2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Defining Positioning in a Core Ontology for Robotics*

Joel Luis Carbonera[†], Sandro Rama Fiorini[†], Edson Prestes[†], Vitor A. M. Jorge[†], Mara Abel[†],
Raj Madhavan[‡], Angela Locoro[§], Paulo Gonçalves[¶], Tamás Haidegger[||], Marcos E. Barreto[**], Craig Schlenoff[††]
[†] Institute of Informatics, UFRGS, Brazil
[‡] Institute for Systems Research, University of Maryland, USA
[§] DIBRIS, Universit degli Studi di Genova, Italy
[¶] Polytechnic Institute of Castelo Branco / Technical University of Lisbon,
Center of Intelligent Systems, IDMEC / LAETA, Portugal
[||] ABC Center for Intelligent Robotics, Óbuda University, Budapest, Hungary
[**] Distributed Systems Laboratory, UFBA, Brazil
[††] Intelligent Systems Division, NIST, EUA

***Abstract*— Unambiguous definition of spatial position and orientation has crucial importance for robotics. In this paper we propose an ontology about positioning. It is part of a more extensive core ontology being developed by the IEEE RAS Working Group on ontologies for robotics and automation. The core ontology should provide a common ground for further ontology development in the field. We give a brief overview of concepts in the core ontology and then describe an integrated approach for representing quantitative and qualitative position information.**

I. Introduction

As robotic and automation systems evolve in complexity, the role of ontologies is becoming more apparent. In brief, ontologies can be viewed as an approach to describe the knowledge in a specific domain. The result of the process of building an ontology is a knowledge artifact, which formally describes the main concepts, relations, and axioms within a domain. The role of ontologies in robotics is twofold. They help to ensure a common understanding among various stakeholders involved in the life-cycle of robotics systems, and they also enable efficient and semantically reliable data integration and information exchange between robotic systems and between robots and other agents.

The *Ontology for Robotics and Automation Working Group* (ORA WG) [1] is an initiative within IEEE RAS with the goal of standardizing knowledge representation in the robotics field. We are actively working with organizations in industry, academia and government to develop a set of ontologies and an associated modeling methodology to be used as a standard in Robotics and Automation (R&A). The ORA WG intends to produce a series of ontologies that will describe the major sub-domains within R&A, such as industrial and service robotics. ORA WG comprises some sub-groups; the ours is called UpOM (Upper Ontology/Methodology). The main responsibility of UpOM is the development of the Core Ontology for R&A, which specifies the main concepts and relations spanning the whole field. It includes concepts such as robot, robotic system, robot part and so on. Its main goal is to serve as a pivot for integrating different sub-ontologies within the group, such as those for industrial and service robotics.

We have introduced the main concepts and commitments of the Core Ontology in previous works [1], [2]. In this paper, we further develop the Core Ontology by introducing the notion of object positioning. Positioning, orientation and pose are intrinsically spatial notions. Space is considered a trivial concept in common sense. However, as discussed in [3], the ontological nature of space (e.g. what is space?) and related notions, have been a subject of debates and controversies, resulting in several alternative conceptions. Moreover, the knowledge representation and qualitative reasoning communities identified several spatial aspects that are important for spatial reasoning, such as [4]: topology, orientation, shape, size, distance, positioning, etc. It is important to note that these aspects are usually handled individually, with specific knowledge representation and reasoning scheme, without a unified perspective. Nevertheless, a suitable ontological account for space-related concepts is necessary in order to improve the semantic interoperability among different robotic systems. This is one of the main pieces of information for allowing planning and movement.

The literature provides some approaches for representing spatial knowledge. For example, Bateman and Farrar [3] propose a unified ontological framework for representing qualitative (relative) positioning in space, but they do not provide explicit treatment of important quantitative positioning notions, like position of an object according to a coordinate system. On the other hand, other approaches, such as Ye et al. [5], represent positions, coordinate systems, and relative positions, however they do not make clear statements about

*Contact information: edson.prestes@ieee.org (Edson Prestes). The authors would like to thank IEEE RAS (Sponsor of the Working Group), IEEE-SA, CNPq, CAPES and Petrobras PRH PB-217 for supporting the work reported in this paper. This work is also supported by the Strategic Project, PEst-OE/EME/LA0022/2011, through FCT (under IDMEC-IST, Research Group: IDMEC/LAETA/CSI). Our thanks are also due to the members of the working group and other volunteers who attended the meetings and provided feedback.

