Extensions to the Core Ontology for Robotics and Automation

Sandro Rama Fiorini[a,*], Joel Luis Carbonera[a], Paulo Gonçalves[b,c], Vitor A. M. Jorge[a], Vítor Fortes Rey[a], Tamás Haidegger[d,e], Mara Abel[a], Signe A. Redeld[f], Stephen Balakirsky[g], Veera Ragavan[h], Howard Li[i], Craig Schlenoff[j], Edson Prestes[a]

[a]*Instituto de Informática, UFRGS, Brazil*
[b]*Polytechnic Institute of Castelo Branco, School of Technology, Portugal*
[c]*LAETA, IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal*
[d]*Óbuda University, Budapest, Hungary*
[e]*Austrian Center for Medical Innovation and Technology (ACMIT), Wiener Neustadt, Austria*
[f]*Naval Surface Warfare Center, USA*
[g]~~*Aerospace, Transportation & Advanced Systems Laboratory (ATAS)*~~*, Georgia Tech Research Institute, USA*
[h]*School of Engineering, Monash University, Sunway Campus, Malaysia*
[i]*Dept. of Electrical and Computer Engineering, University of New Brunswick, Canada*
[j]*Intelligent Systems Division, NIST, USA*

Abstract

The working group *Ontologies for Robotics and Automation*, sponsored by the *IEEE Robotics & Automation Society*, recently proposed a Core Ontology for Robotics and Automation (CORA). This ontology was developed to provide an unambiguous definition of core notions of robotics and related topics. It is based on SUMO, a top-level ontology of general concepts, and on ISO 8373:2012 standard, developed by the ISO/TC184/SC2 Working Group,

*Corresponding author

Email addresses: srfiorini@inf.ufrgs.br (Sandro Rama Fiorini), jlcarbonera@inf.ufrgs.br (Joel Luis Carbonera), paulo.goncalves@ipcb.pt (Paulo Gonçalves), vamjorge@inf.ufrgs.br (Vitor A. M. Jorge), vfrey@inf.ufrgs.br (Vítor Fortes Rey), haidegger@ieee.org (Tamás Haidegger), marabel@inf.ufrgs.br (Mara Abel), signe@ieee.org (Signe A. Redeld), stephen.balakirsky@gtri.gatech.edu (Stephen Balakirsky), veera.ragavan@monash.edu (Veera Ragavan), vhoward@unb.ca (Howard Li), craig.schlenoff@nist.gov (Craig Schlenoff), edson.prestes@ieee.org (Edson Prestes)

which defines — in a natural language — important terms in the domain of Robotics and Automation (R&A). In this paper, we introduce a set of ontologies that complement CORA with notions such as industrial design and positioning. We also introduce updates to CORA in order to give a more ontologically sound account of autonomy and representation of robot parts.

Keywords: Ontologies for robotics and automation, Ontology-based standards, Core ontology, Ontology engineering, Knowledge representation.

1. Introduction

A well-structured *body of knowledge* for ~~robotics~~ is a crucial requirement for unambiguous communication and reasoning not only for robots, but also for ~~knowledge and information sharing about robots among~~ humans and ~~for~~ human-robot interaction. Recently, such bodies of knowledge have been successfully developed using ontologies. Ontologies are information artifacts that specify in a *formal* and *explicit* way the domain knowledge *shared* by a community [1]. The availability of well-founded methodologies allow us to develop ontologies in a principled way. The artifacts that result from this process ensure a mutual agreement among stakeholders, increase the potential of reuse of the knowledge and promote data integration.

In order to specify and clarify the meaning of the core notions common in robotics and automation (R&A), the working group *Ontologies for Robotics and Automation* (ORA), sponsored by the *IEEE Robotics & Automation Society*, have proposed a *Core Ontology for Robotics and Automation* (*CORA*). This ontology is meant to be used by robots and roboticists in tasks that require explicit knowledge about robots, such as robot-robot/robot-human communication, robot design and integration of data about robots. The ORA WG aims to standardize knowledge representation in R&A field [2]. Within this broad context, CORA is intended to provide the core conceptual structure that will integrate other specific ontologies developed for the domain of R&A.

CORA has been developed taking into account notions of formal ontology. In particular, we evaluated many of our ontological choices following guidelines proposed by known methodologies, such as METHONTOLOGY [3] and OntoClean [4]. In particular, CORA specializes SUMO [5], a top-level ontology that aims to define the main ontological categories describing the world. Such an approach is new in developing standards in R&A and has the advantage of producing better founded standard, which requires less work to use, maintain and extend.

This work reports the recent development of the ongoing CORA project, and provides an overview of its current state. The previous version of CORA [6] was extended, introducing new concepts and relations that ~~have~~ been omitted. Thus, this paper presents some changes in modelling decisions that ~~has~~ been implemented since the previous version. The major new contribution can be divided in two broad areas. First, we propose CORAX, an ontology that covers concepts too general to be part of CORA, and that are not covered by SUMO. These include knowledge about *design* (as in the case of product design), *physical environment*, *interaction* and *artificial systems*. ~~Moreover,~~ we ~~have~~ propose~~d~~ extensions and changes to CORA itself, in order to improve its ontological commitment to the domain. Mainly, we are concerned with representation of *operation modes* and *robot parts*. Finally, we discuss some directions regarding new topics~~,~~ yet to be covered (such as control and planning).

2. Ontology Engineering

We developed CORA using a series of ontology tools and frameworks. The main methodology is based on METHONTOLOGY [3], an ontology engineering methodology for constructing ontologies. It provides a methodology for building ontolog~~y~~ either from scratch, by reuse, or re-engineering existing ones. In general, it consists of 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 for each activity, and the resulting products.

We also based many of the underlying *ontological commitments* on *OntoClean* [4]. Ontoclean is a methodology for validating the ontological adequacy of taxonomic relationships. It is based on highly generic 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. OntoClean induces the ontology engineer to make explicit the ontological commitments underlying the concepts that are being modelled. As a result, OntoClean allowed us to identify ambiguities in the definitions provided by other standards to some core notions of R&A (see [6] for more details).

In addition, as a result of an evaluation process carried out in [6], we selected the *Suggested Upper Merged Ontology* (SUMO)[1] [5] as the most suitable top-level ontology for supporting the development of CORA. SUMO was developed by an IEEE working group, and according to our analysis, it is flexible enough to fit the purposes of the project. It includes the main notions and distinctions we would like to introduce in our ontology, such as *agent*, *device* and *agent group*. All concepts in CORA and related ontologies are specializations of concepts in SUMO.

