc. The concept *Robot Architecture Document* represent a specific class of *Content Bearing Objects* that contain information about robot architecture proposition. As any other *Content Bearing Object*, robot architecture documents can be any kind of artifact, from XML files to paper documents, including formulas in OWL.

A robot architecture is composed by elements, which are also *Propositions*. We introduce the binary relation *associationRA* to represent the association between different

elements in a robot architecture. We also introduce two basic types of elements, namely *Layers* and *Modules*. A module is an element that represents an individual aspect of the system. Modules can be considered as black boxes, with inputs and outputs. Layers are elements that include other elements with a similar role. Layers are organized as stacks, commonly representing different levels of functionality.

C. Function, Behavior, and Structure

The architecture of a robot is frequently defined at the design phase. Thus, elements constituting the design are also relevant to the definition of robot architecture. The ontological nature of the design has been discussed intensively in the literature, with a particular focus on the *Function-Behavior-Structure* (FBS) ontology [10]. Indeed, FBS ontology defines the three main elements that constitute the process of designing, namely, function, behavior_b and structure, and also defines the causal relations between these elements. In this paper, we define our versions of these elements in the context of autonomous robots and SUMO/CORA.

In our ontology, *Behavior_a* of a robot is defined as an instance of *Robot Behavior*, which is any process where the agent (i.e. SUMO *agent* relation) is a robot. It is important to understand what such an instance is. As with any process in SUMO, an instance of *Robot Behavior* represents the occurrence of a single event. For example, if a robot picks up a box twice, then such movement implies the existence of two subsequent instances of *Robot Behavior*, with two well-defined boundaries in time. It is important to note that if a robot participates in a behavior_a process, it does not necessarily imply that such behavior_a was a design choice.

A *Robot Function_a* is a proposition describing the designed purpose of the robot as an artifact. An example of instance of *Robot Function_a* is “pick the box”. *Robot Functions_a* are part of *Robot Architecture*. If a robot conforms to a *Robot Function_a*, then there is a subclass of *Robot Behavior* that corresponds to this *Robot Function_a* and it is the *purpose* of this robot (in SUMO terms) to be an agent in instances of that *Robot Behavior_a* class.

A *Robot Structure* is a physical object that is part of a robot and gathers relevant parts together. The notion of relevant parts is entirely subjective and context-dependent. Physical structure and electric structure are different possible types of such structures inhering in a robot. Moreover, the structure is extensional, since it changes if any of its elements change. As the structure is usually considered as a collection of parts *and* spatial relations [11], we restrict robot structure to contain only parts of the robot that forms the required structure, to avoid second-order constructs (i.e. relations of relations).

D. Robot Motion

CORA introduced the notion of *Robot Motion*, defined as “any process of movement where the agent is a robot and the patient is one of its (robot) parts” [12]. The concept of *Robot Motion* is then subsumed by *Process*. Any process that has a robot as an agent is a *Robot Behavior*, therefore *Robot Motion* is subsumed by *Robot Behavior* (Fig. 2).

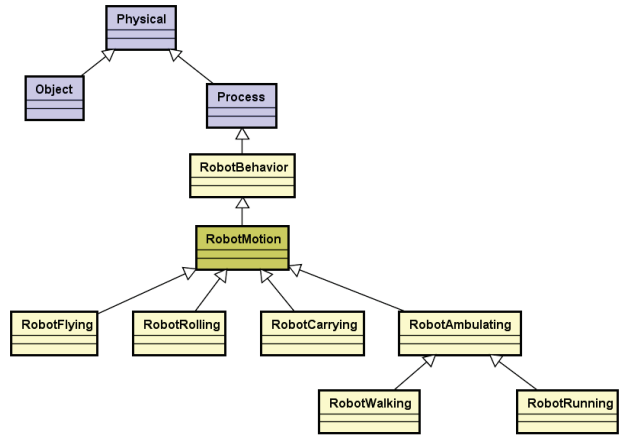


Fig. 2: Robot motion taxonomy.

In this paper, we provide some specific types of *Robot Motion*. For example, *Robot Ambulating* is any *Robot Motion* accomplished by means of Robot’s legs for the purpose of moving from one point to another. Types of ambulating robots are *Robot Walking* which moves in a way that at least one foot is always in contact with the ground; and *Robot Running* which moves in a way that, with each step, neither foot is in contact with the ground for a period of time.

Moreover, *Robot Carrying* is defined as a kind of *Transfer* (*Transfer* in SUMO) from one point to another by means of a Robot. *Robot Rolling* is a type of motion that combines *Rotating* and *Translocation* (in SUMO) of an *Object* with respect to a surface (which can move as well). If ideal conditions exist, both are in contact with each other without sliding, e.g. moving or being moved on wheels. Finally, *Robot Flying* is a movement such as a robot is able to move through the air using an artifact or set of artifact, e.g. wings.