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

their ontological commitments. For instance, they do not provide a clear formal description of what is a coordinate system.

We propose an ontology that provides an abstract and integrated account of quantitative and qualitative positioning. Our goal is *not* to give a full mathematical treatment to positioning, but rather to describe the main concepts and relations associated with positional information. Specializations of the Core Ontology shall "fill in" the specific mathematical details and representations required to employ these models in particular applications. The existence of a common general structure between application models should facilitate the exchange of information between different agents (e.g., robots and humans).

This paper is organized as follows. Section II describes the general aspects of ontologies by presenting some definitions of key notions. Section III gives a brief overview of our Core Ontology for R&A. Section IV presents and justifies the modeling for positioning and how it can be extended for orientation and pose. Finally, in Section V, we draw our conclusions and present our future steps.

II. Ontology: General Aspects

In computer science, ontologies are formal tools that enable the description of objects, properties and relationships among such objects in a given knowledge domain. According to Studer et al.[6], an ontology is "an explicit, formal specification of a shared conceptualization". On the other hand, Guarino [7] stresses the formal aspects of a conceptualization and defines ontologies as "logical theories accounting for the intended meaning of a formal vocabulary". An ontology comprises at least a set of terms and their definitions shared by a given community, formally specified in a machine-readable language, such as first-order logic. Ontologies are particularly important to provide machines with knowledge representation and reasoning capabilities to solve tasks, as well as to allow for semantic interoperability between heterogeneous systems.

The term ontology encompasses disparate ways of structuring its elements. From a mere list of terms and definitions to a formal theory; the structure of what has to be modeled changes dramatically at both extremes. Notwithstanding, the main elements of an ontology can be identified as: *classes*, which stand for concepts at all granularities; *relations*, which stand for associations between concepts; and *formal axioms*, which constrain and add consistency rules to the concept and relationship structures.

Disparate classifications are available for systematizing different kinds of ontologies. In this work, we use the classification based on the "level of generality", introduced in [7]. According to this criteria, ontologies can be classified in four main classes: *Top-level ontologies*, which describe very general concepts (such as space, time, matter, object, event, etc) that are independent of a particular problem or domain; *Domain ontologies*, which describe concepts of a specific domain, by specializing concepts in the top-level ontology; *Task ontologies*, which describe generic tasks or activities (like diagnosing or selling), also specializing the top-level ontology; and, finally, *application ontologies*, which are strictly related to a specific application, describing concepts depending both on a particular domain ontology and task ontology.

In Prestes et al. [2], we propose a *core ontology*. Not present in the classification above, core ontologies can be viewed as mid-level ontologies, positioned in between top-level and domain ontologies [8]. They provide a common definition of the most important concepts in some large domain, to which all other concepts are usually related. For instance, a core ontology for biology would define concepts such as organism, animal, cell, and so on. In robotics, as we shall see, a core ontology specifies concepts such as robot, device, and robotic system as well as their relationships. These concepts permeate other ontologies, such as ontologies for sensors, actuators, etc. Note that in this scenario, a core ontology plays an important role, providing a common foundation of the basic and generic (core) notions of the R&A domain that will be invoked across all the sub-domains. In this sense, the proposed core ontology provides strategies to extend the main generic terms to specific sub-domains and applications. This avoids *ad-hoc* solutions that can lead to an inconsistent set of ontologies.

III. The Core Ontology for Robotics

The development of the *core ontology for R&A* (*CORA*) at UpOM is supported by two well-known methodologies for building ontologies: METHONTOLOGY [9] and OntoClean [10].

METHONTOLOGY is an ontology engineering methodology for building ontologies either from scratch, by reuse, or re-engineering existing ones. In general, it provides a set of guidelines about how to carry out the activities identified in the ontology development process, the kinds of techniques that are the most appropriate in each activity, and the resulting products of each one.

OntoClean is a methodology for validating the ontological adequacy of taxonomic relationships. It is based on highly general ontological notions drawn from philosophy, like essence, identity, and unity. These notions are used to characterize relevant aspects of the intended meaning of the properties, classes, and relations that compose an ontology.

Also, as a result of an evaluation process carried out in [2], we selected the *Suggested Upper Merged Ontology* (SUMO) [11] as the most suitable top-level ontology for supporting the development of our core ontology. SUMO was developed by an IEEE working group and, according to our analysis, is flexible enough to fit well to the purposes of this project. Thus, CORA is being developed in integration with SUMO.