SUMO defines the basic ontological categories across all domains. The remainder of this section gives a brief overview of its main concepts, illustrated in Fig. 1. Detailed information can be found in [5].

The main SUMO category is *Entity*, which is a disjoint partition of *Physical* and *Abstract* entities. Physical represents entities that have a location in space-time. Abstract describes entities that do not have a location in space-time.

Physical is further partitioned into *Object* and *Process*. Object exists in space, keeping its identity in time, and ~~have~~ spatial parts but not temporal parts. Process is the class of instances that happen in time and have temporal parts or stages. This means SUMO follows an *endurantist* perspective instead of a *perdurantist* one. For a perdurantist, an object is composed by every temporal part it has at all times. On the other hand, for an endurantist, an object changes through time, but keeps essential parts that define its identity. A good analogy is to think that perdurantists see object as tunnel-like regions in a 4D space, while endurantists see them as 3D region that travels through the time dimension.

Abstract is further partitioned into *Quantity, Attribute, SetOrClass, Relation* and *Proposition*. Quantity abstracts numeric and physical quantities. Attribute abstracts qualities that cannot or are chosen not to be considered as subclasses of Object. SetOrClass abstracts entities that have *elements* (in the case of sets) or *instances* (in the case of classes). Relation generalizes n-ary relations, functions and lists. Finally, *Propositions* are entities that express a complete thought or a set of such thoughts.

[1] `http://www.ontologyportal.org/`

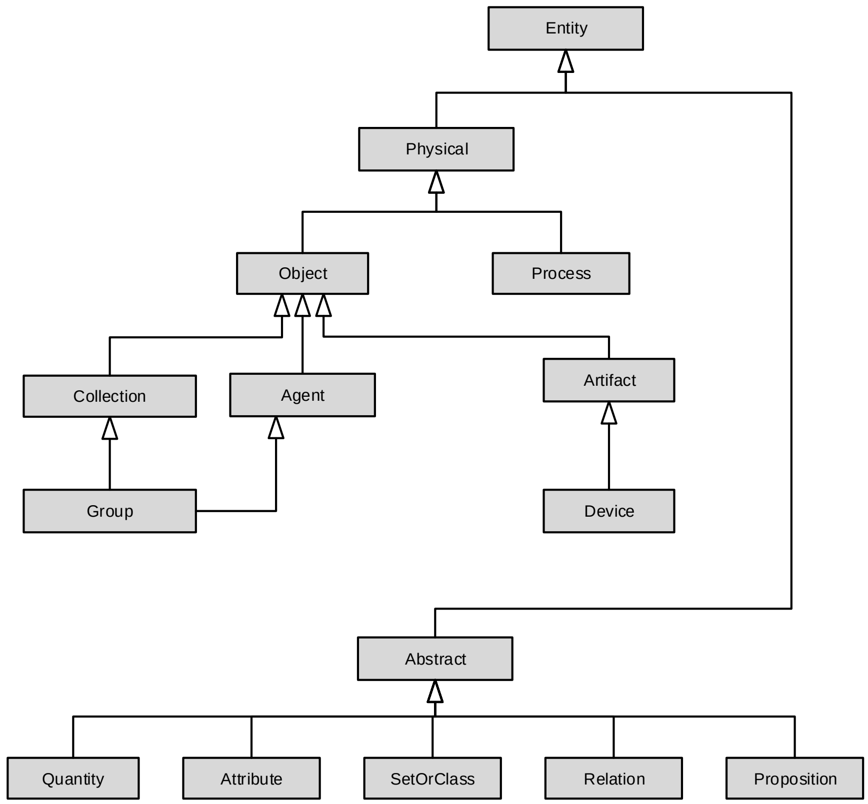

Figure 1: Overview of top-level concepts of SUMO.

3. Overview of CORA

CORA aims to describe what a robot is and how its concept relates to other concepts. It defines three broad entities: *robot*, *robot group* and *robotic system* (Fig. 2). In this paper, we are not going to delve into details about each concept, since they were presented in [6]. Instead, we provide a short description of each domain entity.

The term *robot* may have as many definitions as people 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. 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, our goal is to ensure that CORA covers

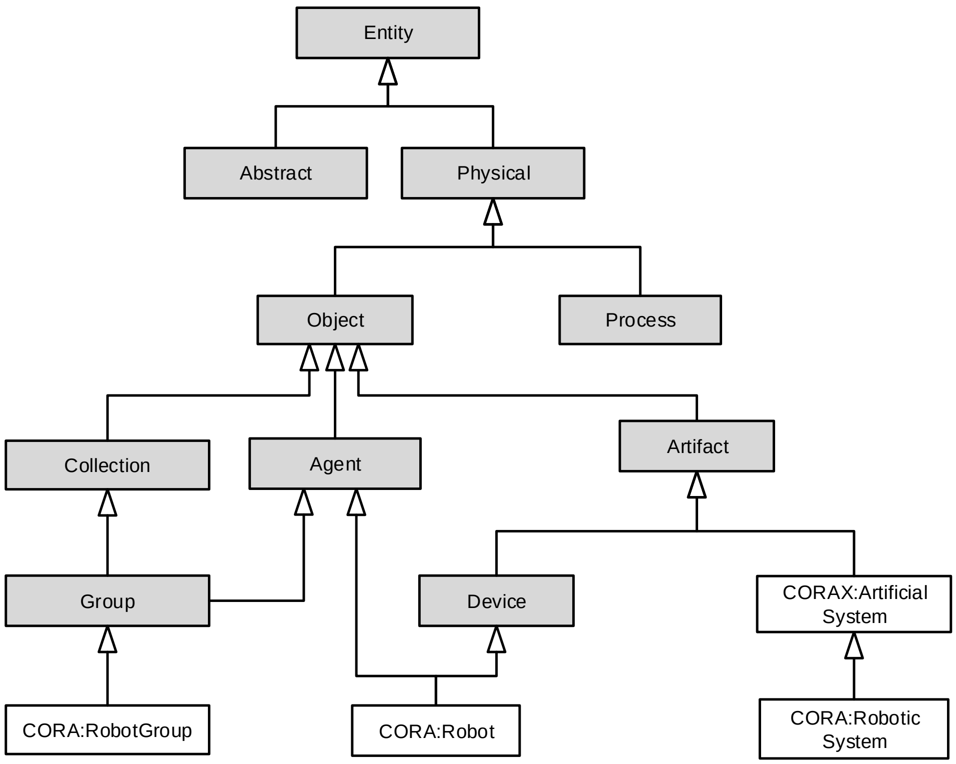

Figure 2: Overview of the main concepts in CORA: *robot*, *robot group* and *robotic system*

~~probably~~ most of the entities that the community actually considers as a robot, at the cost of classifying as a robot some entities that actually are not robots in the point of view of some roboticists. However, the concepts in our ontology could be specialized according to the needs of specific sub-domains or applications of R&A.