E. Robot and Device Taxonomy

A bidirectional causal relationship exists between, on one hand, a robot’s architecture, function, and behaviors and, on the other hand, the kind of robot and devices attached to that robot. Thus, if a robot is categorized and the devices which conform it are known, then it is possible to infer some information about the architecture, function or behaviors of that robot. It is why CORA includes a robot taxonomy that depends on the autonomy of the robot, but that should be extended. Hence, in this paper, we propose a new taxonomy based on the environment where the robot is supposed to work, leading to different kinds (subclasses) of robots such as *stationary robot*, *ground robot*, *aerial robot* and *underwater robot*. CORA also defines four different types of robot parts, namely, *processing*, *actuating*, *sensing* and *communicating* parts. Regarding these concepts, there must be four types of devices that are expected to be parts of a robot. *Processing Device* and *Measuring Device* are already defined in CORA and SUMO, respectively. So, we introduce in ROA *actuating* and *communicating* devices as the necessary devices for robot actuation and communication.

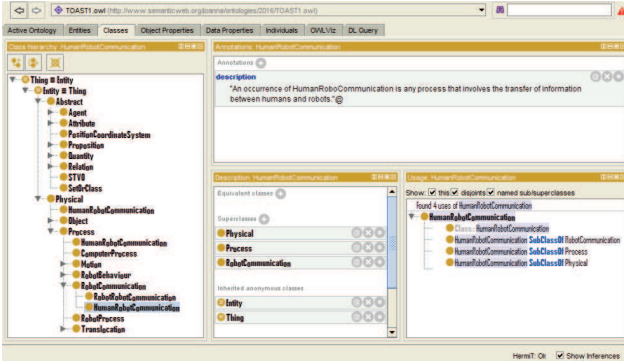


Fig. 3: *HumanRobotCommunication* ontological concept implementation.

F. Implementation of the Ontology for Autonomous Robotics

Our Robot Architecture Ontology (ROA), which covers vocabulary developments, functions, behavior, structure, motion and device taxonomy, has been implemented in the OWL language. For example, the concept of *HumanRobotCommunication*, which is any process that involves the transfer of information between humans and robots, has been encoded as an instance of the *RobotCommunication* class, itself a subclass of the robot physical processes (see Fig. 3.) Moreover, as displayed in Fig. 3, the implementation of the autonomous robot ontology includes also the *patio-Temporal Visual Ontology* (STVO) [13] used to represent the knowledge about the visual information acquired from robot’s sensors, such as cameras, in an ontological form. Indeed, it could contribute, on one hand, towards direct communication using natural language commands between a human user with the autonomous robot to assist the robot in its evolution within its environment, and, on the other hand, towards autonomous reasoning of the robot about the environment where it operates [5].

To meet these requirements, the visual data captured by robot’s digital cameras has been processed in three main computational phases. Firstly, it consists in automatically processing the live stream images to extract visual information such as objects of interests or other parts of the observed scene, using appropriate computer-vision techniques [14]. Secondly, this numerical information is mapped into semantic and abstract concepts as defined in STVO [13]. Thirdly, reasoning is performed based on these concepts and their relations [15]. In particular, qualitative spatial relations (QSR) have been proven to be useful to assist with reasoning about physical environment for scene understanding [15]. Furthermore, semantic directional spatial relations based on the clock model [16] such as “isAt2Oclock” are computationally efficient and complement well grounded spatial information for human-robot interactions [17].

IV. HUMAN-ROBOT INTERACTION CASE STUDY

Human-robot interactions have an important role due to the spread of robots into human daily life. Indeed, through

effective interactions, robots could be able to perform many tasks_a in human society. These tasks_a may include handling various house duties, providing medical care for elderly people, assisting people with motor or cognitive disabilities, educational entertainment (edutainment), personal assistance, hospital logistic aids, collaborative search-and-rescue during disaster situations, giving directions at information points in public places, museum tour guiding, etc. These applications need to develop social robots that can work with humans as partners if not peers in the form of bystanders or team mates [18]. Such robots should have a high level of autonomy enabling the robot to survive in different situations.