CORA is, naturally, about robots. Its main intent is to describe what a robot is and how it relates to other concepts. It defines four big broad entities: robot part, robot, complex robot and robotic system (Figure 1). In this paper, we are not going to delve into details about each concept, since they were presented in [2]. Instead, we provide a short description of each domain entity.

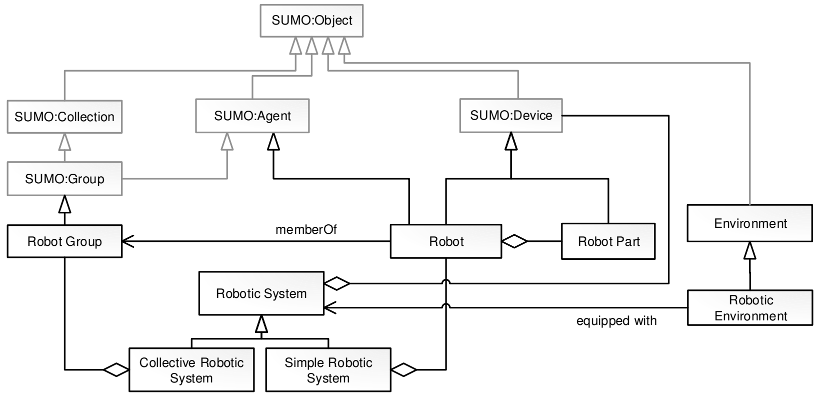

Fig. 1. Overview of the the core ontology for robotics and automation.

The term robot may have as many definitions as authors writing about the subject. This inherent ambiguity in this term might be an issue when one needs to specify an ontology for a broad community, like ours. We acknowledge this ambiguity as an intrinsic feature of the domain and, therefore, we decided to elaborate a definition based purely on necessary conditions, without specifying sufficient conditions. Thus, it is ensured that the CORA will cover all the entities the community actually considers as a robot, at the cost of classifying as a robot some entities which actually are not robots to some roboticists. However, the concepts in our ontology could be specialized according the needs of specific sub-domains or applications of R&A.

More importantly, we decided on a definition of robot that emphasizes its functional aspects. For our general purposes, *robots are agentive devices* in a broad sense, purposed to act in order to accomplish a task. In some cases, the actions of a robot might be subordinated to actions of other agents, such as software agents (bots) or humans. A robot is also a device, composed of suitable mechanical and electronic parts. Robots can form social groups, where they interact to achieve a common goal. A robot (or a group of robots) can form robotic systems together with other devices. An environment equipped with a robotic system is a robotic environment.

A *robot* is a *device* in the sense of SUMO. According to SUMO, a device is an artifact (e.g., a *physical object product of making*), which participates as a tool in a process. Naturally, a device can have parts. We define a specific concept called *Robot Part*, which classifies any other device that composes a robot, from nuts and bolts to manipulators and actuators. Theses devices only assume the role of *Robot Parts* when they are attached to the robot.

A robot is also an *agent*. SUMO states that agent is "*something or someone that can act on its own and produce changes in the world*". Robots perform tasks by acting on the environment or themselves. Action is strongly related to agency, in the sense that the acting defines the agent. A robot can form robotic groups. A robotic group is also an agent; in the sense that its own agency emerges from its participants. This notion can be used to describe robot teams, or even complex robots formed by many independent robotic agents acting in unison.

Robotic systems are systems composed of robots (groups of robots) and other devices that facilitate the operations of robots. A good example of a robotic system is a car assembly cell in a manufacturing site. It is located in an environment equipped with actuated structures that manipulate the car body, in a way that industrial robots can act on them. An environment equipped with a robotic system is a *robotic environment*.

More information about our Core Ontology can be obtained in [2]. Next, we shall concentrate in one particular aspect of this ontology which is left undeveloped and is the target of this paper: position.

IV. Position, orientation and pose

An important information regarding robots and other objects is their pose. It comprises position and orientation – all essential for tasks such as planning and navigation. As we have seen, it is possible to find in the literature all sorts of specialized models for representing position. Nevertheless, roboticists and other domain experts usually utilize two kinds of positional information [5]: *quantitative* and *qualitative* position. In the quantitative case, a position is represented by a *point* in a given coordinate system. On the qualitative case, a position is represented as a *region* defined in function of a reference object. For instance, one can say that a robot is positioned at the coordinates (x, y) in the global coordinate system, or that the robot is positioned in front of the box, where "in front of" comprises a conical region centered on the box and pointed forward.