More importantly, we introduced 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. Robots are also *devices*, 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 (i.e., a *physical object product of making*), which participates as a tool in a process. Being a device, robot inherits from SUMO the notion that devices have parts. Therefore, CORA allows one to represent complex robots with robots parts.

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 robot groups. A *robot 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 (or robot groups) and other devices that facilitate the operations of robots. A good example of a robotic system is a car assembly cell at 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. Finally, as previously stated, an environment equipped with a robotic system is a *robotic environment*. See [6, 7] for a more detailed discussion on CORA's main concepts. Next, we describe new notions that have been integrated to CORA.

4. Updating CORA

CORA has been updated since its initial proposal [6, 7]. The main driving force behind these changes came from aligning it with existing ontologies and more expert involvement in the ~~developing~~ process. We compared CORA with an application *ontology for kitting* developed within the group [8]. Our objective was to investigate whether or not both ontologies could be merged and to check whether all notions in the kitting ontology were present in the combination of SUMO and CORA. This merging process led us to discover important concepts and relations present in kitting ontology that were not covered by ~~n~~either CORA ~~n~~or SUMO. Based on that, we developed a series of new ontology modules to bridge the gap between SUMO and the kitting ontology, which are mostly covered by CORAX and the position ontologies.

Furthermore, more involvement of independent experts and feedback was received after ~~that~~ the preliminary ~~standard~~ draft was completed. Apparently, experts were more comfortable to discuss concepts and relations after a first set of ontological commitments was made. The initial model served as a pivot for articulating new requirements on the ontology. Since it was based on well-founded ontological commitments, the model was more resilient to ad-hoc proposals to change it, translating into a more stable evolution of the ontology. Notably, changes were more eminent in aspects of the ontology that were not well founded in the first version of the ontology, such as autonomy.

In the following sections, we describe the changes made in and around CORA as a result of that process. They consist mostly of sub-ontologies complementing or specializing CORA (see Fig. 3).

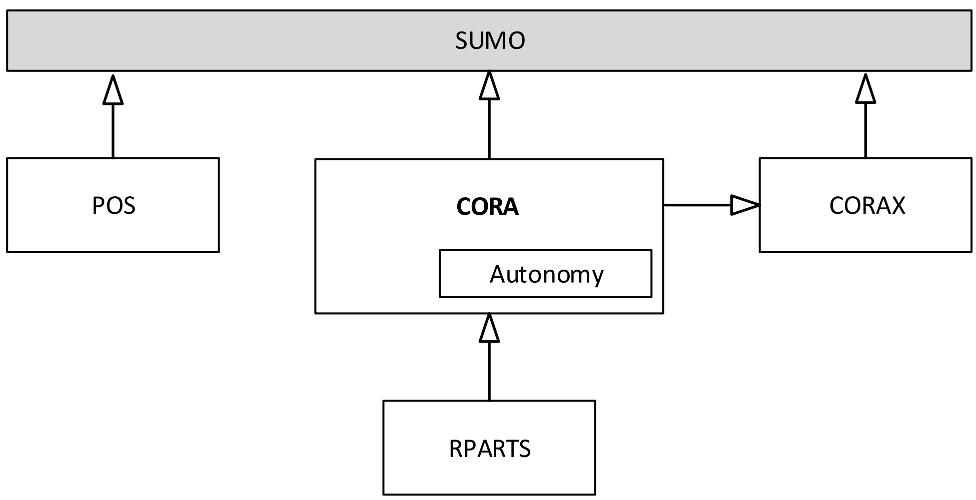

Figure 3: Extensions made to CORA and SUMO. CORAX, POS and RPARTS are extensions made to SUMO and CORA. The way CORA represents autonomy was also updated.

5. CORAX: connecting CORA and SUMO

Naturally, SUMO does not cover every possible aspect of reality, even when we restrict ourselves to R&A. At the same time, some of parts of reality are too general to be included in CORA. Due to this fact, we introduced the CORAX ontology, which plays a role bridging SUMO and CORA. In particular, CORAX includes concepts and relations associated with design, interaction and environment that are not covered in SUMO.

5.1. Design

Design is an important concept in engineering, specially in manufacturing. In R&A, the concept is frequently related to industrial robotics, where robots perform the job of building artifacts. Those robots have to know the design of the artifacts they are building in order to coordinate their actions.

A design is an abstract entity; it does not have materiality in itself. Rather, *content-bearing objects* (in SUMO), such as manuals and blueprints, give materiality to a design. One could reason ~~that~~ this in ~~the other way~~ ~~around:~~ a design is what links a series of related *blueprints*; it is the common abstract content that is represented in different blueprints. Furthermore, an artifact is related to a particular design, so that one should expect that the *artifact* realizes the design.

In our point of view, SUMO does not provide a good specification of design. One of its sub-ontologies—namely the engineering ontology—defines the concept *Model*, which is a type of abstract entity that seems to capture the notion of design described above. However, a model is not clearly related to content bearing objects, ~~neither~~ to *artifacts* in general. SUMO defines the *models* relationship between *Model* and *Engineering Component*, which is too restrictive to our purposes; since we would like to represent models of any kind of artifact.

In response to ~~that,~~ we defined the concept of *Design*, which is a kind of *Proposition*. According to SUMO, a *proposition* is an abstract entity that express a complete thought or a set of thoughts. For instance, the phrases "*the cat is on the mat*" and "*o gato está no tapete*" express the *same proposition* in English and in Portuguese, respectively. Much in the same way, many *blueprints* might express the same *design*.