Human-robot interactions can be more challenging if they occur between multiple robots and multiple humans. Hence, the proposed scenario deals with two cooperative communities, namely, human community and robot community [19]. This scenario includes human-human, robot-robot, and human-robot interactions. In each community, there are a number of cooperative entities which act independently, but if there is a need, they negotiate with each other to form a cooperative group to handle common or individual tasks_a. In this scenario, there are two human agents, H1 and H2, and three robots, R1, R2, and R3, respectively. R1 is an unmanned ground vehicle equipped with a gripper; R2 is equipped with a surveillance camera; and R3 is equipped with both camera and gripper, as illustrated in Fig. 4 where the numbers indicate the order of the communications and messages occurrence. The scenario is described as follows:

- 1) H1 broadcasts a message containing a task_a operation code to all the robots asking one of them to go to point ‘X’ to perform a specific task_a without any preference.
- 2) The robots start to negotiate with each other to determine which robot will perform the requested task_a based on the proximity to the task_a to be performed. R1, R2, and R3 share their distances from Point ‘X’ and compare them to its own. The closest one to the point is chosen to do the task_a (R3 in this case).
- 3) H2 sends an order to R3 to go to Point ‘Y’ to perform a task_a that requires specific resources only available in R3, i.e. a task_a requiring both a camera and a gripper.
- 4) R3 has now two allocated tasks_a. The first one is optional but allocated, while the second one is mandatory as R3 is the only robot which can perform it. So, R3 sends H2 a message informing him that it has been allocated the first task_a and R3 asks H2 to grant permission from the first task_a’s owner H1 to switch the first task_a to another robot.
- 5) Thus, H2 asks H1 for granting a permission to him in order to pass it to R3 to cancel R3’s first job.
- 6) H1 replies by either granting H2 the permission or not. If the permission is given, H2 needs also to receive the operation code, in order to ask R3 to cancel its job.
- 7) H2 asks R3 to go to Point ‘Y’ and passes to it the operation code given by H1.
- 8) R3 tries to match the operation code given by H2 with the original one. If there is a match, R3 will cancel its

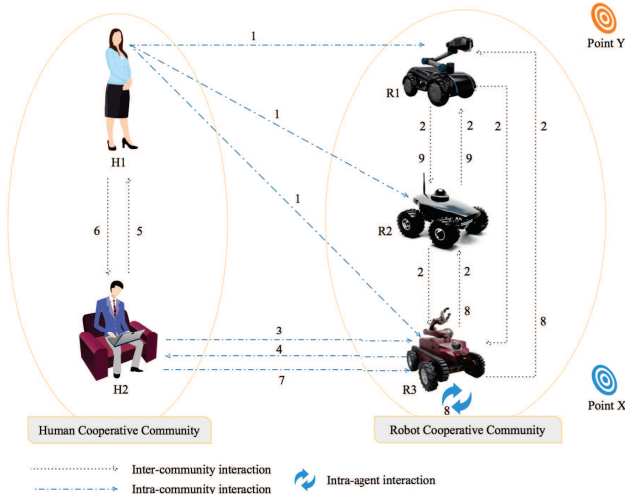


Fig. 4: Human-Robot Interaction (HRI) scenario.

first task_a and send a broadcast to the rest of robots in its community asking the next closest robot to Point ‘X’ to go there. Finally, R3 starts to perform the task_a, i.e. going to point ‘Y’.

- 9) As in step 2, but this time only R1 and R2 start to negotiate again with each other, and the closest one to ‘X’ has to move.

This scenario has been used to validate our ontology presented in Section III. For example, the conceptualization of the property *hasCommunicationWith* allows to automatically set which human has a communication with which robot. Hence, knowing that H2 interacts with R3, our ontology implementation can infer itself, by applying our defined concepts and relations (Sections III.A-E) which are implemented in OWL (Section III.F), and by using an integrated reasoner, that R3 *hasCommunicationWith* H2 (Fig. 5).

V. CONCLUSION

In this paper, AuR group’s first IEEE-RAS ontology standard for autonomous robots (ROA) has been presented. The needs for this ontology, its background information, and its proposed development have been provided. In particular, its concepts, architecture, and vocabulary in terms of core components, functions, behaviors, and structure have been described. Moreover, this ontology has been implemented, tested, and demonstrated in an HRI case study successfully.

Further developments of this work have the potential to achieve an ontology standard for autonomous robots. Hence, the group encourages researchers and industry to contribute to future standardization and development of this ontology.

REFERENCES

- [1] L. Paull, G. Severac, G. Raffo, J. Angel, H. Boley, P. Durst, W. Gray, M. Habib, B. Nguyen, S. V. Ragavan, S. Saeedi, R. Sanz, M. Seto, A. Stefanovski, M. Trentini, and H. Li, “Towards an ontology for autonomous robots,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012, pp. 1359–1364.