In order to capture both notions in our ontology (represented in Figure 2), we sought inspiration in the way that SUMO represents time. According to SUMO, time is a physical quantity that can be attributed to any physical object. A *time measure* can be a *point* in time (i.e., a point in the one-dimensional space representing time), or an *interval* in time, which has points as temporal parts. SUMO does not allow for a similar construction for position in space. SUMO defines that a physical object must be *located* at some region in space and that objects can be oriented in different ways in relation to each other. However, this construction is not enough to allow for precise (quantitative) definitions of position.

We consider that a position q is essentially a *measure* (or *observation*) attributed to a (physical) object o, i.e.,

$$\forall o\ \forall q\ \mathsf{pos}(o,q)\ \rightarrow\ \mathsf{Object}(o)\ \wedge\ \mathsf{PMeasure}(q)\,, \quad (1)$$

where $\mathsf{pos}(o,q)$ means that the object o has a position measure q. Such as with time in SUMO, we introduce the notions of *position point* and *position region*. A position point refers to a point in a coordinate system projected on the physical space. A position region is an abstract region in a coordinate system overlapping the physical spatial region occupied by the object. Both position point and position region are types of position measurement; i.e., $\forall p\ \mathsf{PMeasure}(q)\ \leftrightarrow\ \mathsf{PPoint}(q)\ \vee\ \mathsf{PRegion}(q)$ and $\forall q\ \mathsf{PPoint}(q)\ \rightarrow\ \neg\mathsf{PRegion}(q)$.

Also, it is important to note that all these definitions are synchronic; i.e., they consider only situations like snapshots in time. As such, two objects cannot have the exact same quantitative position; i.e., they can not be located at the same position point.

A coordinate system c is an abstract entity which is defined in relation of a single reference object o; i.e.,

$$\forall c\ \mathsf{CS}(c)\ \rightarrow\ \exists! o\ \mathsf{Object}(o)\ \wedge\ \mathsf{ref}(c,o)\,, \quad (2)$$

where $\mathsf{ref}(c,o)$ is true if o is the reference object of the coordinate system c[1]. For instance, the local coordinate system of a robot is referenced by the robot itself. Additionally, the reference object does not need to be *necessarily* at the origin of the coordinate system.

A position point denotes the quantitative position of an object in a coordinate system. Position points are always defined in a single coordinate system (CS):

$$\forall p\ \mathsf{PPoint}(p)\ \rightarrow\ \exists! c\ \mathsf{CS}(c)\ \wedge\ \mathsf{in}(p,c)\,, \quad (3)$$

where the predicate $\mathsf{in}(x,y)$ is true if x is a point in a coordinate system y.

This ontology does not commit to a particular kind of coordinate system. However, a coordinate system defines at least one dimension in which points get their coordinate values. A n-dimensional coordinate system c is homeomorphic to a subset of $\mathbb{R}^n$, such that a point $p \in c$ can represented as n-tuple

$$\varphi(p) = (x_1(p), x_2(p), \ldots, x_n(p)).$$

In this context, x_i is a coordinate function that attributes to p a real value in the dimension i of the coordinate system [12].

A fundamental aspect of coordinate systems is the notion of *transformation* (denoted by the predicate T). Points in a coordinate system can be mapped to another coordinate system by means of a transformation. Let the predicate $\mathsf{maps}_{\mathsf{CS}}(c,c_r,m)$ denote the mapping from a coordinate system c to another coordinate system c_r by means of a transformation m; and the predicate $\mathsf{maps}_{\circ}(p_1,p_2,m)$ denote the mapping from a point p_1 in a given coordinate system to the point p_2 in another coordinate system by a transformation m. More formally,

$$\begin{aligned}&\forall c,c_r\ \forall m\ \mathsf{maps}_{\mathsf{CS}}(c,c_r,m)\\ &\quad \rightarrow\ \forall p_1\ [\mathsf{in}(p_1,c)\\ &\qquad \rightarrow\ \exists! p_2\ [\mathsf{in}(p_2,c_r)\ \wedge\ \mathsf{maps}_{\circ}(p_1,p_2,m)]].\end{aligned} \quad (4)$$