Furthermore, it is required to express properties of the object that a design is about. For instance, the design of a phone is about an ideal (*idealized*) phone that is materialized in the individual realizations of the design. This ideal phone has ideal properties, such as ideal weight and shape. There are many ways of representing such an object. For instance, one could represent it as a special instance of the concept *Phone*, called prototype. Another alternative is to collapse both the design and the ideal object in the same entity. This is exactly the approach that was adopted in a complex design ontology that is presented in [9] and that is based on SUMO as well. However, regarding this approach, since the ideal object is also a proposition, there might be issues when modelling attributes and parts of it. For instance, if both the design of a phone and the ideal phone (the content of the design) are

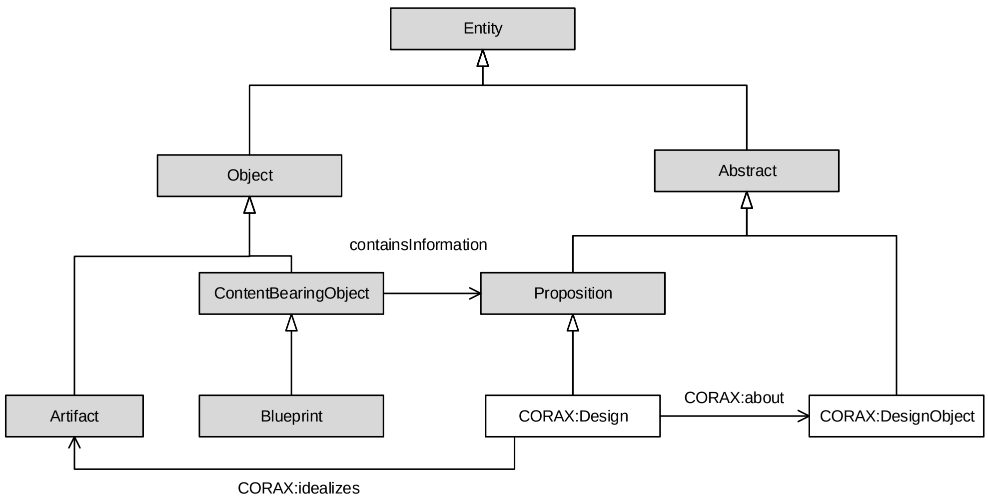

Figure 4: Entities associated to Design in CORAX.

the same entity, this entity, as a proposition, will have a designed color and a designed shape. However, *a proposition cannot have a color or a shape.* Thus, we have proposed to model the ideal object as a separate abstract entity called *Design Object*, which specifies the idealized object that is the *content* of a Design. We believe this definition better matches the intuitive notion that experts have about engineering models, which is close to the one of a blueprint; it also does not introduce a new metacategory in SUMO (such as prototype). As with physical objects, design objects also have properties, such as weight and shape. SUMO provide two main relations to represent properties, namely *attribute* and *measure*. However, these can only predicate physical objects. Thus, we created the relations *designAttribute* and *designMeasure*, which are analog to attribute and measure in SUMO, reusing their domain values. In this way, we can specify that, for instance, an idealized phone (an instance of *Object Design*) has a *design shape* and a *design weight*.

Designs *idealize* artifacts (therefore, the relation *CORAX:idelizes* in Figure 4). It is important to note that it is the *design* that idealize the artifact, and not the design object. The properties of the design object and those of the artifact possibly correlate, but we will not provide a theory about how this correlation occurs at this stage.

5.2. *Physical Environment*

Another important notion missing in SUMO is that of *physical environment*. We added this concept to CORAX in order to support specification of *robotic environments*. In our view, an *environment* is intuitively composed by a physical region, plus other eventual physical entities that characterize the environment. In addition, the definition of physical environment depends on another physical entity, which serves as a kind of landmark from which it is possible to define the main region of an environment. Landmarks can be located within the region of interest of the environment or not. For instance, the environment of an office room depends on the physical configuration of its walls, which are located ~~at~~ the environment. However, we can also define an arbitrary environment cube ~~at~~ outer space that depends on Earth as a landmark. In this case, Earth might not be located at the region defining the borders of the environment.

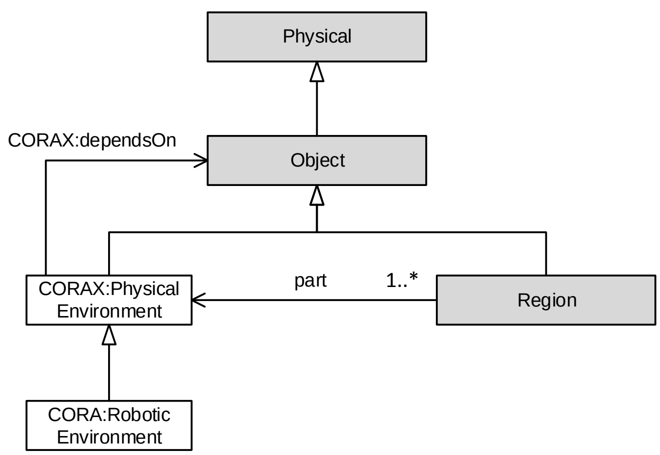

Figure 5: Concepts and relations of Physical Environment in CORAX.

More formally, we defined a physical environment in CORAX as a physical object that has at least one region as *part* and that depends on another entity. All other physical objects being part of an environment must be located at a region that is part of the environment.

5.3. *Interaction and Artificial System*

In order to properly define a *robotic system*, we have to specify what is an *artificial system*. An artificial system is simply an artifact formed from various devices (and other objects) that interact with each other and with the environment in order to fulfill a function.

This requires a basic definition of *interaction*. We define interaction as a process in which two agents participate, where an *action* generated by one agent causes a *reaction* on the other. More specifically, an interaction process is composed by two sub-processes corresponding to action and reaction. The action sub-process initiated by agent x on a patient agent y causes a reaction sub-process, having y as agent and x as patient.

6. CORA: Autonomy revisited

Autonomy is one of the most important terms in R&A, yet one of the hardest to define precisely. In the previous version of CORA, we advocated for a flexible definition that — while not being precise — could distinguish between robots that were clearly autonomous from others with questionable autonomy. In CORA, it ~~is~~ pushed a step further in ~~making~~ the modelling more versatile.

In this new version, our definitions are aligned with those from ALFUS [10], which performed an extensive study about autonomy in unmanned vehicles. In short, ALFUS states that autonomy is generally dependent on *degree of human intervention* and *context*, where the latter is characterized by *type of mission* and *environment*.

In CORA, autonomy is presented in close relation to what ALFUS defines as *modes of operation for unmanned systems*. These modes stretch~~es~~ from fully autonomous to remote controlled, representing the degree of human interaction needed for the robot to perform its task. In our view, they encapsulate the ~~expert's~~ intuitive notion of autonomy in R&A[2]. More specifically, CORA includes:

***Fully autonomous robots*:** A role for a robot performing a given task in which the robot solves the task without human intervention, while adapting to operational and environmental conditions.

***Semi-autonomous robot*:** A role for a robot performing a given task in

[2]ALFUS even goes a step further in trying to characterize absolute levels of autonomy, which correlates with the modes of operation presented here. However, the exact nature of this relation is not clarified.

which the robot and a human operator plan and conduct the task, requiring various levels of human interaction.