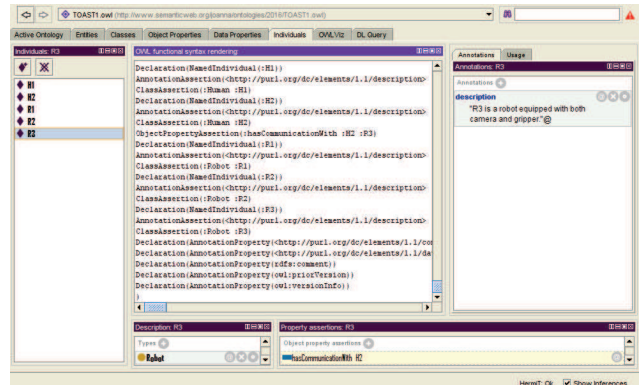


Fig. 5: Snapshot of our ontology run for the HRI scenario.

- [2] S. R. Fiorini, J. Bermejo-Alonso, P. Goncalves, E. P. de Freitas, A. O. Alarcos, J. I. Olszewska, E. Prestes, C. Schlenoff, S. V. Ragavan, S. Redfield, B. Spencer, and H. Li, “A suite of ontologies for robotics and automation,” *IEEE Robotics and Automation Magazine*, 2017.
- [3] B. Bayat, J. Bermejo-Alonso, J. L. Carbonera, T. Facchinetti, S. Fiorini, P. Goncalves, V. A. Jorge, M. Habib, A. Khamis, K. Melo, B. Nguyen, J. I. Olszewska, L. Paull, E. Prestes, V. Ragavan, S. Saeedi, R. Sanz, M. Seto, B. Spencer, A. Vosughi, and H. Li, “Requirements for building an ontology for autonomous robots,” *Industrial Robot: An International Journal*, vol. 43, no. 5, pp. 469–480, 2016.
- [4] T. R. Gruber, “Toward principles for the design of ontologies used for knowledge sharing?” *In. Journal of Human-Computer Studies*, vol. 43, no. 5, pp. 907–928, 1995.
- [5] J. I. Olszewska, “Clock-model-assisted agent’s spatial navigation,” in *Int. Conf. on Agents and Artificial Intelligence*, 2017, pp. 687–692.
- [6] M. Fernández-López, A. Gómez-Pérez, and N. Juristo, “Methontology: from ontological art towards ontological engineering,” in *AAAI Spring Symposium*, 1997.
- [7] M. Uschold and M. King, *Towards a methodology for building ontologies*, 1995.
- [8] E. Prestes, J. L. Carbonera, S. R. Fiorini, V. A. M. Jorge, M. Abel, R. Madhavan, A. Locoro, P. Goncalves, M. E. Barreto, M. Habib, A. Chibani, S. Gerard, Y. Amirat, and C. Schlenoff, “Towards a core ontology for robotics and automation,” *Robotics and Autonomous Systems*, vol. 61, no. 11, pp. 1193–1204, 2013, ubiquitous Robotics.
- [9] I. Niles and A. Pease, “Towards a standard upper ontology,” in *ACM Int. Conf. on Formal Ontology in Information Systems*, 2001, pp. 2–9.
- [10] J. S. Gero and U. Kannengiesser, “The situated function-behaviour-structure framework,” *Design Studies*, vol. 25, no. 4, pp. 373–391, 2004.
- [11] I. Johansson, “Pattern as an ontological category,” in *Conf. on Formal Ontology in Information Systems*, 1998, pp. 86–94.
- [12] IEEE SA, “IEEE Standard Ontologies for Robotics and Automation,” pp. 1–60, 2015.
- [13] J. I. Olszewska, “Spatio-temporal visual ontology,” in *BMVA/EACL EPSRC Workshop on Vision and Language (VL)*, 2011.
- [14] —, “Multi-target parametric active contours to support ontological domain representation,” in *18th RFIA Conference*, 2012, pp. 779–784.
- [15] J. Olszewska and T. McCluskey, “Ontology-coupled active contours for dynamic video scene understanding,” in *IEEE Int. Conf. on Intelligent Engineering Systems*, 2011, pp. 369–374.
- [16] J. I. Olszewska, “3D Spatial reasoning using the clock model,” in *Research and Development in Intelligent Systems*. Springer, 2015, vol. XXXII, pp. 147–154.
- [17] —, “Interest-point-based landmark computation for agents’ spatial description coordination,” in *Int. Conf. on Agents and Artificial Intelligence*, 2016, pp. 566–569.
- [18] A. Khamis, M. Kamel, and M. A. Salichs, “Human-robot interfaces for social interaction,” *Int. Journal of Robotics and Automation*, 2007.
- [19] A. Khamis, M. El-Gazzar, and M. A. Salichs, “Multiagent-based framework for multirobot systems,” in *Cooperation in Robotics*. University Rey Juan Carlos, 2009.