The relation $\mathsf{maps}_{\mathsf{CS}}$ can be defined to be transitive if we assume transformations can be composed. Let the predicate $\mathsf{comp}(m_1,m_2,m)$ the composition of the transformations[2] m_1 and m_2 to into m, then the transitivity of coordinate space mappings can be defined as

$$\begin{aligned}&\forall c_1,c_2,c_3\ \forall m_1,m_2\ \mathsf{maps}_{\mathsf{CS}}(c_1,c_2,m_1)\\ &\qquad \wedge\ \mathsf{maps}_{\mathsf{CS}}(c_2,c_3,m_2)\\ &\quad \rightarrow\ \exists m\ \mathsf{maps}_{\mathsf{CS}}(c_1,c_3,m)\ \wedge\ \mathsf{comp}(m_1,m_2,m)\,.\end{aligned} \quad (5)$$

Furthermore, an object can display multiple positions in different coordinate systems only if there is a transformation that can map between the two; i.e.,

$$\begin{aligned}&\forall o\ \forall p,p_1\ \mathsf{pos}(o,p)\ \wedge\ \mathsf{pos}(o,p_1)\\ &\quad \rightarrow\ \exists! c\ \exists! c_1\ \mathsf{in}(p,c)\ \wedge\ \mathsf{in}(p_1,c_1)\ \wedge\ c \neq c_1\\ &\quad \wedge\ \exists m_1,m_2\ [\mathsf{maps}_{\circ}(p,p_1,m_1)\ \wedge\ \mathsf{maps}_{\circ}(p_1,p,m_2)].\end{aligned} \quad (6)$$

In Robotics (as in other disciplines), coordinate systems are also related through hierarchies (i.e. trees). Usually, an agent chooses an arbitrary coordinate system as the global reference frame, which constitutes the *global coordinate system* (GCS) for that agent. *Local coordinate systems* (LCS) are defined in relation to GCS by hierarchical links. Let the predicate $\mathsf{parent}_{\mathsf{CS}}(c_1,c_2)$ denote that the coordinate system c_2 is defined in c_1. Naturally, if $\mathsf{parent}_{\mathsf{CS}}(c_1,c_2)$ then there is a transformation m_1 such that $\mathsf{maps}_{\mathsf{CS}}(c_2,c_1,m_1)$, as well as a transformation m_2 such that $\mathsf{maps}_{\mathsf{CS}}(c_1,c_2,m_2)$. Note that the simple existence of a transformation does not imply the existence of a hierarchy. The hierarchy is ultimately defined by the agent. Furthermore, if $\mathsf{parent}_{\mathsf{CS}}(c_1,c_2)$, then also the referential object of c_2 has a position point in c_1.

Mappings between arbitrary coordinate systems can be constructed by composing transformation from and to a common ancestor. The ancestor c_1 of a given coordinate system c_3 can be defined as:

$$\begin{aligned}&\forall c_1,c_3\ \mathsf{ancestor}_{\mathsf{CS}}(c_1,c_3)\\ &\quad \rightarrow\ \mathsf{parent}_{\mathsf{CS}}(c_1,c_3)\\ &\qquad \vee\ \exists c_2\ [\mathsf{parent}_{\mathsf{CS}}(c_1,c_2)\ \wedge\ \mathsf{ancestor}_{\mathsf{CS}}(c_2,c_3)].\end{aligned} \quad (7)$$

Given a common ancestor c of the coordinate systems c_1 and c_2, then the transformation of c_1 into c_2 is given by a

[1] Note that $\exists!$ means "*there is one and only one*".

[2] For example, if A, B and C are transformation matrices, and C is the resulting matrix from the composition of A and B, then $C = AB$.