***Teleoperated robot*:** A role for a robot performing a given task, in which a human operator either directly controls the actuators using sensory feedback, or assigns incremental goals on a continuous basis~~,~~ from a location off the robot. A teleoperated robot will complete its last command after the operator stops sending commands, even if that command is complex and time-consuming.

***Remote controlled robot*:** A role for a robot performing a given task, in which the human operator controls the robot on a continuous basis~~,~~ from a location off the robot via only her/his direct observation. In this mode, the robot takes no initiative, and relies on continuous, or nearly continuous input from the human operator.

***Automated robot*:** A role for a robot performing a given task in which the robot acts as an automaton, not adapting to changes in the environment ~~and/or following scripted plans~~.

It is important to note that *automated robot* is not part of ALFUS' modes of operation. Experts in our group~~s~~ determined that certain robots require little human interaction, but at the same time, they are too simple to be characterized as fully autonomous. This is the case of automatons, including automated dolls and toys, which hardly react to changes in environment. Relatively simple code scripts or mechatronics determine the behavior of these robots.

One could mention at this point that some robots are inherently autonomous, or~~,~~ at least, are made with this purpose. Therefore, autonomy would not depend on context. Indeed, there is a correlation between purpose and physical capabilities of a robot, and the modes of operation it can achieve in certain tasks. Yet, this is not the definitive factor in how the robot will operate during its lifetime. It only means that such a robot *can* play a role of autonomous robots.

The fact that this classification is context-dependent also affected our modelling choices. A mode of operation is a *role*, ~~in a modelling sense~~. A role can predicate a given entity at a given time, but it can cease to predicate it at a later time. For instance, the canonical example of role is *Student*: one can predicate a person as a student at a given time, and later cease to do so.

This contrasts with rigid types, such as *Person*, that cannot stop predicating an entity without this entity ceasing to exist. That is, someone cannot cease to be a person without ceasing to exist. In general, a role is also dependent on another entity. For instance, it is necessary to a person to be enrolled at an educational institution in order to be predicated as a student.

A modeler can specify roles in many ways. The earlier version of CORA specified the various modes of operation as concepts. However, SUMO does not support roles as concepts (contrary to other ontologies [11]). For that reason, we modified the modelling of operational modes so that they became a specific type of relation present in SUMO, namely *Case Role*.

A case role in SUMO is a type of relation between an entity and a process. It describes a role that an entity plays in the process in which it participates. In order to define autonomy levels as case roles, we specialized the relation *agent* present in SUMO into the relation *robotAgent*. The relation agent is a relationship that links entities to the processes where they have an "active determinant" behavior. The relation *robotAgent* applies to robots and the processes in which the robot is the active determinant. A given operation mode depend on the way a robot determine the outcome of processes where it is involved. We represent the operational modes as subrelation of robotAgent: *fullyAutonomousRobot*, *semiAutonomousRobot*, *teleoperatedRobot*, *remoteControlledRobot* and *automatedRobot*. When a particular robot assumes a particular operation mode ~~at~~ a particular task, it is predicated with the appropriate relation. For instance, a robot that is autonomous at driving, assumes the role *fullyAutonomousRobot* ~~at~~ the autonomous driving process. The same robot can assume different operation modes in different processes, depending on the context. Interestingly, since processes can have sub-processes, a robot can assume different roles ~~at~~ different sub-processes. For instance, a cleaning robot might be fully autonomous regarding planning routes around the house, but semi-autonomous at detecting dirty places to clean.

7. RPARTS: Robot parts and extensibility

RPARTS is a sub-ontology of CORA that specifies the notions related to specific kinds of robot parts.

According to CORA, robots are (agentive) devices *composed of* other devices. There is a myriad of devices that can be robot parts, and we cannot determine in advance what *kinds* of devices can or cannot be robot parts.

Notice that this is an issue that arises at the *conceptual level*. This is a consequence of the "open-ended" nature of robots, whose design is only constrained by ~~the~~ human needs, human creativity and available technological resources. Therefore, types of devices that have never thought to be parts of a robot so far can be used as a robot part by some designer in the future. An ontology for R&A, as CORA is, must take into account this issue.

Furthermore, there is another issue regarding the notion of robot parts that arises at the *instance level*. According to our analysis, none of the instances that can be classified as robot parts is *essentially* a robot part, since they can exist by themselves when they are not connected to a robot (or when they are connected to other complex devices). For instance, a power source is essentially a device, and we cannot consider power source as a subclass of the class of robot parts, because this would imply that all instances of power sources are always robot parts. This is not true, since a specific instance of power source can be dynamically considered as a part of different complex devices during different specific time intervals. Due to this, CORA assumes that the notion of "robot part" is a *role* (in the sense previously discussed) that can be played by other devices.

In the earlier version of CORA [6], the notion of robot part was considered as a *class*, whose instances are not *essentially* instances of it. Thus, instances of robot part could cease to be robot parts, without ceasing to exist. In this sense, for example, an instance of power source that is considered as a robot part at a given moment (when it is connected to a robot) could cease to be a robot part in another moment without ceasing to exist (as an instance of power source). Thus, *Robot part* was considered as an *anti-rigid* class, in the sense of [4, 11]. Our modelling pattern [6] was developed accordingly, inspired by [11]. It represents how a specific instance of a specific kind of device (e.g., power source) could be classified as a robot part.

~~As a matter of fact,~~ this pattern becomes complex when we take into account the principles advocated in [4, 11]. According to these frameworks, an anti-rigid class (e.g., robot part) cannot subsume a rigid one (e.g., power source). Considering this principle, for each rigid class c that can play the role of robot part, we must create another specific anti-rigid class (a specific role) that will be subsumed by both c and *Robot Part*. For instance, considering an instance of the rigid class *Wheel*; it only becomes a robot part when it is attached to a particular robot. Given this condition, it becomes a member of the more specific class (e.g., "*Wheel as Robot Part*"), which is subsumed by the rigid class *Wheel* and the anti-rigid class *Robot Part* (see [6] for further details.)

We changed the representation of robot parts in the new edition of CORA. One of the reasons that justifies the changes is that the modelling pattern proposed for representing robot parts lead to domain models that are overwhelmingly complex. Some classes that must be created in order to maintain the consistency of the model do not fit well into the domain conceptualization held by most of the practitioners. Moreover, the resulting complexity is hard to manage. Therefore, this modelling pattern could hinder the broad adoption of the ontology in the domain. Another factor leading to the revision was that it is not clear how to fit the dynamical behavior that is expected from roles in the framework of SUMO. The modelling of roles adopted in [4, 11] relies on the notion of *possibility* (a *modal* notion). However, as pointed out in [12], the treatment of possibilities in SUMO is not clear.