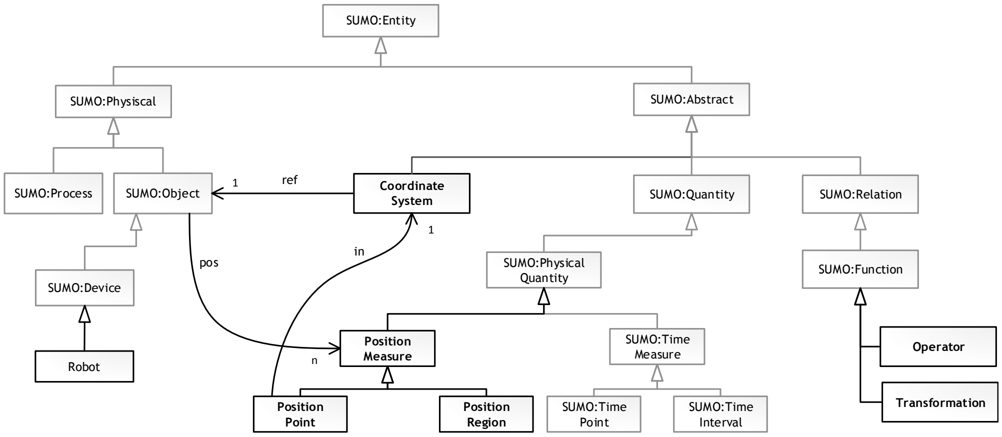

Fig. 2. Overview of concepts and relations about positioning in robotics and automation ontology.

transformation m; i.e.

$$\begin{aligned} &\forall c_1, c_2\ \exists c\ \exists m\ \mathsf{ancestor_{CS}}(c, c_1) \\ &\quad \wedge\ \mathsf{ancestor_{CS}}(c, c_2)\ \wedge\ \mathsf{maps_{CS}}(c_1, c_2, m) \\ &\quad \rightarrow\ \exists m_1, m_2\ [\mathsf{maps_{CS}}(c_1, c, m_1) \\ &\quad \wedge\ \mathsf{maps_{CS}}(c, c_2, m_2)\ \wedge\ \mathsf{comp}(m_1, m_2, m)] \end{aligned} \tag{8}$$

It follows from (3) and (2) that the quantitative position of an object is a point in a coordinate system, which is in turn grounded in a particular object. In certain cases, it is more interesting to define the position of an object in relation to the actual object that grounds the coordinate system. For that we introduce a predicate $\mathsf{pos_{rel}}$, such that for any object o and reference object o_r:

$$\begin{aligned} &\forall o\ \forall p\ \forall o_r\ \mathsf{pos_{rel}}(o, p, o_r)\ \wedge\ \mathsf{PPoint}(p) \\ &\quad \rightarrow\ \mathsf{pos}(o, p) \\ &\quad \wedge\ \exists c\ [\mathsf{CS}(c)\ \wedge\ \mathsf{in}(p, c)\ \wedge\ \mathsf{ref}(c, o_r)]. \end{aligned} \tag{9}$$

Now, we can introduce qualitative positioning between objects. As already stated earlier, qualitative positions are defined in terms of position regions. Example of qualitative positions are “left of”, “in front of”, “on top of”, etc. These expressions define regions in relation to a reference object o_r in which other objects are placed. More specifically, a position region s is defined by position points in a coordinate system c. Consider an overloaded version of the predicate $\mathsf{in}(x, y)$ that also holds if x is a point in a position region y. Thus,

$$\begin{aligned} &\forall s\ \mathsf{PRegion}(s)\ \rightarrow\ \exists! c\ \mathsf{CS}(c) \\ &\quad \wedge\ \forall p\ \mathsf{PPoint}(p)\ \wedge\ [\mathsf{in}(p, s)\ \rightarrow\ \mathsf{in}(p, c)]. \end{aligned} \tag{10}$$

A position region is always *generated* by a *spatial operator* g applied on a reference object o_r:

$$\begin{aligned} &\forall s\ \mathsf{PRegion}(s) \\ &\quad \rightarrow\ \exists! o_r\ \exists g\ \mathsf{SOperator}(g)\ \wedge\ \mathsf{generated}(s, o_r, g) \\ &\quad \wedge\ \mathsf{ref}(c, o_r)\ \wedge\ \forall p\ [\mathsf{in}(p, s)\ \rightarrow\ \mathsf{in}(p, c)]. \end{aligned} \tag{11}$$

The predicate $\mathsf{generated}(s, o_r, g)$ holds when the region s is generated by the operator g applied on the reference object o_r. A spatial operator can be seen as a mathematical function that can map reference objects to regions in a coordinate system.