In the current version of CORA, we have modeled the notion of robot part as a relationship between a given device d and a robot r, indicating that d is playing the role of robot part, when it is connected to r. During the analysis of the domain literature, we have identified some specific types of parts that are important to distinguish beside the notion of robot part. These types of parts—according to our analysis—would be different sub-roles of robot part, which could be played by devices with specific features. Thus, robot parts in CORA can be:

***Robot sensing part*:** responsible for sensing the surrounding environment. Formally, robot sensing parts must be measuring devices connected to the robot. A measuring device, according to SUMO, is *any device whose purpose is to measure a physical quantity*. For example, a *laser sensor* can play the role of robot sensing part~~,~~ when connected to a robot.

***Robot actuating part*:** ~~allow~~ the robot to move and act in the surrounding environment. Formally, robot actuating parts must be devices that are instruments in a process of robot motion, which is any process of movement where the robot is the agent and the patient is one of its parts.

***Robot communicating part*:** ~~role played by any device that serves as instrument in a process of~~ communication between robots and humans, by allowing the robot to send (or receive) information to (or from) a robot or a human.

***Robot processing part*:** ~~allow the robot to process~~ data and information. Formally, robot processing ~~devices~~ must be processing devices connected to the robot. A processing ~~device,~~ on the other hand, is any electric device whose purpose is to serve as an instrument in a subclass of computer process.

It is important to emphasize that although these different types of robot parts are modeled as relations between specific devices and robots, they are intended to behave as roles.

This modelling choice is also interesting regarding modularity issues. This approach ~~allows keeping~~ CORA as a minimal core of high-level concepts that provide the structure to the domain~~,~~ without going deep into details regarding the myriads of different devices that could play the roles specified here. In this sense, this structure of roles can be viewed as an interface (in the sense of *object oriented programming paradigm*) that can be implemented in different ways. Naturally, this schema poses a need of sub-ontologies to define the taxonomies of devices that can play the roles specified in CORA, such as *ontology of sensors, ontology of grippers*, etc.

8. POS: Position, orientation and pose

The position (POS) ontology is an ontology that extends SUMO and complements CORA~~,~~ developed for capturing the main concepts and relations underlying the notions of *position*, *orientation* and *pose*. These are essential for dealing with information about the relation between the robot and the surrounding space. In this section, we summarize the main notions regarding positional information. Figure 6 presents an overview of some of the main notions captured in POS, showing their relationships with concepts of SUMO.

According to the literature, roboticists and other domain experts usually utilize two kinds of positional information [13]: *quantitative* or *qualitative*. 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 as a function of a reference object. For instance, one can describe a robot as being positioned at the coordinates (x, y) in the global coordinate system, or that the robot is positioned *at the front of the box*, where "front" comprises a conical region centered on the box and pointed forward.

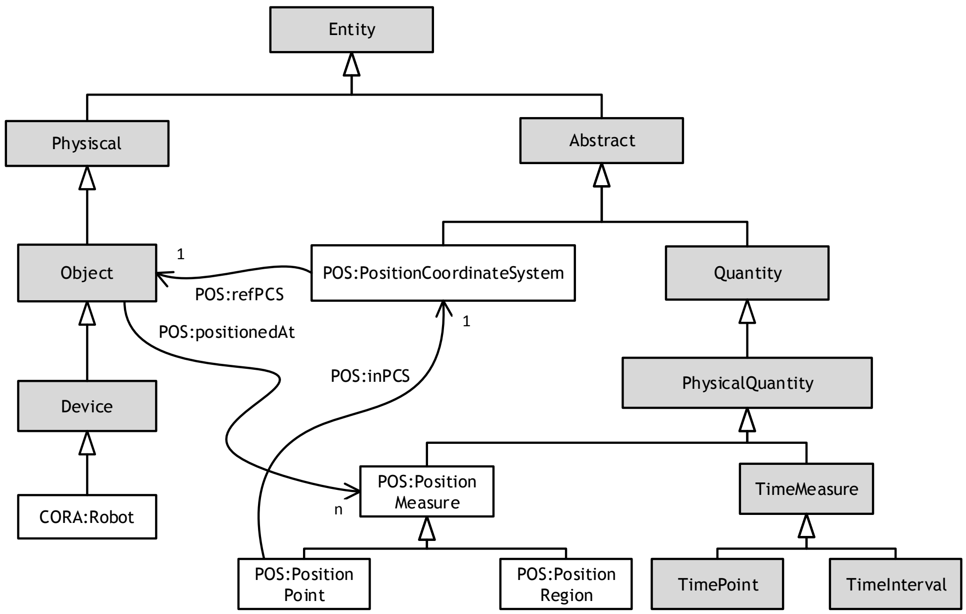

Figure 6: Fragment of POS ontology, presenting the main concepts and relations underlying the notion of *position*.

We consider that a *position* can be attributed to a (physical) *object*. In this sense, when we say that "a robot x is positioned at y", this means that there is a *measure* that relates a given "robot x" to a *position measurement* y.

Position measurements are *physical quantities* that can be *position points* or *position regions*. 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* comprising a series of position points.

A position point denotes the *quantitative* position of an object in a coordinate system. More specifically, position points are always defined in a single coordinate system.

A *coordinate system* is an *abstract* entity that is defined in relation to *a single reference* object, i.e., there is an object that is the reference to each coordinate system. 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.

This ontology does not commit to a particular kind of coordinate system. It can be stated however, that 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 coordinate $p \in c$ can represented as n-tuple $\phi(p) = (x_1(p), x_2(p), \ldots, x_n(p))$. The functions $x_1, x_2, \ldots, x_n$ are coordinate functions that attribute to p a real value in the dimension n of the coordinate system [14].

A fundamental aspect of coordinate systems is the notion of *transformation*, which maps position points in ~~a~~ coordinate system to position points in another coordinate system. The transformations can be composed generating new transformations. In our ontology, an object can display multiple positions in different coordinate systems only if there is a transformation that can map between the two.