The actual qualitative position of an object is given by the position regions that the object overlaps. Let ext be a function mapping an object o to a position region corresponding to its spatial extension (e.g. the volume occupied by the object), we can say that o has a qualitative position if it *overlaps* with the position region: i.e.

$$\begin{aligned} &\forall o\ \forall s\ \mathsf{Object}(o)\ \wedge\ \mathsf{PRegion}(s)\ \wedge\ \mathsf{pos}(o, s) \\ &\quad \rightarrow\ \mathsf{overlaps}(\mathsf{ext}(o)\, , s)\, . \end{aligned} \tag{12}$$

The predicate binary overlaps has the same intuitive interpretation of the overlaps predicate in RCC-8 [4]. That is, if the two regions share at least a point.

We can also reuse the relation $\mathsf{pos_{rel}}$ to explicitly define any qualitative position s between any object o and any reference object o_r.

$$\begin{aligned} &\forall o\ \forall s\ \forall o_r\ \mathsf{PRegion}(s)\ \wedge\ \mathsf{pos_{rel}}(o, s, o_r) \\ &\quad \rightarrow\ \exists g\ [\mathsf{pos}(o, s)\ \wedge\ \mathsf{generated}(s, o_r, g)]. \end{aligned} \tag{13}$$

For example, consider an operator leftOfOp that takes the reference object and generates a conical position region left representing the left region of the reference object. In this case, the proposition $\mathsf{pos_{rel}}(o, \mathsf{left}, o_r)$ means that the object o is positioned at the left of the reference object o_r (according to the operator leftOfOp).

However, it is more natural in some contexts to define qualitative positioning as relations between objects. This can be easily achieved by defining these relations as abstractions of classes of position regions generated by a given operator. For instance, the relation $\mathsf{leftOf}(o, o_r)$ between the objects o and o_r can be defined in the following way:

$$\begin{aligned} &\forall o\ \forall o_r\ \mathsf{leftOf}(o, o_r)\ \rightarrow\ \exists s\ \mathsf{pos_{rel}}(o, s, o_r) \\ &\qquad\qquad \wedge\ \mathsf{generated}(s, o_r, \mathsf{leftOfOp})\, , \end{aligned} \tag{14}$$

where leftOfOp is a constant denoting the operator that generates the left region given a reference object o_r. This same scheme can be used to define other qualitative relations between object.

Again here, we do not commit to any particular formalism to represent positional regions, nor any particular kind of operator. We can however indicate some general axioms about different kinds of qualitative positions. For instance, our ontology provides means for representing notions, such as $\forall o\ \forall o_r\ \mathsf{leftOf}(o, o_r) \rightarrow \neg\mathsf{rightOf}(o, o_r)$; and $\forall o\ \forall o_r\ \mathsf{frontOf}(o, o_r) \rightarrow \neg\mathsf{backOf}(o, o_r)$. This sort of modeling allows us to reuse the spatial attributes in SUMO. SUMO defines qualitative positioning by a 3-place predicate called orientation, with a similar structure as to our $\mathsf{pos}_{\mathsf{rel}}$. However, SUMO define types of position regions as simple "spatial attributes", which are disjoint to regions.

The usual notion of orientation is analogue to position regarding formal structure, including notions such as *orientation measure*, *orientation point* and *orientation region*. Given the space restrictions, we only give a brief overview. An object can have a *quantitative* orientation defined as an *orientation point* in an orientation coordinate system, as well as a *qualitative orientation* defined as an *orientation region* in relation to a reference object. For instance, an example of use of orientation point is in "the robot is oriented 54 degrees in relation to the reference object". As it happens with position points, orientation points in one coordinate system can be mapped to other coordinate systems. On the other hand, orientation regions capture a less intuitive notion. The expression "the robot is orientated to north" allows for interpretations where the robot is generally pointed towards an interval of orientation values around 0 degrees in a compass. Thus, we can model "north" as a region (or interval) that overlaps with the general orientational extension of the object. Note that, eventually, position regions and orientation regions can be denoted by similar words. For instance, one can say a robot is *at the north*, *facing north*. The former relates to a position region; i.e., the north region of a given country; the later relates a orientation region; i.e., the interval around north on the compass.

A position and an orientation constitute a pose. The pose of an object is the description of any position and orientation bearing the same object:

$$\begin{aligned}
&\forall o\ \forall e\ \mathsf{pose}(o, e) \\
&\quad \rightarrow \exists x\ \exists y\ \mathsf{Pose}(e) \\
&\qquad \wedge\ \mathsf{PMeasure}(x) \wedge \mathsf{hasPosition}(e, x) \\
&\qquad \wedge\ \mathsf{OMeasure}(y) \wedge \mathsf{hasOrientation}(e, y)
\end{aligned} \tag{15}$$

Often, a pose is defined with a position and an orientation to different coordinate systems/reference objects. Also, since objects can have many different positions and orientation, they can also have many different poses.