In addition, coordinate systems are related through *hierarchies* (i.e. trees). We say that a given coordinate system~~,~~ c_1~~,~~ is parent of a coordinate system~~,~~ c_2~~,~~ if there is a transformation~~,~~ t_1 that maps the points of c_1 ~~in~~ points in c_2 and there is a transformation~~,~~ t_2~~,~~ ~~which~~ maps the points of c_2 ~~in~~ points in c_1. According to this, if two coordinate systems share a parent node in the hierarchy tree, there is a transformation between them. Usually, an agent chooses a coordinate system as the global reference frame~~,~~ ~~which~~ constitutes the *global coordinate system* (GCS) for that agent. This GCS can be *arbitrarily* chosen and does not have reference a particular coordinate frame. *Local coordinate systems* (LCS) are defined in relation to GCS by hierarchical links. This notion of hierarchy is an arbitrary one, defined by the agent.

As already stated earlier, besides the quantitative position, our ontology also provides concepts about qualitative positions~~,~~ ~~which~~ 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* is composed by poses in the coordinate system generated by a *spatial operator* on the reference object. The spatial operator is a *mathematical function* that maps reference objects to regions in a coordinate system~~,~~ in arbitrary ways.

Our ontology also allows the representation of *relative positions* of ~~the~~ objects regarding a given reference object. In general, this kind of information is represented trough *spatial relations* that are held between objects. An example of this kind of information is the relation leftOf(o, o_r), which represents that the object o is positioned at left of the object o_r. This kind of relation can be defined in our framework using the notions of *relative position* and *spatial operator*. For example, the relation leftOf(o, o_r) holds when there is a qualitative position s (a position region) that was generated by the spatial operator leftOfOp over the reference object o_r and the object o has the relative position s regarding o_r. Through this mechanism, our ontology provides the semantics ~~to~~ spatial relations like "at left of".

The usual notion of orientation is similar to position regarding its conceptual structure. Due to this, we will provide only a brief overview. An object can have a quantitative orientation defined as a value in an orientation coordinate system, as well as a qualitative orientation defined as a region in relation to a reference object. For instance, an example of use of orientation is ~~in~~ the phrase "the robot is oriented at 54 degrees"; the orientation value in this case is 54 in the circular, one-dimensional coordinate system of a compass. On the other hand, orientation regions capture a less intuitive notion. The expression "the robot is orientated to the north of the Earth" allows for interpretations where the robot has a range of possible orientation points around ~~the~~ 0 degrees. Thus, we model "north" as a region (or interval) in the one-dimensional compass coordinate system that overlaps with the general orientational extension of the object.

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

It is important to note that the current version of the POS ontology is *synchronic*. That is, it considers only facts about a single time point, just like a snapshot in time. As such, two objects cannot have the exact same quantitative position, i.e. occupy the same pose. One of the future extensions to this ontology will consider the modelling of the world along different instants. Thus, in the future, a *diachronic* version of the POS ontology will be developed.

9. Discussion

The importance of information sharing in R&A emphasizes the necessity of standardization in the field. These standards must be *clear*, *precise* and *easy to use*. CORA is ~~supposed~~ to fill that necessity: it specifies the central concepts of R&A and related fields. In this paper, we presented new additions to CORA and its adjoint domains, providing concepts about positioning, autonomy (and modes of operation) and interaction. These can already be used for building more detailed sub-domain ontologies and algorithms.

Several scenarios could take advantage of using CORA (and the related ontologies) in R&A. Firstly, CORA can be immediately applied in offline *meaning negotiation* among roboticists. That is, our ontologies could be used as *reference conceptual models* for ensuring the mutual agreement among humans regarding the meaning concepts of R&A domains.

Moreover, used as a *software component*, the ontology has an appealing application for enhancing communication among (heterogeneous) robots~~,~~ as well as~~,~~ among robots and humans. For example, a natural application for CORA is to be used as a tool developing a *middleware* for communication ~~among~~ robots, ensuring the semantic interoperability ~~among~~ them.

Our ontologies can be used as *reusable knowledge components* in *knowledge-based problem-solving processes*. Using CORA, ~~when performing its tasks~~, a robot can apply high-level logical reasoning capabilities, taking advantage of its high-level knowledge about the world for deciding the suitable action that it should perform for achieving a goal. In general, robots can use ontologies in this way to support tasks as *planning* [15, 16, 17] and *navigation* [18]. In addition, other ontologies can be integrate with our ontologies, providing a wide range of concepts and relations for allowing richer descriptions of the robot's world. Such semantic descriptions can be used by the robot in perception processes~~,~~ such as in [19, 20, 21, 22]~~,~~ for enhancing tasks that require *object recognition* through *visual perception*~~, for example~~.

Furthermore, our ontologies can be used for defining the notions underlying *robot programming frameworks*. CORA could provide these frameworks with a conceptual structure that fits the conceptualization that is shared among the roboticists. ~~Thus, for~~ instance, an object-oriented programming framework for robots based on concepts and relations in CORA would be easier to be assimilated by new programmers. In this way, dealing with these frameworks would become more natural for the practitioners of R&A. In addition, our ontologies could define standard *interfaces* for these frameworks, promoting the semantic interoperability among them.

CORA can also be used for promoting *data integration* and *semantic interoperability* among robot databases. This could have positive impacts to the *knowledge management* process of companies that commercialize products and components for R&A field.

10. Future work: what should we expect from now?

CORA and related ontologies still do not cover some important areas in R&A. For instance, *control* still needs to be taken into account. This issue is complex, since it involves other important concepts in robotics~~,~~ such as perception, planning and action. It should also incorporate information ranging from simple classical controllers — such as PID — to complex non-linear control. It should account for different control strategies as well.

The notion of *task* is also important in the domain. Since ~~nowadays~~ robots should be able to operate in complex scenarios, task definitions must be clear to allow robots to communicate with each other, other machines and humans. In this sense, ontologies play a clear role in task~~s~~ specification. CORA must be designed to allow several types of tasks in various environments, e.g., grasp, move, scan and so on. Future work will be devoted to the ontological characterization of what kind of entity a task is. We believe that a good starting point is to separate *tasks* from *task~~s~~ executions*, ~~for example~~. With this distinction, we acknowledge that tasks are *abstract* entities that describe goals to be reached; while tasks executions are *events* composed by *actions* that are performed by robots in the world in order to reach a given goal. Moreover, in future steps it is necessary to identity the basic kinds of tasks that robots usually perform. These task~~s~~ definitions will be the basis of more complex task definitions. CORA must define clearly the interfaces to domain ontologies, like industrial [23] or surgical [24] [25].

Furthermore, planning is also an important related issue. Given a task, the *plan* is an abstract partially ordered set of references to actions, which when performed contribute ~~for~~ the task execution. Possibly, any development in this area should take into account SUMO concepts related to plan.

Finally, CORA and related ontologies do not represent change in time (e.g. changes in sensor data). We envisage *diachronic* version of CORA, where time is taken into account.

Acknowledgment

The IEEE-SC WG is supported by the IEEE Robotics and Automation Society. This work is partially supported by FCT, through IDMEC, under LAETA Pest-OE/EME/LA0022. The authors acknowledge the support of Brazilian CNPq, Petrobras PRH PB-17 and the Hungarian Eötvös Scholarship. T.H. is a Bolyai Fellow of the Hungarian Academy of Sciences.

References

[1] Studer, R., Benjamins, V.R., Fensel, D.. Knowledge engineering: Principles and methods. Data and Knowledge Engineering 1998;25(1-2):161–197. URL `http://dx.doi.org/10.1016/S0169-023X(97)00056-6`.

[2] Schlenoff, C., Prestes, E., Madhavan, R., Gonçalves, P.J.S., Li, H., Balakirsky, S., et al. An IEEE standard ontology for robotics and automation. In: Proc. of the 2012 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems,Vilamoura. 2012, p. 1337–1342.

[3] Fernández, M., Gómez-Pérez, A., Juristo, N.. METHONTOLOGY: from ontological art towards ontological engineering. In: Ontological Engineering; vol. 6 of *AAAI Spring Symposium*. 1997, p. 33–40.

[4] Guarino, N., Welty, C.A.. An overview of OntoClean. In: Staab, S., Studer, R., editors. Handbook on Ontologies. Intl. Handbooks on Information Systems; Springer Berlin Heidelberg. ISBN 978-3-540-70999-2, 978-3-540-92673-3; 2009, p. 201–220. URL `http://link.springer.com/chapter/10.1007/978-3-540-92673-3_9`.

[5] Niles, I., Pease, A.. Towards a standard upper ontology. In: Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001. FOIS '01; New York, NY, USA: ACM. ISBN 1-58113-377-4; 2001, p. 29. doi:\bibinfo{doi}{10.1145/505168.505170}. URL `http://doi.acm.org/10.1145/505168.505170`.

[6] Prestes, E., Carbonera, J.L., Rama Fiorini, S., M. Jorge, V.A., Abel, M., Madhavan, R., et al. Towards a core ontology for robotics and automation. Robotics and Autonomous Systems 2013;61(11):1193–1204.

[7] Carbonera, J.L., Fiorini, S.R., Prestes, E., Jorge, V.A., Abel, M., Madhavan, R., et al. Defining positioning in a core ontology for robotics. In: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. IEEE; 2013, p. 1867–1872.

[8] Balakirsky, S., Kootbally, Z., Kramer, T., Pietromartire, A., Schlenoff, C., Gupta, S.. Knowledge driven robotics for kitting applications. Robotics and Autonomous Systems 2013;61(11):1205–1214.

[9] torga, M., Andreasen, M.M., Marjanovi, D.. The design ontology: foundation for the design knowledge exchange and management. Journal of Engineering Design 2010;21(4):427–454. doi:\bibinfo{doi}{10.1080/09544820802322557}.

[10] Huang, H.M., Messina, E., Albus, J.. Toward a generic model for autonomy levels for unmanned systems (alfus). Tech. Rep.; DTIC Document; 2003.

[11] Guizzardi, G.. Ontological foundations for structural conceptual models. PhD thesis; University of Twente; The Netherlands; 2005.

[12] Oberle, D., Ankolekar, A., Hitzler, P., Cimiano, P., Sintek, M., Kiesel, M., et al. Dolce ergo sumo: On foundational and domain models in the smartweb integrated ontology (swinto). Web Semantics: Science, Services and Agents on the World Wide Web 2007;5(3):156–174.

[13] Ye, J., Coyle, L., Dobson, S., Nixon, P.. A unified semantics space model. In: Location-and context-awareness. Springer; 2007, p. 103–120.

[14] Morita, S.. Geometry of differential forms. No. v. 201 in Translations of mathematical monographs; Providence, R.I: American Mathematical Society; 2001. ISBN 0821810456.

[15] Provine, R., Schlenoff, C., Balakirsky, S., Smith, S., Uschold, M.. Ontology-based methods for enhancing autonomous vehicle path planning. Robotics and Autonomous Systems 2004;49(1):123–133.

[16] Galindo, C., Fernández-Madrigal, J.A., González, J., Saffiotti, A.. Robot task planning using semantic maps. Robotics and Autonomous Systems 2008;56(11):955–966.

[17] Belouaer, L., Bouzid, M., Mouaddib, A.. Ontology based spatial planning for human-robot interaction. In: Temporal Representation and Reasoning (TIME), 2010 17th International Symposium on. IEEE; 2010, p. 103–110.

[18] Bateman, J., Farrar, S.. Modelling models of robot navigation using formal spatial ontology. In: Spatial Cognition IV. Reasoning, Action, Interaction. Springer; 2005, p. 366–389.

[19] Modayil, J., Kuipers, B.. Autonomous development of a grounded object ontology by a learning robot. In: Proceedings of the national conference on Artificial intelligence; vol. 22. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999; 2007, p. 1095.

[20] Suh, I.H., Lim, G.H., Hwang, W., Suh, H., Choi, J.H., Park, Y.T.. Ontology-based multi-layered robot knowledge framework (omrkf) for robot intelligence. In: Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on. IEEE; 2007, p. 429–436.

[21] Johnston, B., Yang, F., Mendoza, R., Chen, X., Williams, M.A.. Ontology based object categorization for robots. In: Practical Aspects of Knowledge Management. Springer; 2008, p. 219–231.

[22] Lim, G.H., Suh, I.H., Suh, H.. Ontology-based unified robot knowledge for service robots in indoor environments. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 2011;41(3):492–509.

[23] Balakirsky, S., Kootbally, Z., Kramer, T.R., Pietromartire, A., Schlenoff, C., Gupta, S.. Knowledge driven robotics for kitting applications. Robotics and Autonomous Systems 2013;61(11):1205–1214.

[24] Gonçalves, P.. Towards an ontology for orthopaedic surgery, application to hip resurfacing. In: Proceedings of the Hamlyn Symposium on Medical Robotics. London, UK. ISBN 978-0-9563776-4-7; 2013, p. 61–62.

[25] Haidegger, T., Barreto, M., Gonçalves, P., Habib, M.K., Ragavan, V., Li, H., et al. Applied ontologies and standards for service robots. Robotics and Autonomous Systems 2013;61(11):1215–1223. doi:\bibinfo{doi}{10.1016/j.robot.2013.05.008}. URL `http://dx.doi.org/10.1016/j.robot.2013.05.008`.