V. Concluding Remarks and Future Work

In this article, we described the ongoing work of the IEEE RAS ORA/UpOM group in developing a core ontology for robotics and automation. In particular, we discussed the notions of qualitative and quantitative positioning, orientation and pose for robots, highlighting some of the ontological commitments of our ontology. We expect that our general definitions for these notions will serve as a common ground for other sub-ontologies in the working group to build upon. The inclusion of a general ontological account of these notions in the core ontology is a necessary step for allowing the integration of the other sub-ontologies, preserving the interoperability. We hope that the final ontology defines the key-elements that will allow for unambiguous communication between humans and/or robots. Furthermore, it could be widely used within our community, either by researchers, consumers or institutions.

The next step is to map or align the notions presented here with the same concepts in other ontologies within the group. We believe that the notions of qualitative and quantitative position can be unified if we assume a common ontological characterization of physical space in terms of topological manifolds; we plan to further investigate this possibility. Also, a possible extension to this theory is to include time in the predicates, yielding a diachronic ontology of positioning.

References

[1] C. Schlenoff, E. Prestes, R. Madhavan, P. Goncalves, H. Li, S. Balakirsky, T. Kramer, and E. Miguelanez, "An ieee standard ontology for robotics and automation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vilamoura, Algarve, Portugal: Springer-Verlag, 2012, pp. 1337–1342.

[2] E. Prestes, J. L. Carbonera, S. R. Fiorini, V. A. M. Jorge, M. Abel, R. Madhavan, A. Locoro, P. Gonçalves, M. E. Barreto, M. Habib, A. Chibani, S. Grard, Y. Amirat, and C. Schlenoff, "Towards a core ontology for robotics and automation," *Robotics and Autonomous Systems*, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2013.04.005

[3] J. Bateman and S. Farrar, "Modelling models of robot navigation using formal spatial ontology," in *Spatial Cognition IV. Reasoning, Action, Interaction*, ser. Lecture Notes in Computer Science, C. Freksa, M. Knauff, B. Krieg-Brckner, B. Nebel, and T. Barkowsky, Eds. Springer Berlin Heidelberg, 2005, vol. 3343, pp. 366–389. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-32255-9_21

[4] A. G. Cohn and J. Renz, *Handbook of Knowledge Representation*. Elsevier, 2007, ch. Qualitative Spatial Representation and Reasoning, pp. 551–596.

[5] J. Ye, L. Coyle, S. Dobson, and P. Nixon, "A unified semantics space model," in *Location- and Context-Awareness*, ser. Lecture Notes in Computer Science, J. Hightower, B. Schiele, and T. Strang, Eds. Springer Berlin Heidelberg, 2007, vol. 4718, pp. 103–120.

[6] R. Studer, V. R. Benjamins, and D. Fensel, "Knowledge engineering: Principles and methods," *Data and Knowledge Engineering*, vol. 25, no. 1-2, pp. 161–197, March 1998.

[7] N. Guarino, "Formal ontology in information systems," in *Proceedings of International Conference on Formal Ontology in Information Systems (FOIS 98)*, N. Guarino, Ed. Trento, Italy: IOS Press, 1998, pp. 3–15.

[8] L. Obrst, *Theory and Applications of Ontology: Computer Applications*. Springer, 2010, ch. Ontological architectures, pp. 27–66.

[9] M. Fernndez, A. Gmez-Prez, and N. Juristo, "METHONTOLOGY: from ontological art towards ontological engineering," in *Ontological Engineering*, ser. AAAI Spring Symposium, vol. 06, 1997, AAAI Spring Symposium, pp. 33–40.

[10] N. Guarino and C. A. Welty, "An overview of OntoClean," in *Handbook on Ontologies*, ser. International Handbooks on Information Systems, S. Staab and R. Studer, Eds. Springer Berlin Heidelberg, Jan. 2009, pp. 201–220.

[11] I. Niles and A. Pease, "Towards a standard upper ontology," in *Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001*, ser. FOIS '01. New York, NY, USA: ACM, 2001, p. 29.

[12] S. Morita, *Geometry of differential forms*, ser. Translations of mathematical monographs. Providence, R.I: American Mathematical Society, 2001, no. v. 201.